Highlights

Hybrid Deep learning methods for forecasting COVID-19 using metaheuristic optimizer in Case of studies 4 top countries in all continents

Mousa Alizadeh,Azam Seilsepour,Ali Ostadi,Mobina Mousapour Mamoudan,Mohammad T H Beheshti

- Proposing a hybrid deep neural network model for forecasting the trend of infected, recovered, and dead people of the coronavirus in 24 countries from six continents.
- Fine-tuning the hyperparameters of proposed method using Whale Optimization Algorithm.
- Comparing the forecasted results with other state-of-the-art methods.

Hybrid Deep learning methods for forecasting COVID-19 using metaheuristic optimizer in Case of studies 4 top countries in all continents

Mousa Alizadeh[a], Azam Seilsepour[b], Ali Ostadi[c], Mobina Mousapour Mamoudan[d] and Mohammad T H Beheshti[a]

[a]Department of Electrical and Computer Engineering, Tarbiat Modares University, Tehran, Iran
[b]Department of Computer Engineering, Islamic Azad University, Central Tehran Branch, Tehran, Iran
[c]Department of Industrial Engineering, Islamic Azad University, North Tehran Branch, Tehran, Iran
[d] School of Industrial and Systems Engineering, College of Engineering, University of Tehran, Tehran, Iran

ARTICLE INFO

Keywords:
COVID-19
hybrid deep learning
forecasting
meta-heuristic optimization
CNN
LSTM
WOA

ABSTRACT

The novel Coronavirus, also known as COVID-19, was declared by World Health Organization (WHO) as a global pandemic On March 11th, 2020. Originating in Wuhan, Hubei province in China around December 2019, it has dramatically spread over the world and brought new challenges to the governments and the research community. Handling the huge number of infected people and managing the available resources effectively has become a crucial task for healthcare systems. Therefore, accurate short-term prediction of the number of infected, cured, and dead people can help the authorities improve managing the available resources. Recently, considerable research attention has been paid to the use of deep learning models in forecasting time-series data. This paper presents a hybrid deep learning model, composed of Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM), to predict the number of infected, cured, and dead people in twenty-four countries from six continents. Moreover, the Whale Optimization Algorithm (WOA) is utilized to fine-tune the hyperparameters of the proposed method. Comparing the results demonstrates that the proposed method outperforms the other state-of-the-art methods.

1. Introduction

In late 2019, an acute respiratory syndrome coronavirus 2 (SARS-CoV-2) was identified and reported in the city of Wuhan, China, and spread rapidly around the world, so the World Health Organization (WHO) declared it a pandemic. This pandemic spread significantly around the world and challenged the healthcare systems of most developing and developed countries. As a result, the need to effective management of patients increased for governments and hospitals. On the other hand, researchers and policymakers found that for strategic planning in the health and treatment sector, there will be a need to predict the number of corona patients. Because this leads to improving the use of available resources in hospitals, management strategies for optimal management of infected patients, and determining macro policies [1].

In other words, it can be said that predicting, identifying, and understanding the trend of the number of confirmed and recovered cases and the number of people who died due to COVID-19 can lead to improved decisions to slow down or stop the spread of this disease. That is why this topic has been of interest to academics, global communities, and policymakers, and researchers have tried to show when the rate of infection of COVID-19 reaches its maximum, how long it will take for the pandemic to stop, the total number of people who will be affected by corona disease will be, and how many people will recover or die from the corona [2, 3]. Various forecasting methods, such as base curve fitting methods, traffic interaction models, and machine learning approaches, have been developed by researchers in different scientific disciplines, but the results show that data-based research and machine learning-based approaches can provide rich insights into society's challenges [4].

On the other hand, data related to epidemics such as COVID-19 may not be fundamentally considered linear systems. As a result, the conventional statistical models such as Auto Regressive Moving Average (ARMA), Moving

*Corresponding author
ORCID(s):
[1]m_alizadeh@modares.ac.ir

Average (MA), Auto Regressive (AR) methods are not appropriate for forecasting this type of data. However, deep learning techniques, in issues related to COVID-19, has filled the gap between different approaches and medical sciences. In other words, it can be said that predictive algorithms can potentially reduce the pressure on healthcare systems by establishing relationships between various parameters and speeding up processes to obtain optimized results [5].

Deep learning, a subset of machine learning methods, is one of the most popular and widely used methods that has received a lot of attention in the past few years. In addition to reaching the optimal solution and saving time, deep learning algorithms have achieved a very high accuracy in diagnosis, which is why they are trendy among researchers. On the other hand, each method has its pros and cons.

One of the weak points of neural networks is that their performance is highly dependent on the values of hyperparameters. Researchers use the random and grid search methods to fine-tune the hyperparameters. These methods are time-consuming methods that have more complex mathematical calculations. To improve the performance of prediction models and increase the accuracy of these algorithms, the advanced approaches are required. Recently, the researchers use the metaheuristic methods to find the optimal values of hyperparameters.

Convolutional Neural Networks (CNN) are capabale of extracting local and deep features, and Long Short-Term Memory (LSTM) models can capture long-term dependencies. They are widely used to model sequentional data, such as time-series analysis.This study uses a hybrid neural network model called CNN-LSTM, a combination of CNN and LSTM, to forecast the number of infected, cured, and dead people of Covid-19. Additionally, to fine-tune the hyperparameters of the CNN-LSTM, Whale Optimization Algorithm (WOA) has been used.

The above model has been analyzed on the time series data set of infected people, cured people, and dead people on six continents worldwide. Since various factors, such as weather countries' policies regarding quarantine and the population of each continent, may affect the number of people infected with COVID-19, these factors are different in different continents. From each continent, the four countries that had the highest rate of corona infection have been selected so that the number of infected people, the number of people who have recovered, and the number of people who have died due to corona are examined.

In critical situations such as the COVID-19 epidemic, in addition to the need to save medical, logistical, and human resources, time is also one of the main priorities because even one hour of saving time can save the lives of several people. The presentation of the above model and using the WOA to optimize the Hyperparameters of the model, in addition to increasing the prediction speed and accuracy of the models, can lead to reduction and saving of time in calculating the algorithms. This article aims to introduce a reliable model based on time series data to predict the short-term trend of COVID-19 in twenty-four world countries. These predictions can be helpful for policymakers and the health systems sector of countries because using these results can identify the best algorithm during a crisis and give policymakers time to react to changes caused by disease outbreaks.

To achieve the above goals, the second part is dedicated to the literature review, and the building blocks of the proposed method are described briefly in Section 3. The fourth section is data analysis, and the proposed method is described in Section 5. Section 6 includes the configuration settings, hyperparameter tuning, and evaluation results. Finally, the seventh section is dedicated to the conclusion and future work.

2. Literature Review

COVID-19 affected the lives of people all over the world and took the lives of thousands around the world. On the other hand, the social and economic consequences of COVID-19 sometimes inflicted irreparable damage on countries, and it was able to confront the world with new challenges and crises. Therefore, many researchers have tried to minimize the damage caused by this disease by providing models to predict the rate of coronary artery disease. Various approaches to forecasting have also been developed. For example, [6] developed a discrete-time stochastic model to describe the dynamics of pandemic expansion. Prediction Models Traditional time series have been routinely studied to predict COVID-19 cases [7], [8], [9]. Because the traditional Automatic Integrated Moving Average (ARIMA) model is suitable for describing short-term autocorrelation in time series data, researchers have commonly used it. [10] used time series methods such as the ARIMA to predict the number of validated items. Other studies have been developed in India using traditional ARIMA modeling [11]. In 2022, other researchers from Bangladesh used ARIMA to predict verified people, deaths, and recoveries in COVID-19 [12]. [13] used a hybrid ARIMA and Prophet Model to predict daily confirmed and cumulative confirmed cases in India.

[4] used the Patient Information Based Algorithm (PIBA) to estimate the number of COVID-19-related deaths in China. Their results show that the overall mortality rate in Hubei and Wuhan is 13 percent and between 0.75 percent and 3 percent in the rest of China.[14] developed a hybrid algorithm for predicting short-term COVID-19 cases in Louisiana, USA. Their proposed algorithm combines selecting similar day features using Xgboost, K-Means, and LSTM. In 2020, researchers in India compared the Convolutional LSTM, BILSTM, and Stacked LSTM algorithms to the MAPE test to use an optimal algorithm to predict the number of coronary arteries in India [15]. Other researchers used simple Recurrent Neural Network (RNN) algorithms, LSTM, two-way LSTM (BiLSTM), Gateway Recurrent Units (GRUs), and Variable Automatic Encoder (VAE) to predict COVID-19 in six countries [1].

[16] used time-series data related to data from COVID-19 in Iran. Using the LSTM algorithm, they sought to include the interaction of all classes of coronary arteries, deceased coronary arteries, and the number of recovered individuals in the prediction process using this method. [17] used the multi-head attention-based method (ATT_BO), CNN-based method (CNN_BO), and LSTM-based method (LSTM_BO) to forecast short-term and long-term COVID-19. For this reason, they perform two types of datasets. [18] used ARIMA, cubist regression (CUBIST), random forest (RF), Ridge Regression (RIDGE), Support Vector Regression (SVR), and stacking-ensemble learning to forecast the total number of patients in Brazil in 1, 3, and 6 days. [5] used a Network Inference based Prediction Algorithm (NIPA) to forecast the corona in the Chinese city of Hobby and the Netherlands. They also compared their method with LSTM and sigmoid curve methods.

In 2022, other researchers used the ANN algorithm to forecast the number of cumulative cases of infection and deaths in Brazil. They also used a substantial mitigation procedure adopted (mandatory use of masks) was experimented as an input to evaluate the improvement in the results [19]. In the same years, [20] from the USA presented a relatively non-parametric random forest model to forecast the number of COVID-19 cases in the U.S. [21] used CNN, LSTM, and the CNN-LSTM algorithm to predict the number of COVID-19 in Brazil, India, and Russia. They also compared the performance of these models with the previously developed deep learning models. [22] used the Interpretable Temporal Attention Network (ITANet) algorithm to forecast COVID-19. By using this algorithm, they wanted to infer the importance of government interventions.

Although much research has been developed during the Corona to provide the best predictive results, very few studies have used a combination of different neural network algorithms to optimize results and increase prediction accuracy. In other words, to use the important and practical features of different algorithms, it combines them together to get better results from the models. Also, the use of Metaheuristic algorithms has rarely been seen in articles. To this end, and to optimize the results, this study, in addition to using a hybrid neural network, for the first time combines two Metaheuristic algorithms of gray wolf and whale on the proposed neural network. Finally, the proposed model is examined on the data of twenty-four countries from six continents in the world.

3. Basic of research

This section briefly explains the proposed method's building blocks, including CNN, LSTM, and WOA. The proposed method combines the CNN and LSTM for prediction and employs the WOA for hyperparameter tuning.

