



Enhanced bat algorithm for COVID-19 short-term forecasting using optimized LSTM

Hafiz Tayyab Rauf¹ · Jiechao Gao² · Ahmad Almadhor³ · Muhammad Arif⁴ · Md Tabrez Nafis⁵

Accepted: 22 July 2021 / Published online: 11 August 2021

© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2021

Abstract

The highly infectious COVID-19 critically affected the world that has stuck millions of citizens in their homes to avoid possible spreading of the disease. Researchers in different fields are continually working to develop vaccines and prevention strategies. However, an accurate forecast of the outbreak can help control the pandemic until a vaccine is available. Several machine learning and deep learning-based approaches are available to forecast the confirmed cases, but they lack the optimized temporal component and nonlinearity. To enhance the current forecasting frameworks' capability, we proposed optimized long short-term memory networks (LSTM) to forecast COVID-19 cases and reduce mean absolute error. For the optimization of LSTM, we applied bat algorithm. Furthermore, to tackle the premature convergence and local minima problem of BA, we proposed an enhanced variant of BA. The proposed version utilized Gaussian adaptive inertia weight to control the individual velocity in the entire swarm. In addition, we substitute random walk with the Gaussian walk to observe the local search mechanism. The proposed LSTM examines the personal best solution with the swarm's local best and preserves the optimal solution by combining the Gaussian walk. To evaluate the optimized LSTM, we compared it with the non-optimal version of LSTM, recurrent neural network, gated recurrent units, and other recent state-of-the-art algorithms. The experimental results prove the superiority of the optimized LSTM over other recent algorithms by obtaining 99.52 % accuracy.

Keywords COVID-19 · Gaussian distribution · Gaussian inertia weight · LSTM

✉ Jiechao Gao
jg5ycn@virginia.edu

Hafiz Tayyab Rauf
hafiztayyabrauf093@gmail.com

Ahmad Almadhor
aaalmadhor@ju.edu.sa

Muhammad Arif
arifmuhammad36@hotmail.com

Md Tabrez Nafis
tabrez.nafis@gmail.com

¹ Department of Computer Science, Faculty of Engineering & Informatics, University of BRADFORD, Bradford, UK

² Department of Computer Science, University of Virginia, Charlottesville, Virginia, US

³ Department of Computer Engineering and Networks, Jouf University, Sakakah, Saudi Arabia

⁴ School of Computer Science, Guangzhou University, Guangzhou 510006, China

⁵ Department of Computer Science and Engineering, Jamia Hamdard, New Delhi, India

1 Introduction

The entire world is experiencing a continuous pandemic called the coronavirus (COVID-19) disease due to severe acute respiratory syndrome coronavirus two (SARS-CoV-2) (Abrams et al. 2020). It has been arisen from Wuhan, the capital of Hubei Province in China, through December 2019 (WH Organization et al. 2020). The virus has been discovered on 7th January and found that it is distributed by human-to-human transmission through direct contact or droplet (Wang et al. 2020; Cucinotta and Vanelli 2020). Covid-19 was estimated to be an average incubation period of 6.4 times and a first reproduction number of 2.24–3.58. It has been spread over the entire world, and so the World Health Organization (WHO) had announced COVID-19, a worldwide outbreak on 11th March 2020 (Huang et al. 2020).

COVID-19 contains a few taxonomy symbols as it belongs to the coronavirus family. All such viruses hold several essential proteins fastened in the viral membrane. As it is well worth discovering, the viral plot displays a large diameter, nearly double of a standard organic layer (Bárcena

et al. 2009). The genome of SARS-CoV-2 includes six notable open-reading structures (ORFs), usually investigated in several CoVs. A number of the genes received less than 80 % nucleotide chain identification to SARS-CoV (Zhou et al. 2020). With ultraviolet warmth and rays, COVID-19 is fragile. There is a common misconception that at 27 C, this virus might have disappeared. Additionally, Covid-19 may be inactivated by chloroform, peroxyacetic acid, chlorine-containing disinfectant, ether (75 percent), except for chlorhexidine (Casella et al. 2020).

In 1995, a large-scale study proved that primary clinical symptoms are dyspnea (21.9 percent of cases), expectoration (28.2 percent of cases), fatigue or myalgia (35.8 % of cases), cough (68.6 % of cases), and fever (88.5 percent of cases). In contrast, the minor ones contain vomiting and nausea (3.9 % of cases), nausea (4.8 % of cases), headache, or nausea (12.1 % of cases) (Lq et al. 2020). The frequency of novel coronavirus, like many pathogens, is thought to transpire by respiratory droplets. Thus, the immense bulk of scattering cases is restricted to the adjacent spaces (Casella et al. 2020).