3.1. Convolutional Neural Network

In the 1960s, Hubel et al. conducted biological research. Their research demonstrated that the transmission of visual information from the retina to the brain was complemented by multiple levels of receptive field excitation, which eventually suggested a CNN. CNN is a feed-forward neural network composed of the input, convolution, pooling, fully connected, and output layer. The data in the input layer is processed through feature transformation and extraction using the convolution and pooling layer. This local information of the convolution and the pooling layer is further integrated from the fully connected layers and mapped to the output signals by the output layer. The output layer generates output after receiving the properties [23]. The CNN's structure is shown in Figure 1. Equation 1 also shows the CNN calculation formula. In Equation 1, N, W, and F denote the output size, input size, and kernel size, respectively. Also, P and S indicate padding size and step size, respectively.

$$N = \frac{W - F - 2P}{S + 1} \tag{1}$$

The convolution layer is responsible for extracting features from the input data. In general, the convolution layer includes the convolution kernel, the convolutional layer parameters, and the activation function. The convolution layer

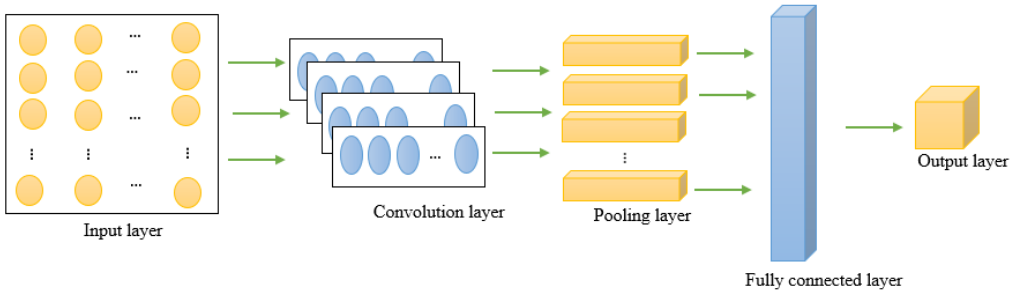

Figure 1: The CNN Structure

is the most significant and unique layer of CNN. It extracts the features of input variables by convolution kernels. The kernel size of convolutions is smaller than the input matrix. The convolution layer uses convergence operations to output the feature map instead of the general matrix operation [24]. The calculation of each element in the feature map is in the form of Equation 2. In Equation 2, $x_{i,j}^{out}$ is the output value in row i and column j of the feature map. $x_{i+m,j+n}^{in}$ is the value in row i and column j of the input matrix. $f_{cov}(0)$ is the selected activation function. $w_{m,n}$ shows the weight in row m and column n for the convolution kernel and b shows the bias of the convolution kernel.

$$x_{i,j}^{out} = f_{cov}(\sum_{m=0}^{k}\sum_{n=0}^{k} w_{m,n}x_{i+m,i+n}^{in} + b) \tag{2}$$

In general, the input matrix uses multiple kernels for the convolution layer. Each convolution kernel extracts a feature from the input matrix and creates a feature map. After that, the pooling layer reduces the length and width of the previous feature map and improves the computational efficiency with down-sampling. The output of feature vectors by the convolutional layer can be reduced through the pooling layer. Also, the results can be improved at the same time. Because CNN has a good ability to extract grid data features, m variables of any type were expanded to n stations to obtain a matrix of m rows and n columns. We can say that CNN as a whole, the fully connected layer is a classifier. It is located at the end of the network and performs regression classification on the extracted features. Thus, we can compose CNN in two parts. The first part includes feature extraction (convolution, activation function, pooling) and the second part is classification and recognition (fully connected layer) [25]. Figure 4 shows the structure of the CNN model.

3.2. Long Short-Term Memory

RNNs are widely used in time-series analysis and sequence modeling because they are able to process the current data using previous data and directed cycles. They employ a memory mechanism to remember the state of previous data but they face Vanishing and Exploding Gradient problems when learning long-term dependencies. The LSTM and GRU networks are advanced models of the RNNs, proposed to solve these problems [26, 27, 28].

"cells" are the main information-processing units in LSTM. These cells are more sophisticated neurons in typical MLP. LSTM cells, like neurons, can be connected and stacked to transmit temporal information. LSTM can turn information into a cellular state. This feature is called a gate. Figure **??** shows the structure of the LSTM network. Each LSTM unit has three gates: input gate, forget gate, and output gate. These gates are used to provide read, write and reset functions, respectively. Cell mode is the path of information transfer that allows the transfer of information in order. Gates are used to updating or discarding historical information. This feature helps the LSTM to decide which information is useful in the long term. In Equations of LSTM, C_{t-1} is the cell state from the previous module, d_{t-1} is the output of the previous module, X_t is the current input, used to generate new memory, and the output information includes the cell state $C_t$transmitted later, new output d_t.

The forget gate of LSTM is a valve. A lot of information will flood into the memory when the input gate is always open. At this time, a forgetting mechanism is required to remove the information in the memory. We call this the forget gate. It looks at d_{t-1}(previous output) and X_t(current input) and outputs a number among 0 with 1 for every digit in the cell state C_{t-1}(previous state) that 1 shows completely saved, and 0 shows fully deleted. The calculation formula

is shown in Equation 3. In Equation 3, W_f is the weight matrix, b_f is the bias term, and F is the output through this network with a number in the range (0, 1) that indicates the probability of the previous cell state being forgotten. 1 means "completely reserved", and 0 means "completely discarded".

$$f_t = sigmoid(W_f[d_{t-1}, X_t] + b_f) \tag{3}$$

In LSTM, after the circulating neural network "forgets" part of the previous state the input gate requires supplementing the newest memory from the current input. This process could be fulfilled by the "input gate". In LSTM, the input gate consists of two parts. The first part is about, a sigmoid layer named the "input threshold layer" that decides which values we need to renew. The second part is about, a *tanh* layer, that establishes a new candidate vector $\tilde{C}_t$, which will be increased to this state. This relation shows in Equations 4, 5, and 6.

$$h_t = \sigma(W_n.[d_{t-1}, X_t] + b_n) \tag{4}$$

$$\tilde{C}_t = \tanh(W_m.[d_{t-1}, X_t] + b_m) \tag{5}$$

$$C_t = F_t * C_{t-1} + h_t * \tilde{C}_t \tag{6}$$

In Equations 4, 5, and 6, W_n represents the weight matrix, b_n represents the deviation element, W_m represents the weight matrix to update the unit status, b_m represents the offset element to update the unit status [29], and C_t represents the status of the updated memory unit. In Equation 7, enter the gate h_t and $_t$, run the dot product to decide whether to update the state of the time step memory unit; the forgetting gate F_t takes the scalar product with C_{t-1} to decide whether it is necessary to keep the initial state of the unit memory of the time step.

The output gate in LSTM is the current time output that must be generated after calculating the new status. Also, it uses to control the level of filtering of the storage unit status in this layer. The output gate sets the output at this time based on the last state, the last time output, and the current input. Its calculation formula is as Equations 7 and 8. First of all, we have to use the sigmoid activation function to get an O_t whose value is in the range [0, 1]. Then multiply the state of the memory cell C_t by the tanh activation function then multiplies by O_t, this is the output of this layer. d_t is not only associated with the input X_t in time step t and with the activation value d_{t-1} of the hidden layer at the previous time step but also with the state of the memory unit C_t under the time step.

$$d_t = O_t * \tanh(C_t) \tag{7}$$

$$O_t = \sigma(W_0.[d_{t-1}, X_t] + b_0) \tag{8}$$

3.3. Hyperparameter Tuning

Since the performance of deep learning networks highly depends on values of hyperparameters, the researchers commonly use the Grid and Random search methods to find the optimal values for the hyperparameters of deep neural networks [30]. Recently, researchers have employed metaheuristics algorithms to fine-tune the hyperparameters [31]. The proposed method uses the WOA to find the optimal values of hyperparameters. The following subsection explains this algorithm in more detail.

3.3.1. Whale optimization algorithm

One of the nature-based and population-based optimization algorithms is the WOA [32]. The whale has a specific method of hunting known as the Bubble-net method. Whales prefer to hunt a group of krill or small fish near the surface of the water. For this reason, by creating index bubbles along a circle, they pull small fish to the surface of the water. Encircling, exploitation and exploration are three steps of the WOA algorithm.

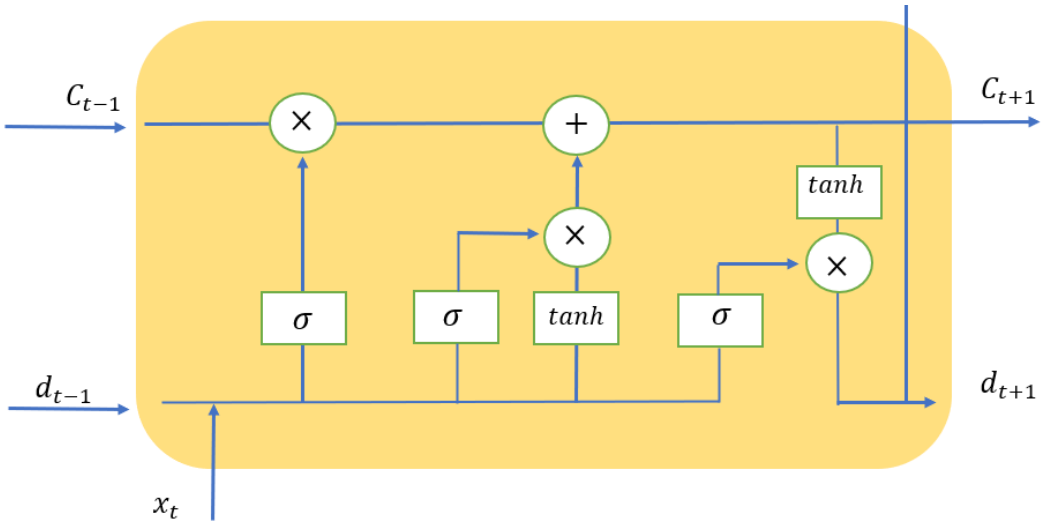

Figure 2: The LSTM structure

Whales can identify hunting grounds and surround them. Since the optimal design location in the search space is not known, by comparison, the algorithm assumes that the best candidate solution at the moment is target hunting or close to optimal. Once the best search engine is identified, other search agents try to update their location to the best search engine that is shown in Equations 9 and 10. In these equations, t represents the current iteration, $\overrightarrow{A}$ and $\overrightarrow{C}$ are the coefficient vectors, $\overrightarrow{X}^*$ is the location vector of the best solution obtained now, and $\overrightarrow{X}$ is the location vector. It should be noted that if there is a better solution, $\overrightarrow{X}^*$ should be updated in each iteration. $\overrightarrow{A}$ and $\overrightarrow{C}$ are calculated as Equations 11 and 12. It should be noted that $\vec{a}$ decreases linearly from 2 to 0 during repetitions (in both exploration and extraction phases) and $\vec{r}$ is a random vector between 0 and 1.

$$\overrightarrow{D} = |\overrightarrow{C}.\overrightarrow{X}^*(t) - \overrightarrow{X}(t)| \tag{9}$$

$$\overrightarrow{X}(t+1) = \overrightarrow{X}^*(t) - \overrightarrow{A}.\overrightarrow{D} \tag{10}$$

$$\overrightarrow{A} = 2\vec{a}.\vec{r} - \vec{a} \tag{11}$$

$$\overrightarrow{C} = 2.\vec{r} \tag{12}$$

The exploitation phase in the Whale Optimization Algorithm is designed in two methods, which are called as shrinking encircling mechanism and spiral updating position. The shrinking encircling mechanism is achieved by increasing the value of $\vec{a}$, which is shown in Equation 11. The oscillation range $\overrightarrow{A}$ is reduced by $\vec{a}$. In other words, $\overrightarrow{A}$ is a random value in the distance a to $-a$, and a decreases from 2 to 0 during repetitions. By selecting random values of $\overrightarrow{A}$ at intervals of 1 to -1, the new location of the search agent can be defined anywhere between the main location of the agent and the location of the current best agent. Spiral updating location first calculates the distance between the wall located in the X^* and Y coordinates of the bait in X^* and Y^*. A spiral equation is created between the position of the whale and the prey to mimic the whale-shaped spiral motion shown in Equation 13. In Equation 13, $\overrightarrow{D}'$ refers to the distance from the 1st whale to the prey (the best solution obtained so far), b is a constant for defining the shape of a logarithmic helix and is a random number between 1 and -1. It should be noted that the humpback whale swims around its prey along

a contractile circle and at the same time in a spiral path. To model this simultaneous behavior, it is assumed that the whale selects one with a 50% probability of *a* shrinking encircling mechanism or spiral updating position model to update the position of the whales during optimization. The mathematical model is shown in Equation 14.

A similar method based on $\overrightarrow{A}$ variation can be used to search for prey. In fact, humpback whales search randomly, depending on each other's location. Therefore, $\overrightarrow{A}$ is used with random values greater than or less than -1 to force the search agent to move away from the reference whale. Unlike the extraction phase, in order to update the position of the search agent in the exploration phase, instead of using the data of the best search agent, random agent selection has been used. This mechanism, along with $|\overrightarrow{A}|>1$, emphasizes exploration and allows the WOA algorithm to perform a global search. The mathematical model is shown in Equations 15 and 16.

$$\overrightarrow{X}(t+1) = \overrightarrow{D}'.e^{bl}.cos(2\pi l) + \overrightarrow{X}^*(t) \tag{13}$$

$$\overrightarrow{X}(t+1) = \begin{cases} \overrightarrow{X}^*(t) - \overrightarrow{A}.\overrightarrow{D}, & if\ p < 0.5 \\ \overrightarrow{D}'.e^{bl}.cos(2\pi l) + \overrightarrow{X}^*(t), & if\ p \geq 0.5 \end{cases} \tag{14}$$

$$\overrightarrow{D} = |\overrightarrow{C}.\overrightarrow{X}_{rand} - \overrightarrow{X}| \tag{15}$$

$$\overrightarrow{X}(t+1) = \overrightarrow{X}_{rand} - \overrightarrow{A}.\overrightarrow{D} \tag{16}$$

In this equation, $\overrightarrow{X}_{rand}$ is the randomly selected position vector (random whale) of the current population. The WOA algorithm starts solving the problem with a set of random solutions. In each iteration, the search agents update their position according to the randomly selected search agent with the best current solution. Parameter *a* is reduced from 2 to 0 to provide exploration and extraction, respectively. A random search agent is selected in $|\overrightarrow{A}|>1$ mode, while the best solution is selected when $|\overrightarrow{A}|<1$ to update the position of the search agents. Depending on the value of *p*, the WOA algorithm has the ability to choose between circular or spiral motion. Finally, the WOA algorithm ends with satisfying the termination conditions.

4. Data Description

The first case of COVID-19 was reported in China in 2019 and quickly spread worldwide, claiming many lives. In this study, the trend of the growth of COVID-19 in the countries with the highest infection rate with COVID-19 has been investigated. In general, twenty-four countries from six continents have been studied. Figure 3 shows the growth trend of this disease in these twenty-four countries.

The dataset used in this article is time series and multivariate. The variables used are the number of people infected with corona disease, the number of people who died, and the number of people who recovered from 3/22/2020 to 12/1/2022, collected daily. The proposed model has been implemented for all variables for all countries to predict the number of people who have died, the number of people who have recovered, and the number of people with corona disease for all countries. Also, the first 80% of the data is used for training the model, and the last 20% of the data is used for testing and evaluating the model.

In general, the data from twenty-four countries from six continents have been used to evaluate and check the accuracy of the proposed model. The countries surveyed on the American continent are Canada, the United States, Argentina, Brazil, Mexico, Peru, Panama, and Colombia. European countries studied include Russia, France, Italy, New Zealand, and the United Kingdom. The four countries with the highest rates of COVID-19 in Asia are Iran, India, Turkey, and Indonesia. Selected countries from the African continent are Egypt, Morocco, Africa, and Tunisia. Also, the COVID-19 data in New Guinea, Australia, and Fiji from Australia and Oceania were obtained. In the next step, we examined the growth and decline of the number of people with coronary heart disease among these twenty-four countries to use deep learning to predict the number of patients, the number of people who have recovered, and the number of people who have died from coronary artery disease. The twenty-four countries examined in this study are shown in Figure 5.

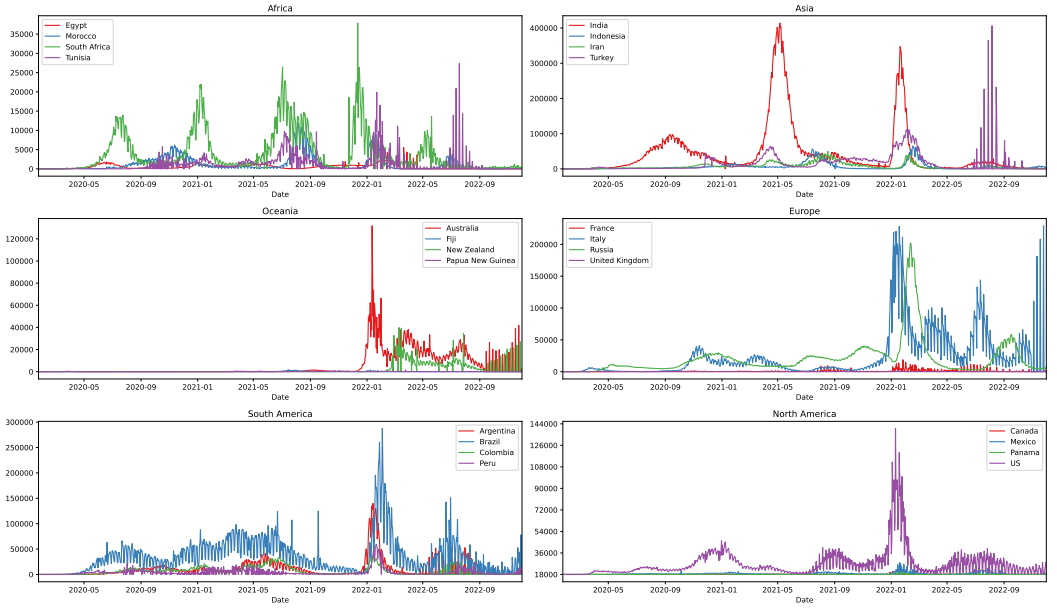

Figure 3: The growth trend of confirmed covid 19 infected patients in 24 countries by continent

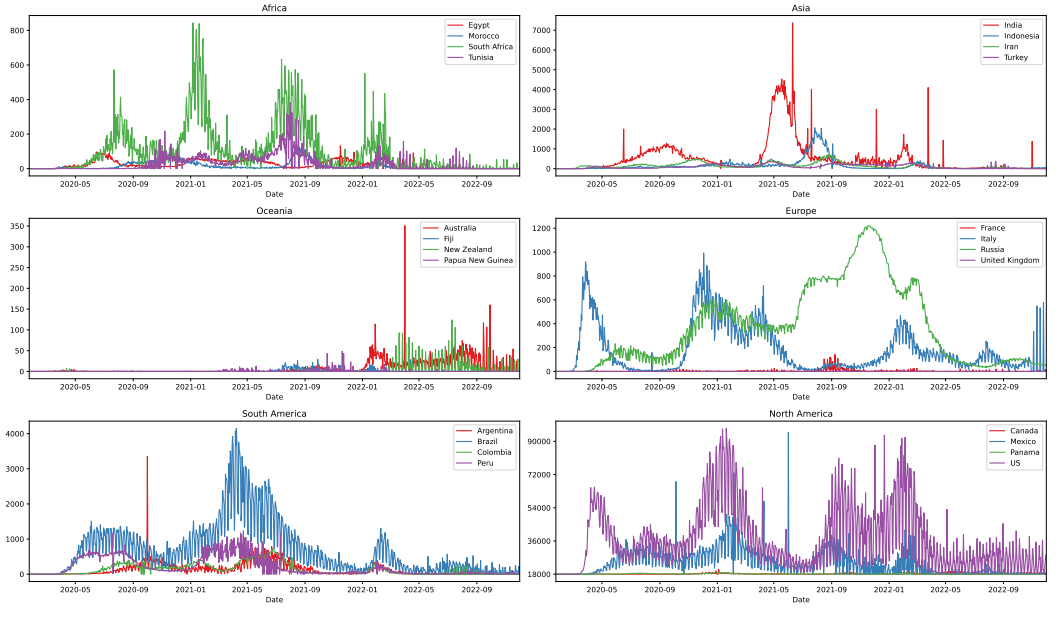

Figure 4: The growth trend of confirmed covid 19 infected patients in 24 countries by continent

5. Proposed Method

The proposed method combines the CNN and LSTM networks for forecasting the number of infected, cured, and dead people. As described in 3.1, the CNNs are able to extract local and deep features using their convolutional layers. On the other hand, the LSTM models can capture long-term dependencies, making them a good choice for sequence modeling. As a result, their combination can improve the accuracy of prediction [33].

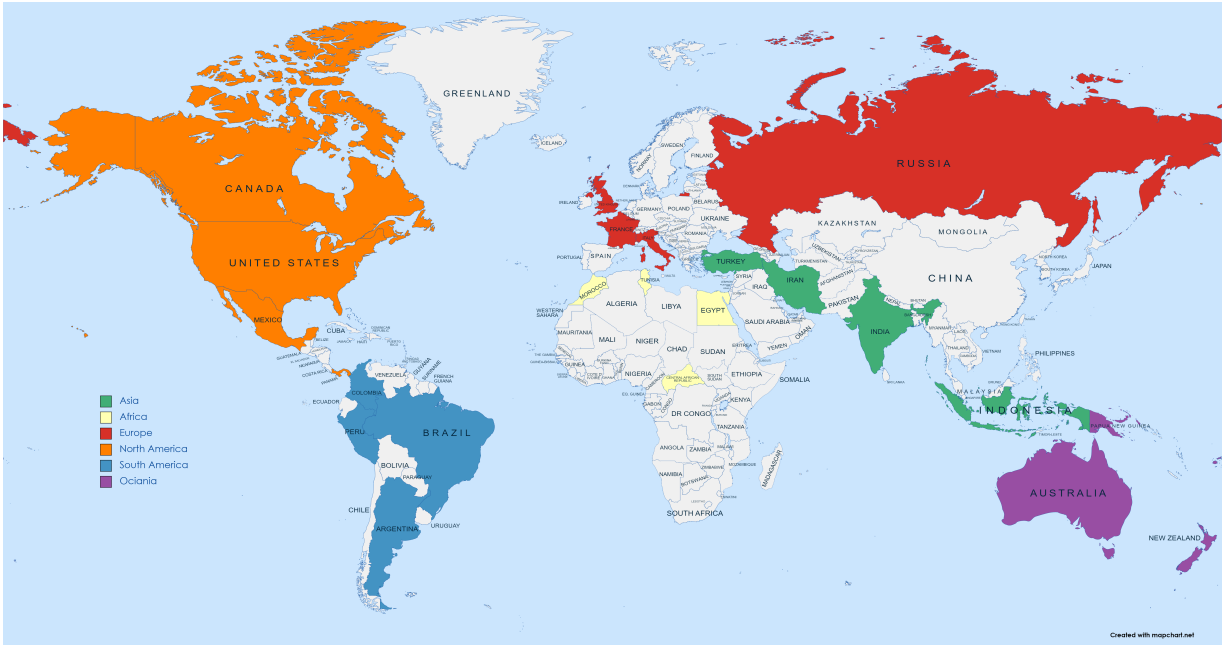

Figure 5: Map of area (all 24 selected countries in every 6 continents according to the legend)

Therefore, in this paper, we combine the properties of the CNN network with the LSTM network. To do this, we use the parallel connection of the network to achieve the CNN-LSTM network model, which takes advantage of the time and capacity of the net. Expression of the spatial characteristics of the two networks. Figure 6 shows the overall structure of proposed method. Firstly, the dataset is preprocessed, and the preprocessed data will be fed into the Forecasting Engine module. Later on, the feature information extracted from the CNN and the feature information extracted from the LSTM is processed in the same dimension, respectively, through the map layer. Moreover, the outputs of the CNN and LSTM are processed in parallel connection by concatenation. Finally, it is classified by the output layer activation function.