The SARS-CoV-2 is a pathogenic human coronavirus below the beta coronavirus genus. In the last decade, the two pathogenic species MERS-CoV and SARS-CoV were outbreaks in 2012 and 2002 in the Middle East and China, respectively (Lu et al. 2020; Cui et al. 2019). The laboratory of China put at the NCBI GenBank by discovering the whole genomic sequence (Wuhan-HU1) of the massive RNA virus (SARS-CoV-2) on 10th January (Yang 2020). The SARS-CoV-2 is one positive-stranded RNA virus (Lu et al. 2020).

Following the WHO, no anti-inflammatory medicines and vaccines are not yet prepared for this pandemic (Basu and Chakraborty 2020), and medical industries are looking hard to acquire the vaccine. The vaccine may take at least 18–24 months until it is available, following the quick tracking of the normal vaccine interval of 5–10 decades, and may take additional time to make it appropriate for the large organizations of the world (Grenfell and Drew 2020). Additionally, we do not understand just how long a vaccine could remain successful since the virus mutates. Every attempt was adopted to slow down the coronavirus spread and prepare reasonable medical systems to protect front-line medical staff with sufficient supplies of protective equipment such as personal protective equipment (PPE) masks and other essentials. Consequently, if we know ahead of the number of new coronavirus cases for the next ten days, we could plan our necessary actions. As compared to Asian countries, the USA has been greatly affected by COVID-19. USA COVID-19 cases summary from Feb 2020 to Sep 2020 is illustrated in Fig. 1.

The success of healthcare technologies is a key to artificial intelligence (Panch et al. 2019). Data is structured in smart devices and increases the efficiency of healthcare machine learning (Knight et al. 2016). Several COVID-19 forecasting approaches have been proposed based on machine learning,

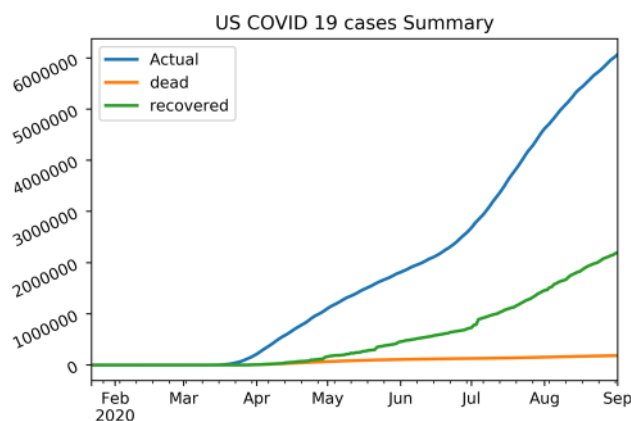


Fig. 1 USA COVID-19 cases summary from Feb 2020 to Sep 2020

deep learning, and statistical learning in the past few weeks. However, the primary issue is they lack the temporal components and nonlinearity in terms of machine learning where deep learning approaches are limited to comparative analysis, and uni-model forecasting (Benvenuto et al. 2020; Wiczorek et al. 2020a). Furthermore, some studies considered epidemiological models that need to make hypothesis-based parameter initialization. That model tends to low the net precision due to its under-fitting data nature (Wiczorek et al. 2020a; Gao et al. 2019).

Several optimization algorithms have been used in previous studies to solve time series problems for the weight optimization of neural networks, such as the arithmetic optimization algorithm (Abualigah et al. 2021), group search optimizer (Abualigah 2020), dragonfly algorithm (Alshinwan et al. 2021), genetic algorithm (Momani et al. 2016), reproducing kernel algorithm (Arqub et al. 2017; Arqub 2017) and fuzzy conformable fractional approaches (Arqub and Al-Smadi 2020).

To predict the distribution of COVID-19 in various regions, the authors used Google trend and ECDC data term frequency (Prasanth et al. 2021). To pick the successful COVID-related search words, they used Spearman correlation. The optimization of hyperparameters through the LSTM network proposed a new technique based on a meta-heuristic GWO algorithm.

Three approaches are suggested (Abbasimehr and Paki 2021) that combine Bayesian optimization and deep learning. The optimized values for hyperparameters are effectively chosen by Bayesian optimization in their process. The system architecture is considered to be a process of multiple-output forecasting. Their proposed methods performed better than the reference model on data from the COVID-19 time series.

In order to forecast the COVID-19 outbreak in Saudi Arabia, a study of various deep learning models is proposed (Elsheikh et al. 2021). Officially recorded data was used to evaluate the model. The optimal values of the parameters of

the model that optimize the accuracy of forecasting have been determined. They used seven statistical evaluation parameters to forecast the accuracy of the model.