In addition, the proposed method uses the WOA optimization algorithm to fine-tune the hyperparameters of the CNN-LSTM model. Figure 3 shows the overall architecture of the proposed method. In the following subsections, each step is explained in more detail.

5.1. Preprocessing

Preprocessing involves cleansing and normalizing the dataset. To cleanse the dataset, all missing and defective values are replaced by the average value of the next day and the previous day of their respective column. Next, the values are normalized by the Min-Max Scaler method of Keras.

5.2. Forecasting Engine

As shown in Figure 6, this module is composed of a combination of CNN and LSTM. Firstly, the preprocessed input data is fed into the convolutional layer. The convolutional layer extracts the local features and makes the feature maps. The pooling layer receives the feature maps, reduces the dimensions, and finds the essential features. Indeed, this layer decreases the dimension and computation time. The proposed method utilizes Max Pooling which is common in dimension reduction.

The flattened layer receives the output of the pooling layer, and flats the data. Later on, the Repeat vector layer changes the dimension of data to be compatible with the LSTM layer. Then, the LSTM layer is utilized to capture the long-term dependencies. The output of LSTM is fed into the Dense layer to prepare the output.

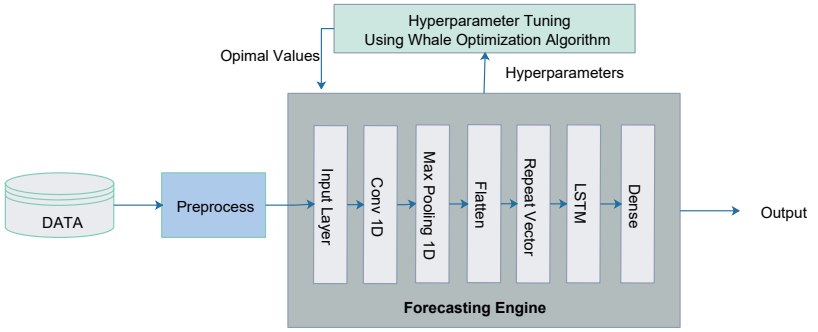

Figure 6: Overall structure of the proposed method

5.3. Hyperparameter Tuning Module

As described in Section 3.3, the performance of deep learning methods is highly dependent on the values of hyperparameters. Therefore, the proposed method uses the WOA algorithm to fine-tune the hyperparameters of the CNN-LSTM model such as the number of filters and kernel size of the convolutional layer, the pool size of the Max-pooling layer, and the number of units of the LSTM layer. We run the proposed method with the data related to each country individually and find the optimal values of hyperparameters by the WOA.

6. Results and Discussion

In this section, we evaluate the proposed method with the real-world dataset, described in Section 4. Addtionally, we compare it with other state-of-the-art methods such as CNN, RCLSTM, CNN-GRU, BILSTM, and ConvLSTM. This section includes Evaluation Setup and metrics, Hyperparameter tuning, and Comparison with other methods.

6.1. Evaluation Setup and Metrics

The proposed method was implemented in Python language, version 3.8, in the Google Colab[2] platform equipped with a K80 GPU and 12 GB of RAM. Additionally, we have employed the NiaPy [34] and Keras [35] libraries to implement the metaheuristic algorithms and deep learning models, respectively. Since the deep learning networks work based on random initialization and give different results at each run, each algorithm is run ten times, and we reported the average results. Moreover, we considered the batch size as 1 and the number of epochs as 100.

As mentioned earlier, the Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Mean Square Error (MSE), and R-square (R^2) methods are utilized to evaluate performance and predictive effects. The MAE calculation formula is shown in Equation 17. In Equation 17, y_i and x_i denote the predictive, and the true value, respectively.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - x_i| \tag{17}$$

The RMSE calculation formula is shown in Equation 18. In Equation 18, y_i and x_i denote the predictive, and the true value, respectively.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - x_i)^2} \tag{18}$$

[2]https://colab.research.google.com/

Table 1
Configuration settings of metaheuristic algorithms

Algorithm	Hyperparameter	Value
GWO	$\vec{a}$	Linearly decreased from 2 to 0
	$\vec{r}_1$	A random vector in [0,1]
	$\vec{r}_2$	A random vector in [0,1]
WOA	$\vec{a}$	Linearly decreased from 2 to 0
	$\vec{r}$	A random vector in [0,1]
GA	Crossover	Uniform crossover
	Mutation	Uniform mutation
	Crossover rate	0.25
	Mutation rate	0.25
FA	α	1
	γ	0.01
	β_0	1

The MSE calculation formula is shown in Equation 19. In Equation 19, y_i and x_i denote the predictive, and the true value, respectively.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - x_i)^2 \tag{19}$$

The R^2 calculation formula is shown in Equation 20. In Equation 20, y_i, x_i, and $\bar{y}_i$ indicate the predictive, the true, and the average value, respectively. The R^2's value range is (0, 1).

$$R^2 = 1 - \frac{(\sum_{i=1}^{n}(y_i - x_i)^2)/n}{(\sum_{i=1}^{n}(\bar{y}_i - x_i)^2)/n} \tag{20}$$

6.2. Hyperparameter Tuning

As mentioned in Section 5.3, the proposed method uses the WOA to fine-tune the hyperparameters of the CNN-LSTM model, such as the number of filters and kernel size of the convolutional layer, the pool size of the Max-pooling layer, and the number of units of the LSTM layer. Moreover, we have compared the results of WOA with the Genetic Algorithm (GA)[36], Grey Wolf Optimization (GWO) [37], and FireFly (FA) algorithms [38]. Table 1 shows the configuration settings of these algorithms as proposed in the NiaPy library.

We fine-tune the hyperparameters of the CNN-LSTM model for each country, which includes the number of filters (candidate values: 32 and 64), kernel size (candidate values: 3, 4, 5, 6, 7, and 8), pool size (candidate values: 2, 3, and 4), and the number of LSTM units (candidate values: 10, 15, 20, and 25). Table 2 shows the results of hyperparameter tuning. Table 2 The results of hyperparameter tuning (Tun: Tunisia, Egy: Egypt, Mor: Morocco, Afr: Africa, Ind: India, Indo: Indonesia, Ira: Iran, Tur: Turkey, Fra: France, Rus: Russia, Ita: Italy, Can: Canada, Mex: Mexico, Pan: Panama, Arg: Argentina, Bra: Brazil, Col: Colombia, Per: Peru, Aus: Australia, New: New Zealand, PNG: Papua New Guinea).

6.3. Comparison with other methods

In this section, firstly, we compare the results of the proposed method with other methods such as CNN, RCLSTM, BiLSTM, CNN-GRU, and ConvLSTM.

6.3.1. Africa

As shown in Table 3, in the case of infected and cured people, the proposed method, which uses the CNN-LSTM model, achieved the lowest loss in all countries compared to other models. In addition, regarding the dead people, in Egypt, Morocco, and Africa, the CNN-LSTM outperforms other methods in terms of loss, but the BiLSTM achieved a lower error in Tunisia. On the other hand, in Tunisia, the combined models performed better. Moreover, it can be seen that in Egypt, predicting the dead people has shown the lowest error, in some cases half of the other methods. Also, in Morocco, predicting the infected and cured people achieved a lower loss than the dead people. In Africa, the proposed method has reached the lowest error for predicting the infected, cured, and dead people.

6.3.2. Asia

Regarding Asia, we have forecasted the number of infected, cured, and dead people in India, Indonesia, Iran, and Turkey. As can be seen in Table 4, the proposed method obtained the lowest loss for forecasting the number of infected people in Indonesia, Iran, and Turkey, but in India, the CNN method showed the lowest error. In addition, the CNN-LSTM model forecasted the number of cured people in four Asian countries with the lowest errors. In the case of forecasting the dead people of Indonesia and Turkey, the CNN model achieved the lowest error. Still, the proposed method performed better than other models in India and Iran.

Also, it can be seen that for India, the CNN-GRU model performs better in forecasting the cured and dead people, but the CNN model achieves the lowest error in forecasting the infected people. For Indonesia and Turkey, the proposed method achieved the lowest loss in predicting the number of infected and cured people, but the CNN model outperformed the other models in forecasting the number of dead people. What is more, regarding Iran, the proposed method forecasted the number of infected, cured, and dead people with the lowest error.

6.3.3. Europe

Moreover, we have forecasted the number of infected, cured, and dead people in four European countries: France, Italy, Russia, and the UK. As listed in Table 5, in the case of forecasting the number of infected people, the proposed method achieved the lowest error for Italy and Russia. Also, the CNN and CNN-GRU models performed better than other models for France and UK, respectively.

Moreover, the proposed method achieved the lowest error for forecasting the number of cured people in Italy and Russia. The CNN model outperformed better than other models for France and UK. In the case of forecasting the dead people of Russia and the UK, the CNN-LSTM model showed the lowest error. Also, the CNN-GRU and BiLSTM models obtained the lowest error for France and Italy, respectively. For forecasting the number of infected and cured people in France, the CNN model performed better, and in forecasting the number of dead people, the CNN-GRU achieved the lowest loss. For forecasting the number of infected and cured people of Italy, the CNN-LSTM performed better than others.

Table 2
The results of hyperparameter tuning

	GWO				WOA				GA				FA			
Hyper parameters	Filters	Kernel Size	Pool Size	Units	Filters	Kernel Size	Pool Size	Units	Filters	Kernel Size	Pool Size	Units	Filters	Kernel Size	Pool Size	Units
Tun	32	3	2	20	32	4	2	20	32	4	2	20	64	4	2	15
Egy	32	3	2	20	32	4	2	20	32	3	2	20	64	4	2	15
Mor	32	3	2	20	32	4	2	20	32	4	2	20	64	4	2	15
Afr	32	3	2	20	32	4	2	20	32	3	2	20	64	4	2	15
Ind	64	8	2	20	32	8	2	20	32	4	2	20	32	4	2	25
Indo	32	3	2	20	32	4	2	20	32	3	2	20	32	4	2	15
Ira	32	3	2	20	32	4	2	20	32	3	2	20	32	4	2	15
Tur	32	3	2	20	32	4	2	20	32	3	2	20	32	4	2	15
Fra	32	3	4	15	32	4	2	15	32	3	2	15	32	4	2	15
Rus	64	8	2	20	32	8	2	20	32	4	2	20	64	4	2	25
Ita	32	4	2	20	32	3	2	20	32	4	2	20	32	4	2	15
UK	64	4	2	20	32	4	2	20	32	3	2	20	32	4	2	15
Can	32	4	2	20	32	3	2	20	32	4	2	20	32	4	2	15
Mex	32	4	2	20	32	4	2	20	32	4	2	20	32	4	2	15
Pan	32	4	2	20	32	3	2	20	32	4	2	20	32	4	2	15
US	64	8	2	20	32	8	2	20	32	4	2	20	64	4	2	25
Arg	32	3	2	20	32	3	2	20	32	4	2	15	32	4	2	15
Bra	32	3	2	20	32	3	2	20	32	4	2	20	32	4	2	15
Col	32	3	2	20	32	3	2	20	32	4	2	20	32	4	2	15
Per	32	3	2	20	32	3	2	20	32	4	2	20	32	4	2	15
Aus	32	4	2	20	32	3	2	20	32	4	2	20	64	4	2	15
Fiji	32	4	2	20	32	3	2	20	32	4	2	20	32	4	2	15
New	32	4	2	20	32	3	2	20	32	4	2	15	32	4	2	15
PNG	32	4	2	20	32	3	2	20	32	4	2	15	32	4	2	15

Table 3
Comparing forecasting results of African Countries

		confirmed				deaths				recovered			
	Model	LOSS	MSE	RMSE	MAE	LOSS	MSE	RMSE	MAE	LOSS	MSE	RMSE	MAE
Tunesia	CNN	0.00215	0.00215	0.18200	0.03104	0.00044	0.00044	0.01975	0.01704	0.00087	0.00087	0.02852	0.06672
	RCLSTM	0.00118	0.00118	0.02796	0.11443	0.00043	0.00043	0.02014	0.01591	0.00070	0.00070	0.02508	0.02151
	CNN-LSTM	**0.00011**	0.00011	0.00997	0.00769	**0.00035**	0.00035	0.01856	0.01536	0.00075	0.00075	0.02652	0.02331
	CNN-GRU	0.00083	0.00083	0.02774	0.01823	0.00044	0.00348	0.01943	0.01768	0.00041	0.00041	0.01997	0.01320
	BILSTM	0.00185	0.00185	0.03866	0.03200	0.00080	0.00080	0.02772	0.01964	**0.00034**	0.00034	0.01818	0.01168
	CONVLSTM	0.01537	0.00272	0.04750	0.16420	0.00591	0.00251	0.04056	0.03530	0.01217	0.00067	0.02488	0.01900
Egypt	CNN	0.00042	0.00042	0.01867	0.01443	0.00040	0.00040	0.01894	0.01519	0.00072	0.00072	0.02633	0.02115
	RCLSTM	0.00060	0.00060	0.02062	0.01690	0.00070	0.00070	0.02466	0.02034	0.00271	0.00271	0.04987	0.04066
	CNN-LSTM	**0.00006**	0.00006	0.00756	0.00628	**0.00007**	0.00007	0.00778	0.00644	**0.00004**	0.00004	0.00601	0.00390
	CNN-GRU	0.00033	0.00033	0.01853	0.01474	0.00068	0.00068	0.02577	0.01996	0.00308	0.00308	0.04245	0.03488
	BILSTM	0.00035	0.00035	0.01842	0.01427	0.00029	0.00029	0.01640	0.01182	0.00062	0.00062	0.02406	0.01821
	CONVLSTM	0.00096	0.00052	0.02128	0.01732	0.00140	0.00023	0.01473	0.01101	0.01549	0.00047	0.02137	0.01807
Morocco	CNN	0.00092	0.00092	0.03011	0.02201	0.00064	0.00064	0.02526	0.01860	0.00080	0.00080	0.02808	0.02243
	RCLSTM	0.00092	0.00092	0.02335	0.02223	0.00172	0.00172	0.03751	0.03166	0.00091	0.00091	0.02810	0.02324
	CNN-LSTM	**0.00038**	0.00038	0.01874	0.01467	**0.00038**	0.00038	0.01887	0.01498	**0.00055**	0.00055	0.02488	0.02016
	CNN-GRU	0.00327	0.00327	0.05286	0.04632	0.00076	0.00076	0.02729	0.01960	0.00149	0.00149	0.03544	0.02923
	BILSTM	0.00064	0.00064	0.02498	0.02206	0.00317	0.00317	0.10425	0.01449	0.00233	0.00233	0.04402	0.03413
	CONVLSTM	0.01319	0.00129	0.02027	0.01520	0.00911	0.00028	0.01664	0.01258	0.00947	0.00036	0.01887	0.01436
Africa	CNN	0.00030	0.00030	0.01702	0.01390	0.00038	0.00038	0.01897	0.01410	0.00075	0.00075	0.02660	0.02176
	RCLSTM	0.00324	0.00324	0.06202	0.05463	0.00623	0.00623	0.04832	0.03512	0.00441	0.00441	0.06579	0.05096
	CNN-LSTM	**0.00018**	0.00018	0.01265	0.01071	**0.00018**	0.00018	0.01318	0.02322	**0.00012**	0.00012	0.01008	0.00788
	CNN-GRU	0.00113	0.00113	0.03344	0.02735	0.00076	0.00076	0.02548	0.01766	0.00088	0.00088	0.02930	0.02129
	BILSTM	0.00145	0.00145	0.03630	0.03031	0.00055	0.00055	0.02178	0.01749	0.00086	0.00086	0.02778	0.02177
	CONVLSTM	0.00282	0.00210	0.02208	0.01649	0.00477	0.00065	0.02493	0.01688	0.00506	0.00092	0.02943	0.02089

Table 4
Comparing forecasting results of Asian countries

		confirmed				deaths				recovered			
	Model	LOSS	MSE	RMSE	MAE	LOSS	MSE	RMSE	MAE	LOSS	MSE	RMSE	MAE
India	CNN	0.00009	0.00009	0.00905	0.00743	0.00010	0.00010	0.00923	0.00826	0.00015	0.00015	0.01177	0.01055
	RCLSTM	0.00034	0.00034	0.01830	0.01515	0.00096	0.00096	0.02715	0.02611	0.00120	0.00120	0.03069	0.03149
	CNN-LSTM	0.00013	0.00013	0.01123	0.00938	0.00024	0.00024	0.01405	0.01303	0.00008	0.00008	0.00897	0.00705
	CNN-GRU	0.00011	0.00011	0.00895	0.00772	**0.00008**	0.00008	0.00841	0.00752	**0.00007**	0.00007	0.00802	0.00706
	BILSTM	**0.00006**	0.00006	0.00778	0.00644	0.00011	0.00011	0.01005	0.00880	0.00007	0.00007	0.00845	0.00760
	CONVLSTM	0.00758	0.00018	0.01122	0.00973	0.00343	0.00007	0.00769	0.00659	0.00166	0.00007	0.00800	0.00704
Indonesia	CNN	0.00032	0.00032	0.01688	0.01407	0.00039	0.00039	0.01912	0.01499	**0.00099**	0.00099	0.02986	0.02558
	RCLSTM	0.00115	0.00115	0.03210	0.02735	0.00423	0.00423	0.03204	0.02688	0.00260	0.00260	0.04688	0.03945
	CNN-LSTM	**0.00022**	0.00022	0.01452	0.01171	**0.00028**	0.00028	0.01549	0.01213	0.00486	0.00486	0.03039	0.02678
	CNN-GRU	0.00023	0.00023	0.01472	0.01073	0.00038	0.00038	0.01843	0.01467	0.00274	0.00274	0.04717	0.03617
	BILSTM	0.00072	0.00072	0.02657	0.01986	0.00030	0.00030	0.01662	0.01237	0.00579	0.00579	0.04161	0.03368
	CONVLSTM	0.00666	0.00029	0.01544	0.01109	0.00548	0.00048	0.01908	0.01469	0.00608	0.00056	0.02269	0.01915
Iran	CNN	0.00169	0.00169	0.04071	0.03363	0.00090	0.00090	0.02984	0.02341	0.00065	0.00065	0.02502	0.01779
	RCLSTM	0.00137	0.00137	0.03582	0.03329	0.00096	0.00096	0.02685	0.02480	0.00234	0.00234	0.04142	0.03637
	CNN-LSTM	**0.00032**	0.00032	0.01640	0.01331	0.00050	0.00050	0.01969	0.01566	0.00061	0.00061	0.02289	0.02046
	CNN-GRU	0.00136	0.00136	0.03369	0.02881	0.00152	0.00152	0.03771	0.03221	0.00108	0.00108	0.03172	0.02374
	BILSTM	0.00066	0.00066	0.02442	0.01872	**0.00034**	0.00034	0.01780	0.01510	**0.00051**	0.00051	0.02005	0.01730
	CONVLSTM	0.01215	0.00052	0.02208	0.01707	0.00656	0.00024	0.01485	0.01275	0.01003	0.00036	0.01763	0.01393
Turkey	CNN	0.03030	0.03030	0.17372	0.08486	0.00088	0.00088	0.02884	0.02397	**0.01386**	0.01386	0.11955	0.05846
	RCLSTM	0.03818	0.03818	0.19293	0.10607	0.00072	0.00072	0.02585	0.02055	0.04001	0.04001	0.19578	0.13023
	CNN-LSTM	**0.02560**	0.02560	0.15811	0.06857	**0.00046**	0.00046	0.02056	0.01471	0.01515	0.01515	0.12269	0.06791
	CNN-GRU	0.04451	0.04451	0.20503	0.11568	0.00102	0.00102	0.02649	0.02087	0.06989	0.06989	0.22274	0.15024
	BILSTM	0.02588	0.02588	0.16078	0.07575	0.00091	0.00091	0.02735	0.03475	0.02764	0.02764	0.16510	0.09447
	CONVLSTM	0.04757	0.02273	0.15076	0.06194	0.01109	0.00140	0.01780	0.01240	0.08929	0.07487	0.20893	0.13757