Likewise, the previous studies on COVID-19 did not consider the hyperparameter optimization of neural networks that can help boost the performance of models.

To overcome the issue as mentioned above, we proposed a deep learning model that predicts real-time transmission using optimized LSTM. For the optimization of LSTM, we employed BA. To further deals with the premature convergence (Perwaiz et al. 2020; Rauf et al. 2020b), and local minima problem (Rauf et al. 2020a) of BA, we proposed an enhanced variant of BA. The proposed version consists of two significant enhancements. Firstly, we carried out Gaussian adaptive inertia weight to control the individual velocity in the entire swarm. Secondly, we substitute the random walk with the Gaussian walk to explore the local search mechanism.

2 Methodology

2.1 Proposed BA

The real-world challenges are becoming more complicated every day. Swarm intelligence (SI) is the subset of meta-heuristic algorithms employed to tackle complex optimization problems of continuous nature. We used the self-learning nature of this meta-heuristic to optimize the neural network training parameters. Such features clearly state that local interaction is essential between the swarm-based system components to preserve their survival.

In this research, we have carried out an enhanced version of BA to optimize LSTM training weights. The optimized LSTM dynamically adopt optimal training parameter and decide the execution cycle timeline based on the global convergence manner of enhanced BA. We bring two modifications to classical BA. Firstly, we proposed Gaussian adaptive inertia weight to improve the velocity updating mechanism. Lastly, we update each individual’s local searching strategy to retain local solutions based on the weighted mean of their personal best and the current global solution of the entire swarm.

Properties of standard BA are as follows:

- Every micro-bat estimates distance within surroundings and prey by utilizing its property of echolocation.
- Frequency of fixed range is utilized to find micro-bat’s velocity from location beside different loudness and distinct wavelength while searching for prey.
- Emission pulse rate increases to adjust its pulse frequency while estimating distance among prey and micro-bat.

- Loudness will decrease from a considerable positive value to a smaller value.

BA follows three fundamental rules to converge toward an optimal solution.

- Each bat is represented by \bar{x}_i^t for $i = \{1, 2, 3 \dots \bar{N}_p\}$ with the whole population \bar{N}_p in an entire search space S and use sonar echolocation to sense the prey and measure the estimated difference of the distance to the prey.
- During the convergence process, each bat \bar{x}_i^t moves with velocity \bar{v}_i^t and the frequency of f_{min}^t . The current position of individual can be represented by \bar{x}_{ip}^t where p represents the partial coordinate of the current search space. The frequency f_{min}^t consolidates with bat wavelength ω and variation of loudness A_o .
- The variation of loudness A_o depends on the current location \bar{x}_{ip}^t and the weighted distance D_{ip}^t .

Population of fixed size S_p , in our case $S_p = 40$, is initialized with the random initial values following the uniform distribution $\bar{x}_i^t \in [\bar{x}_l, \bar{x}_u]$, where l and u are lower and upper limits of uniformly distributed sequence. After population initialization, the mutation operators are used to encourage the bats’ movement in the multidimensional search space. The ultimate objective of this phase is to obtain the new local solution, while the frequency f_{min}^t factor controls the step-size of the solution. For each individual \bar{x}_i^t , the current frequency f_i^t , current velocity \bar{v}_i^t and the current bats position \bar{x}_{ip}^t can be updated using the following equations.

$$f_i^t = f_{min}^t + (f_{max}^t - f_{min}^t) \cdot R \tag{1}$$

$$\bar{v}_i^{t+1} = \bar{v}_i^t + (\bar{x}_{ip}^t - \bar{x}_{ig}^t) \cdot f_i^t \tag{2}$$

$$\bar{x}_{ip}^{t+1} = \bar{x}_{ip}^t + \bar{v}_i^{t+1} \tag{3}$$

Referred to equation 1, $f_{max}^t - f_{min}^t$ are the difference of lower and upper corresponding frequency where R indicates the random number over the interval of $[0, 1]$. Velocity of each individual \bar{x}_i^t can be updated using equation 2, where $\bar{x}_{ip}^t - \bar{x}_{ig}^t$ is the mean difference of local solution \bar{x}_{ip}^t of entire swarm and global solution \bar{x}_{ig}^t of all swarms. Likewise, the new vector solution \bar{x}_{ip}^{t+1} can be determine using equation 3.