Table 5
Comparing forecasting results of European countries

		confirmed				deaths				recovered			
	Model	LOSS	MSE	RMSE	MAE	LOSS	MSE	RMSE	MAE	LOSS	MSE	RMSE	MAE
France	CNN	0.00072	0.00072	0.02666	0.01992	**0.00072**	0.00072	0.02662	0.02060	0.00143	0.00143	0.03779	0.03216
	RCLSTM	0.00109	0.00109	0.03198	0.02737	0.00167	0.00167	0.03927	0.03347	0.00933	0.00933	0.08550	0.07595
	CNN-LSTM	**0.00020**	0.00020	0.01353	0.01142	0.00077	0.00077	0.02765	0.02439	0.00595	0.00595	0.07367	0.06285
	CNN-GRU	0.00106	0.00106	0.03212	0.02427	0.00095	0.00095	0.03037	0.02347	**0.00132**	0.00132	0.03626	0.03059
	BILSTM	0.00078	0.00078	0.02717	0.02176	0.00154	0.00154	0.03852	0.03192	0.00165	0.00165	0.04030	0.03316
	CONVLSTM	0.00719	0.00227	0.04144	0.03464	0.00620	0.00054	0.02300	0.01792	0.00254	0.00204	0.04376	0.03710
Italy	CNN	0.00638	0.00638	0.04516	0.05037	0.00152	0.00152	0.03720	0.02996	0.00572	0.00572	0.06652	0.05209
	RCLSTM	0.00476	0.00476	0.06589	0.05268	0.00189	0.00189	0.04305	0.03695	0.00367	0.00367	0.05701	0.04549
	CNN-LSTM	**0.00062**	0.00062	0.02415	0.01780	**0.00021**	0.00021	0.01362	0.01032	0.00141	0.00141	0.03577	0.02864
	CNN-GRU	0.00537	0.00537	0.05118	0.03819	0.00334	0.00334	0.05249	0.04193	0.00185	0.00185	0.04282	0.03296
	BILSTM	0.00939	0.00939	0.09308	0.07156	0.00351	0.00351	0.05497	0.04475	**0.00134**	0.00134	0.03650	0.02666
	CONVLSTM	0.03101	0.00304	0.05065	0.03778	0.01493	0.00195	0.04216	0.03301	0.01809	0.00276	0.05112	0.03935
Russia	CNN	0.00079	0.00079	0.02695	0.01916	0.00165	0.00165	0.03759	0.02733	0.00169	0.00169	0.03920	0.02793
	RCLSTM	0.00084	0.00084	0.02506	0.02151	0.00104	0.00104	0.02892	0.02581	0.00104	0.00104	0.02777	0.02165
	CNN-LSTM	**0.00008**	0.00008	0.00838	0.00747	**0.00012**	0.00012	0.01012	0.00882	**0.00009**	0.00009	0.00898	0.00743
	CNN-GRU	0.00117	0.00117	0.03295	0.02518	0.00122	0.00122	0.03486	0.02316	0.00069	0.00069	0.02594	0.01739
	BILSTM	0.00159	0.00159	0.03805	0.02778	0.00064	0.00064	0.02496	0.01770	0.00102	0.00102	0.02767	0.02249
	CONVLSTM	0.00878	0.00078	0.02566	0.01918	0.00918	0.00017	0.01281	0.00989	0.01306	0.00166	0.03819	0.02578
UK	CNN	0.00124	0.00124	0.03403	0.02333	0.00307	0.00307	0.05135	0.04409	0.00435	0.00435	0.06130	0.04275
	RCLSTM	0.00100	0.00100	0.03056	0.02205	0.00644	0.00644	0.07962	0.07094	0.00144	0.00144	0.03699	0.02945
	CNN-LSTM	0.00207	0.00207	0.04475	0.03036	**0.00107**	0.00107	0.03245	0.02760	**0.00020**	0.00020	0.01310	0.01075
	CNN-GRU	**0.00059**	0.00059	0.02384	0.01804	0.01625	0.01625	0.10611	0.09402	0.00270	0.00270	0.04978	0.03489
	BILSTM	0.00097	0.00097	0.02955	0.02086	0.00385	0.00385	0.05980	0.05294	0.00463	0.00463	0.06330	0.04746
	CONVLSTM	0.01087	0.00109	0.03257	0.02016	0.00767	0.00357	0.05860	0.05195	0.00987	0.00169	0.04017	0.02584

6.3.4. North America

We have investigated the data from Canada, Mexico, Panama, and the US in America. Table 6 shows the forecasted results. As it is obvious in Table 6, for forecasting the infected and dead people, the CNN-LSTM model and the cured people with the BiLSTM model have achieved the lowest loss (0.0000804, 0.000836, 0.000066).

Regarding Canada, Panama, and the US, the CNN-LSTM outperformed the other models for forecasting the number of infected people (0.0001094, 0.0000804, 0.0002446). Also, to forecast the number of infected, cured, and dead people in Mexico, the BiLSTM method achieved the lowest loss (0.000066). For all four countries, the CNN model achieved the lowest loss. Totally the BiLSTM model showed better performance while forecasting the cured people of Mexico (0.000066).

6.3.5. South America

In South America, the infected, cured, and dead people of Argentina, Brazil, Colombia, and Peru are forecasted, and the results are shown in Table 7. As listed in Table 5, in the case of infected and cured people, the proposed method showed the lowest loss (0.000107, 0.000028) when forecasting the data of Peru and Brazil, respectively. Regarding the dead people, the CNN-LSTM has achieved the lowest results when forecasting the data for Brazil (0.0000556).

On the other hand, in the case of Argentina, the proposed method has achieved the lowest loss for forecasting the number of infected and cured people (0.000563, 0.0001626), but for forecasting the number of dead people, the BiLSTM model performed better (0.0000924). Also, the CNN-LSTM forecasted the number of cured and dead people in Brazil with the lowest loss (0.000028,0.0000556). The BiLSTM model achieved the lowest loss in forecasting the number of infected people in Brazil (0.0000421). For forecasting the number of infected and cured people in Colombia, the proposed method outperformed the other models (0.000159,0.000826), but the RCLSTM model performed better in the case of dead people (0.001186). Regarding the number of infected and dead people in Peru, the CNN-LSTM model has achieved the lowest loss (0.000107, 0.000072), but for forecasting the number of cured people, the CNN-GRU method outperformed the other models (0.000064). Generally, the CNN-LSTM model has achieved the best result in forecasting the number of dead people in Brazil (0.0000556).

6.3.6. Oceania

In Oceania, we have forecasted the number of infected, cured, and dead people in Australia, Fiji, New Zealand, and Papua New Guinea. Table 8 shows the evaluation results.

For forecasting the number of infected and dead people, the proposed method outperformed other New Zealand and Australian models, respectively (0.000083, 0.00002425). Regarding the cured people, the BiLSTM model performed better (0.00006929) in Australia.

The BiLSTM model has achieved the lowest loss for forecasting the number of infected and cured people in Australia (0.000228,0.000069). The CNN-LSTM performed better than other methods for forecasting the number of dead people in Australia (0.000024). In the case of forecasting the number of infected and cured people of Fiji, the CNN-LSTM has achieved the lowest loss (0.00136,0.0001159), but the RCLSTM model outperformed the other models in forecasting the number of dead people (0.001676). Regarding New Zealand, the proposed method has shown the lowest loss in forecasting the number of infected people (0.000083). The ConvLSTM model outperformed the other methods for forecasting the number of infected people (0.0001974), and for forecasting the number of dead people, the CNN-GRU has shown the lowest loss (0.0001046). The proposed method has achieved the lowest loss in forecasting the number of infected, cured, and dead people (0.00136, 0.002173, 0.000552).

The proposed method generally performed better than other methods in forecasting the number of dead people in Australia (0.00002425).

6.3.7. Comparison of continents

In addition, we have evaluated the optimization of the proposed method on the data of six continents by the whale algorithm. As listed in Table 9, for forecasting the number of infected, cured, and dead people, the proposed method has achieved the lowest results in Africa, South America, and North America (0.00018235, 0.00013915, 0.00033195) and the results were well-matched by The whale meta-heuristic algorithm has been improved.

Table 6
Comparing forecasting results of North American countries

		confirmed				deaths				recovered			
	Model	LOSS	MSE	RMSE	MAE	LOSS	MSE	RMSE	MAE	LOSS	MSE	RMSE	MAE
Canada	CNN	0.00215	0.00215	0.18200	0.03104	0.00044	0.00044	0.01975	0.01704	0.00087	0.00087	0.02852	0.06672
	RCLSTM	0.00118	0.00118	0.02796	0.11443	0.00043	0.00043	0.02014	0.01591	0.00070	0.00070	0.02508	0.02151
	CNN-LSTM	**0.00011**	0.00011	0.00997	0.00769	**0.00035**	0.00035	0.01856	0.01536	0.00075	0.00075	0.02652	0.02331
	CNN-GRU	0.00083	0.00083	0.02774	0.01823	0.00044	0.00348	0.01943	0.01768	0.00041	0.00041	0.01997	0.01320
	BILSTM	0.00185	0.00185	0.03866	0.03200	0.00080	0.00080	0.02772	0.01964	**0.00034**	0.00034	0.01818	0.01168
	CONVLSTM	0.01537	0.00272	0.04750	0.16420	0.00591	0.00251	0.04056	0.03530	0.01217	0.00067	0.02488	0.01900
Mexico	CNN	0.00125	0.00125	0.03492	0.02647	0.00009	0.00009	0.00918	0.00721	0.00015	0.00015	0.01209	0.00962
	RCLSTM	0.00124	0.00124	0.03514	0.02717	0.00071	0.00071	0.02397	0.02184	0.00102	0.00102	0.02629	0.02395
	CNN-LSTM	**0.00069**	0.00069	0.02279	0.01263	0.00007	0.00007	0.00792	0.00703	**0.00008**	0.00008	0.00887	0.00712
	CNN-GRU	0.00215	0.00215	0.04471	0.03355	0.00059	0.00059	0.02130	0.01645	0.00044	0.00044	0.02010	0.01558
	BILSTM	0.00115	0.00115	0.03382	0.02290	**0.00007**	0.00007	0.00796	0.00625	0.00039	0.00039	0.01912	0.01381
	CONVLSTM	0.00432	0.00142	0.03769	0.02478	0.00259	0.00005	0.00724	0.00575	0.00462	0.00014	0.01175	0.00941
Panama	CNN	0.00033	0.00033	0.01742	0.01495	0.00108	0.00108	0.02947	0.02175	0.00037	0.00037	0.01929	0.01440
	RCLSTM	0.00145	0.00145	0.03393	0.03031	0.00273	0.00273	0.04780	0.04200	0.00368	0.00368	0.05427	0.04615
	CNN-LSTM	**0.00008**	0.00008	0.00858	0.00677	**0.00015**	0.00015	0.01187	0.00998	**0.00025**	0.00025	0.01352	0.01159
	CNN-GRU	0.00173	0.00173	0.03275	0.02378	0.00050	0.00050	0.02183	0.01505	0.00233	0.00233	0.04456	0.03520
	BILSTM	0.00033	0.00033	0.01737	0.01350	0.00024	0.00024	0.01537	0.01100	0.00042	0.00042	0.01983	0.03986
	CONVLSTM	0.01919	0.00103	0.02607	0.02393	0.00499	0.00026	0.01594	0.01155	0.00852	0.00039	0.01938	0.01517
US	CNN	0.00039	0.00039	0.01821	0.01573	0.00026	0.00026	0.01561	0.01307	0.00040	0.00040	0.01880	0.03083
	RCLSTM	0.00213	0.00213	0.04483	0.04209	0.00114	0.00114	0.02875	0.02582	0.00193	0.00193	0.04176	0.03449
	CNN-LSTM	**0.00024**	0.00024	0.01508	0.01226	**0.00052**	0.00052	0.02171	0.01866	**0.00025**	0.00025	0.01531	0.02718
	CNN-GRU	0.00203	0.00203	0.04183	0.03327	0.00137	0.00137	0.03610	0.03066	0.00099	0.00099	0.03071	0.02084
	BILSTM	0.00426	0.00426	0.05845	0.04026	0.00068	0.00068	0.02317	0.01910	0.00411	0.00411	0.02912	0.02209
	CONVLSTM	0.01134	0.00067	0.02441	0.01914	0.00413	0.00039	0.01906	0.01563	0.00771	0.00041	0.01900	0.01485

Table 7
Comparing forecasting results of South American Countries

		confirmed				deaths				recovered			
	Model	LOSS	MSE	RMSE	MAE	LOSS	MSE	RMSE	MAE	LOSS	MSE	RMSE	MAE
Argentina	CNN	0.00097	0.00097	0.02779	0.02378	0.00255	0.00255	0.04117	0.03353	0.00025	0.00025	0.01507	0.01230
	RCLSTM	0.00081	0.00081	0.02732	0.02646	0.00250	0.00250	0.04392	0.04157	**0.00009**	0.00009	0.00946	0.00758
	CNN-LSTM	**0.00056**	0.00056	0.02318	0.01744	**0.00016**	0.00016	0.01235	0.01052	0.00026	0.00026	0.01466	0.01267
	CNN-GRU	0.00069	0.00069	0.02382	0.01843	0.00046	0.00046	0.02043	0.01792	0.00031	0.00031	0.01649	0.01378
	BILSTM	0.00227	0.00227	0.04221	0.04006	0.00207	0.00207	0.03898	0.03523	0.00025	0.00025	0.01527	0.01191
	CONVLSTM	0.00412	0.00050	0.02092	0.01756	0.00289	0.00073	0.02325	0.02064	0.00244	0.00040	0.01817	0.01507
Brazil	CNN	0.00049	0.00049	0.02177	0.01788	0.00019	0.00019	0.01201	0.00970	0.00028	0.00028	0.01652	0.01342
	RCLSTM	0.00075	0.00075	0.02593	0.02209	0.00031	0.00031	0.01721	0.01501	0.00182	0.00182	0.03668	0.03413
	CNN-LSTM	0.00045	0.00045	0.02064	0.01698	**0.00003**	0.00003	0.00499	0.00436	**0.00006**	0.00006	0.00719	0.00588
	CNN-GRU	0.00048	0.00048	0.02133	0.01724	0.00027	0.00027	0.01573	0.01336	0.00078	0.00078	0.02699	0.02270
	BILSTM	**0.00042**	0.00042	0.01980	0.01575	0.00029	0.00029	0.01699	0.01414	0.00060	0.00060	0.02421	0.02117
	CONVLSTM	0.00293	0.00040	0.01865	0.01570	0.00135	0.00021	0.01399	0.01188	0.00349	0.00029	0.01656	0.01251
Colombia	CNN	0.00023	0.00023	0.01487	0.01141	0.00012	0.00012	0.01045	0.02612	0.00133	0.00133	0.03358	0.02839
	RCLSTM	0.00059	0.00059	0.02014	0.01758	0.00184	0.00184	0.03679	0.03386	**0.00089**	0.00089	0.02837	0.02248
	CNN-LSTM	**0.00016**	0.00016	0.01251	0.00889	**0.00009**	0.00009	0.00919	0.00686	0.00119	0.00119	0.03234	0.02796
	CNN-GRU	0.00032	0.00032	0.01747	0.01895	0.00012	0.00012	0.00991	0.00826	0.00666	0.00666	0.06757	0.06388
	BILSTM	0.00093	0.00093	0.02573	0.02223	0.00042	0.00042	0.01813	0.01568	0.00354	0.00354	0.05582	0.05068
	CONVLSTM	0.00514	0.00035	0.01746	0.01458	0.00782	0.00020	0.01318	0.01147	0.00456	0.00075	0.02686	0.02035
Peru	CNN	0.00023	0.00023	0.01291	0.01226	0.00008	0.00008	0.00829	0.00716	0.00028	0.00028	0.01485	0.01382
	RCLSTM	0.00064	0.00064	0.02325	0.02204	0.00053	0.00053	0.01902	0.01838	0.00087	0.00087	0.02635	0.02493
	CNN-LSTM	**0.00011**	0.00011	0.00950	0.00796	0.00028	0.00028	0.01617	0.01435	**0.00007**	0.00007	0.00745	0.00649
	CNN-GRU	0.00019	0.00019	0.01238	0.01102	**0.00006**	0.00006	0.00757	0.00618	0.00021	0.00021	0.01414	0.01290
	BILSTM	0.00022	0.00022	0.01325	0.01269	0.00043	0.00043	0.01704	0.01499	0.00014	0.00014	0.01051	0.00941
	CONVLSTM	0.00070	0.00007	0.00841	0.00704	0.00058	0.00033	0.01760	0.01653	0.00050	0.00011	0.00986	0.00944

Table 8
Comparing forecasting results of Oceania Countries

		confirmed				deaths				recovered			
	Model	LOSS	MSE	RMSE	MAE	LOSS	MSE	RMSE	MAE	LOSS	MSE	RMSE	MAE
Australia	CNN	0.00049	0.00049	0.01999	0.01893	0.00016	0.00016	0.01074	0.02063	0.00015	0.00015	0.01188	0.01094
	RCLSTM	0.00041	0.00041	0.01856	0.01807	0.00066	0.00066	0.02166	0.02011	0.00022	0.00022	0.01231	0.01111
	CNN-LSTM	0.00044	0.00044	0.01946	0.01827	0.00024	0.00024	0.01441	0.01312	**0.00002**	0.00002	0.00447	0.00367
	CNN-GRU	0.00033	0.00033	0.01579	0.01370	0.00032	0.00032	0.01390	0.01279	0.00015	0.00015	0.01107	0.00959
	BILSTM	**0.00023**	0.00023	0.01304	0.01230	**0.00007**	0.00007	0.00801	0.00735	0.00030	0.00030	0.03697	0.01284
	CONVLSTM	0.00024	0.00023	0.01452	0.01314	0.00016	0.00016	0.01034	0.00974	0.00005	0.00005	0.00677	0.00573
Fiji	CNN	0.00167	0.00167	0.04085	0.03112	0.00050	0.00050	0.01845	0.01667	0.00207	0.00207	0.04491	0.03742
	RCLSTM	0.00230	0.00230	0.04716	0.03147	0.00153	0.00153	0.03575	0.03473	**0.00167**	0.00167	0.04081	0.03115
	CNN-LSTM	**0.00136**	0.00136	0.03672	0.02404	**0.00012**	0.00012	0.01003	0.00825	0.00212	0.00212	0.04507	0.03444
	CNN-GRU	0.00200	0.00200	0.04355	0.03200	0.00063	0.00063	0.01911	0.01663	0.00180	0.00180	0.04200	0.03107
	BILSTM	0.00170	0.00170	0.04111	0.02709	0.00036	0.00036	0.01782	0.01623	0.00326	0.00326	0.05599	0.04603
	CONVLSTM	0.00528	0.00514	0.03722	0.02552	0.00076	0.00076	0.02410	0.02272	0.00777	0.00147	0.03832	0.02937
Newzealand	CNN	0.00019	0.00019	0.01261	0.01107	0.00054	0.00054	0.01910	0.01795	0.00011	0.00011	0.00957	0.00747
	RCLSTM	0.00010	0.00010	0.00881	0.01526	0.00061	0.00061	0.02071	0.02053	0.00028	0.00028	0.01528	0.01454
	CNN-LSTM	**0.00008**	0.00008	0.00866	0.00739	0.00034	0.00034	0.01653	0.01519	0.00004	0.00004	0.00609	0.01153
	CNN-GRU	0.00015	0.00015	0.01155	0.00962	0.00091	0.00091	0.01655	0.02514	**0.00010**	0.00010	0.00960	0.00862
	BILSTM	0.00029	0.00029	0.01549	0.01411	0.00026	0.00026	0.01577	0.01512	0.00027	0.00027	0.01499	0.01431
	CONVLSTM	0.00047	0.00008	0.00877	0.00741	**0.00020**	0.00020	0.01190	0.01124	0.00048	0.00009	0.00895	0.00815
Papua New Guinea	CNN	0.00235	0.00235	0.04659	0.03129	0.00353	0.00353	0.05892	0.03952	0.00156	0.00156	0.03882	0.03230
	RCLSTM	0.00198	0.00198	0.04356	0.02917	0.00314	0.00314	0.05557	0.04109	0.00355	0.00355	0.05838	0.04961
	CNN-LSTM	**0.00136**	0.00136	0.03666	0.02826	**0.00217**	0.00217	0.04646	0.03858	**0.00056**	0.00056	0.02350	0.01822
	CNN-GRU	0.00151	0.00151	0.03815	0.02103	0.00295	0.00295	0.05318	0.03883	0.00272	0.00272	0.05153	0.04266
	BILSTM	0.00162	0.00162	0.03922	0.02521	0.00293	0.00293	0.05386	0.03937	0.00314	0.00314	0.05522	0.04748
	CONVLSTM	0.00330	0.00185	0.04254	0.02741	0.00607	0.00339	0.05619	0.04002	0.00398	0.00162	0.03963	0.03243

Table 9
Comparison optimized forecasting of the proposed method in all 24 countries on six continents

	confirmed				deaths				recovered			
	Loss	MSE	RMSE	MAE	Loss	MSE	RMSE	MAE	Loss	MSE	RMSE	MAE
Africa	0.00011	0.00011	0.00997	0.00769	0.00035	0.00035	0.01856	0.01536	0.00075	0.00075	0.02652	0.02331
	0.00006	0.00006	0.00756	0.00628	0.00007	0.00007	0.00778	0.00644	0.00004	0.00004	0.00601	0.00390
	0.00038	0.00038	0.01874	0.01467	0.00038	0.00038	0.01887	0.01498	0.00055	0.00055	0.02488	0.02016
	0.00018	0.00018	0.01265	0.01071	0.00018	0.00018	0.01318	0.02322	0.00012	0.00012	0.01008	0.00788
Asia	0.00013	0.00013	0.01123	0.00938	0.00024	0.00024	0.01405	0.01303	0.00008	0.00008	0.00897	0.00705
	0.00022	0.00022	0.01452	0.01171	0.00028	0.00028	0.01549	0.01213	0.00486	0.00486	0.03039	0.02678
	0.00032	0.00032	0.01640	0.01331	0.00050	0.00050	0.01969	0.01566	0.00061	0.00061	0.02289	0.02046
	0.02560	0.02560	0.15811	0.06857	0.00046	0.00046	0.02056	0.01471	0.01515	0.01515	0.12269	0.06791
Europe	0.00020	0.00020	0.01353	0.01142	0.00077	0.00077	0.02765	0.02439	0.00595	0.00595	0.07367	0.06285
	0.00062	0.00062	0.02415	0.01780	0.00021	0.00021	0.01362	0.01032	0.00141	0.00141	0.03577	0.02864
	0.00008	0.00008	0.00838	0.00747	0.00012	0.00012	0.01012	0.00882	0.00009	0.00009	0.00898	0.00743
	0.00207	0.00207	0.04475	0.03036	0.00107	0.00107	0.03245	0.02760	0.00020	0.00020	0.01310	0.01075
North America	0.00011	0.00011	0.00997	0.00769	0.00035	0.00035	0.01856	0.01536	0.00075	0.00075	0.02652	0.02331
	0.00069	0.00069	0.02279	0.01263	0.00007	0.00007	0.00792	0.00703	0.00008	0.00008	0.00887	0.00712
	0.00008	0.00008	0.00858	0.00677	0.00015	0.00015	0.01187	0.00998	0.00025	0.00025	0.01352	0.01159
	0.00024	0.00024	0.01508	0.01226	0.00052	0.00052	0.02171	0.01866	0.00025	0.00025	0.01531	0.02718
South America	0.00056	0.00056	0.02318	0.01744	0.00016	0.00016	0.01235	0.01052	0.00026	0.00026	0.01466	0.01267
	0.00045	0.00045	0.02064	0.01698	0.00003	0.00003	0.00499	0.00436	0.00006	0.00006	0.00719	0.00588
	0.00016	0.00016	0.01251	0.00889	0.00009	0.00009	0.00919	0.00686	0.00119	0.00119	0.03234	0.02796
	0.00011	0.00011	0.00950	0.00796	0.00028	0.00028	0.01617	0.01435	0.00007	0.00007	0.00745	0.00649
Oceania	0.00044	0.00044	0.01946	0.01827	0.00024	0.00024	0.01441	0.01312	0.00002	0.00002	0.00447	0.00367
	0.00136	0.00136	0.03672	0.02404	0.00012	0.00012	0.01003	0.00825	0.00212	0.00212	0.04507	0.03444
	0.00008	0.00008	0.00866	0.00739	0.00034	0.00034	0.01653	0.01519	0.00004	0.00004	0.00609	0.01153
	0.00136	0.00136	0.03666	0.02826	0.00217	0.00217	0.04646	0.03858	0.00056	0.00056	0.02350	0.01822

7. Conclusion

In this paper, we proposed a prediction model composed of a CNN and LSTM network to forecast the number of infected, cured, and dead people in 24 countries from six continents. CNNs are widely used to extract local and deep features. They utilize the convolutional layers to select features and the pooling layers to reduce dimensions. The LSTM models are a kind of RNN networks that capture long-term dependencies. These networks are appropriate for modeling sequential data such as time series. We combine the CNN and LSTM models to enhance the accuracy of short-term prediction.

Moreover, we have used the WOA algorithm to find the optimal values of hyperparameters of the proposed model, such as the number of filters, kernel size, pool size, and the number of LSTM units. We have fine-tuned the proposed model for each country separately and compared the results of WOA with GA, GWO, and FA algorithms.

Additionally, we compared the proposed method with other models, such as CNN, RCLSTM, BiLSTM, CNN-GRU, and ConvLSTM. Approximately in all experiments, the proposed method outperforms other models. Furthermore, we compared the proposed model based on the data of continents. The proposed method has achieved the lowest errors in Africa, South America, and North America.

For the future, to enhance the proposed method, hybrid optimization algorithms, such as GWO-WOA or FA-BAT, can be employed to fine-tune the hyperparameters. In addition, the impact of external data, such as age, location, and sex, needs to be investigated to improve the performance of the proposed method.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Abdelhafid Zeroual, Fouzi Harrou, Abdelkader Dairi, and Ying Sun. Deep learning methods for forecasting covid-19 time-series data: A comparative study. *Chaos, Solitons & Fractals*, 140:110121, 2020.

[2] Linchao Li, Jiasong Zhu, Hailong Zhang, Huachun Tan, Bowen Du, and Bin Ran. Coupled application of generative adversarial networks and conventional neural networks for travel mode detection using gps data. *Transportation Research Part A: Policy and Practice*, 136:282–292, 2020.

[3] Debanjan Parbat and Monisha Chakraborty. A python based support vector regression model for prediction of covid19 cases in india. *Chaos, Solitons & Fractals*, 138:109942, 2020.

[4] Peipei Wang, Xinqi Zheng, Jiayang Li, and Bangren Zhu. Prediction of epidemic trends in covid-19 with logistic model and machine learning technics. *Chaos, Solitons & Fractals*, 139:110058, 2020.

[5] Massimo A Achterberg, Bastian Prasse, Long Ma, Stojan Trajanovski, Maksim Kitsak, and Piet Van Mieghem. Comparing the accuracy of several network-based covid-19 prediction algorithms. *International journal of forecasting*, 2020.

[6] Sha He, Sanyi Tang, and Libin Rong. A discrete stochastic model of the covid-19 outbreak: Forecast and control. *Mathematical biosciences and engineering : MBE*, 17 4:2792–2804, 2020.

[7] Adam J Kucharski, Timothy W Russell, Charlie Diamond, Yang Liu, John Edmunds, Sebastian Funk, Rosalind M Eggo, Fiona Sun, Mark Jit, James D Munday, et al. Early dynamics of transmission and control of covid-19: a mathematical modelling study. *The lancet infectious diseases*, 20(5):553–558, 2020.

[8] Joseph T Wu, Kathy Leung, and Gabriel M Leung. Nowcasting and forecasting the potential domestic and international spread of the 2019-ncov outbreak originating in wuhan, china: a modelling study. *The Lancet*, 395(10225):689–697, 2020.

[9] Zian Zhuang, Shi Zhao, Qianying Lin, Peihua Cao, Yijun Lou, Lin Yang, and Daihai He. Preliminary estimation of the novel coronavirus disease (covid-19) cases in iran: A modelling analysis based on overseas cases and air travel data. *International Journal of Infectious Diseases*, 94:29–31, 2020.

[10] Luuk D Collou, GWJ Bruinsma, and MJ van Riemsdijk. Digitalization of hrm: designing a simulation model for hr decision making: The development of a simulation for the alignment of hr-practices. 2019.

[11] Rajan Gupta and Saibal K Pal. Trend analysis and forecasting of covid-19 outbreak in india. *MedRxiv*, 2020.

[12] Mossamet Kamrun Nesa, Md Rashed Babu, and Mohammad Tareq Mamun Khan. Forecasting covid-19 situation in bangladesh. *Biosafety and Health*, 4(1):6–10, 2022.

[13] Sumit Mohan, Anil Kumar Solanki, Harish Kumar Taluja, Anuj Singh, et al. Predicting the impact of the third wave of covid-19 in india using hybrid statistical machine learning models: A time series forecasting and sentiment analysis approach. *Computers in Biology and Medicine*, 144:105354, 2022.

[14] Shashank Reddy Vadyala, Sai Nethra Betgeri, Eric A Sherer, and Amod Amritphale. Prediction of the number of covid-19 confirmed cases based on k-means-lstm. *Array*, 11:100085, 2021.

[15] Parul Arora, Himanshu Kumar, and Bijaya Ketan Panigrahi. Prediction and analysis of covid-19 positive cases using deep learning models: A descriptive case study of india. *Chaos, Solitons & Fractals*, 139:110017, 2020.

[16] Rahele Kafieh, Narges Saeedizadeh, Roya Arian, Zahra Amini, Nasim Dadashi Serej, Atefeh Vaezi, and Shaghayegh Haghjooy Javanmard. Isfahan and covid-19: Deep spatiotemporal representation. *Chaos, Solitons & Fractals*, 141:110339, 2020.

[17] Hossein Abbasimehr and Reza Paki. Prediction of covid-19 confirmed cases combining deep learning methods and bayesian optimization. *Chaos, Solitons & Fractals*, 142:110511, 2021.

[18] Matheus Henrique Dal Molin Ribeiro, Ramon Gomes da Silva, Viviana Cocco Mariani, and Leandro dos Santos Coelho. Short-term forecasting covid-19 cumulative confirmed cases: Perspectives for brazil. *Chaos, Solitons & Fractals*, 135:109853, 2020.

[19] Kathleen CM de Carvalho, João Paulo Vicente, and João Paulo Teixeira. Covid-19 time series forecasting–twenty days ahead. *Procedia computer science*, 196:1021–1027, 2022.

[20] Joseph Galasso, Duy M Cao, and Robert Hochberg. A random forest model for forecasting regional covid-19 cases utilizing reproduction number estimates and demographic data. *Chaos, Solitons & Fractals*, 156:111779, 2022.

[21] Lu Xu, Rishikesh Magar, and Amir Barati Farimani. Forecasting covid-19 new cases using deep learning methods. *Computers in biology and medicine*, 144:105342, 2022.

[22] Binggui Zhou, Guanghua Yang, Zheng Shi, and Shaodan Ma. Interpretable temporal attention network for covid-19 forecasting. *Applied Soft Computing*, 120:108691, 2022.

[23] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[24] Huaizhi Wang, Haiyan Yi, Jianchun Peng, Guibin Wang, Yitao Liu, Hui Jiang, and Wenxin Liu. Deterministic and probabilistic forecasting of photovoltaic power based on deep convolutional neural network. *Energy conversion and management*, 153:409–422, 2017.

[25] Shunyu Yao, Yi-Peng Xu, and Ehsan Ramezani. Optimal long-term prediction of taiwan's transport energy by convolutional neural network and wildebeest herd optimizer. *Energy Reports*, 7:218–227, 2021.

[26] Shikhar Srivastava and Stefan Lessmann. A comparative study of lstm neural networks in forecasting day-ahead global horizontal irradiance with satellite data. *Solar Energy*, 162:232–247, 2018.

[27] Min-Seung Ko, Kwangsuk Lee, Jae-Kyeong Kim, Chang Woo Hong, Zhao Yang Dong, and Kyeon Hur. Deep concatenated residual network with bidirectional lstm for one-hour-ahead wind power forecasting. *IEEE Transactions on Sustainable Energy*, 12(2):1321–1335, 2020.

[28] Meng Zhang, Jiazheng Li, Yang Li, and Runnan Xu. Deep learning for short-term voltage stability assessment of power systems. *IEEE Access*, 9:29711–29718, 2021.

[29] Wei Wang, Tianzhen Hong, Xiaodong Xu, Jiayu Chen, Ziang Liu, and Ning Xu. Forecasting district-scale energy dynamics through integrating building network and long short-term memory learning algorithm. *Applied Energy*, 248:217–230, 2019.

[30] Mousa Alizadeh, Mohammad TH Beheshti, Amin Ramezani, and Hadis Saadatinezhad. Network traffic forecasting based on fixed telecommunication data using deep learning. In *2020 6th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS)*, pages 1–7. IEEE, 2020.

[31] Mousa Alizadeh, Sadegh E Mousavi, Mohammad TH Beheshti, and Ali Ostadi. Combination of feature selection and hybrid classifier as to network intrusion detection system adopting fa, gwo, and bat optimizers. In *2021 7th International Conference on Signal Processing and Intelligent Systems (ICSPIS)*, pages 1–7. IEEE, 2021.

[32] Seyedali Mirjalili and Andrew Lewis. The whale optimization algorithm. *Advances in engineering software*, 95:51–67, 2016.

[33] Xifeng Guo, Qiannan Zhao, Di Zheng, Yi Ning, and Ye Gao. A short-term load forecasting model of multi-scale cnn-lstm hybrid neural network considering the real-time electricity price. *Energy Reports*, 6:1046–1053, 2020.

[34] Grega Vrbančič, Lucija Brezočnik, Uroš Mlakar, Dušan Fister, and Iztok Fister Jr. NiaPy: Python microframework for building nature-inspired algorithms. *Journal of Open Source Software*, 3, 2018.

[35] François Chollet et al. Keras. `https://keras.io`, 2015.

[36] Mobina Mousapour Mamoudan, Zahra Mohammadnazari, Ali Ostadi, and Ali Esfahbodi. Food products pricing theory with application of machine learning and game theory approach. *International Journal of Production Research*, pages 1–21, 2022.

[37] Seyedali Mirjalili, Seyed Mohammad Mirjalili, and Andrew Lewis. Grey wolf optimizer. *Advances in engineering software*, 69:46–61, 2014.

[38] Amir Hossein Gandomi, Xin-She Yang, and Amir Hossein Alavi. Mixed variable structural optimization using firefly algorithm. *Computers & Structures*, 89(23-24):2325–2336, 2011.