In the proposed BA, we introduced Gaussian adaptive inertia weight to update the velocity in such a manner to avoid more long jumps leading to exploration and to avoid more short jumps leading to exploitation. The proposed Gaussian adaptive inertia weight can help the velocity updating mechanism achieve each individual’s optimal convergence steps. The Gaussian function can be defined as:

$$f(x) = xe^{-\frac{(x-y)^2}{2\sigma^2}} \tag{4}$$

Table 1 Recent related works with their dataset details and results

Ref.	Dataset	Model	Results
Wieczorek et al. (2020b)	Government repositories	NAdam training model	Accuracy above 99%
Chowdhury et al. (2020)	Bangladesh COVID-19	Neuro-fuzzy inference system (ANFIS)	Correlation coefficient 0.75, MAPE 4.51, and RMSE 6.55
Dutta et al. (2020)	WHO official	CNN and RNN	CNN-LSTM approach outperforms
Chimmula and Zhang (2020)	Dataset Canadian Health Authority	LSTM	Gained highly accurate results
Arora et al. (2020)	Indian dataset	LSTM	Yields high accuracy
Pathan et al. (2020)	The patient's dataset of different countries	RNN and LSTM	Obtained optimum results
Alakus and Turkoglu (2020)	Laboratory data	Clinical predictive models	Accuracy of 86.66% and F1-score of 91.89%,
Tuli et al. (2020)	Data by Hannah Ritchie	ML-based improved model	Yields high accuracy
Kavadi et al. (2020)	Indian dataset	Linear regression model	Outperformed state-of-the-art methods
Pinter et al. (2020)	Data from Hungary	Multi-layered perceptron-imperialist competitive algorithm (MLP-ICA)	Obtained promising results
Prasanth et al. (2021)	Google trend and ECDC data	A hybrid GWO algorithm	Reduce MAPE by 74% results
Abbasimehr and Paki (2021)	Live time series data	Bayesian optimization-based algorithm	Mean SMAPE is 0.25 results
Elsheikh et al. (2021)	Official data from Saudi Arabia	LSTM and other variants	Obtained highly accurate results

where (x, y, z) are real constant that can be varied over the nature of the problem. A bell shape curve in the Gaussian distribution indicates the height of bell curves and can help the population control the exploration process with the following probability density function.

$$g(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}. \quad (5)$$

In equation 5, $\mu = y$ and can be interpreted as the expected value with variance $\sigma^2 = z^2$.

In order to generate optimal location vectors \bar{g}_i^{t+1} through Gaussian distribution over t iterations and D dimensions, the mathematical definition following the adaptive process can be:

$$\bar{g}_i^{t+1} = \bar{g}_{min} + (\bar{g}_{max} - \bar{g}_{min}) * \bar{g}_i^t \quad (6)$$

where $\bar{g}_{max} - \bar{g}_{min}$ are upper and lower intervals $[0, 1]$ of Gaussian distribution. The proposed BA utilized the following equation to update the velocity of each bat \bar{v}_{gi}^{t+1} .

$$\bar{v}_{gi}^{t+1} = \bar{g}_i^{t+1} * \bar{v}_i^t + (\bar{x}_{ip}^t - \bar{x}_{ig}^t) * f_i^t. \quad (7)$$

In equation 7, \bar{g}_i^{t+1} shows the proposed Gaussian adaptive inertia weight factor, controlling the exploration and exploitation during the entire convergence process. Gaussian bell curves in the adaptive inertia weight dynamically select each bat's speed to help the local best vector holder bat to escape local minima. Apart from velocity \bar{v}_{gi}^{t+1} , updated local solutions \bar{x}_{ip}^{new} play an essential role in the exploitation of bats. Consider the speed is regulated, but the newly generated local solutions \bar{x}_{ip}^{new} are not robust enough to limit the boundary of the entire swarm's global best \bar{x}_{ig}^t . In that case, premature convergence can be held. Standard BA uses the following equation to select the best solution among all existing vectors in the swarm:

$$\bar{x}_{ip}^{new} = \bar{x}_{ig}^t + \varepsilon A_i^t. \quad (8)$$

ε is a random walk generator throughout $[0, 1]$ and A_i^t represents the average loudness factor. The random walk can produce the best solution in the current iteration t and build the worst one in the next iteration $t + 1$. The entire local best holder individual will likely follow the best solution \bar{x}_{ig}^t , which is the worst in the next iteration $t + 1$ and leads to the local minima and premature convergence problem. To avoid this random selection that leads to the worst local best solution and effect exploitation, we replace this random walk

with a Gaussian walk and propose a local search mechanism. Our proposed variant of BA will use the following equation to attain the local best solution \bar{x}_{iG}^{new} .

$$\bar{x}_{iG}^{new} = \bar{x}_{ig}^t + \bar{g}_i^{t+1}(\bar{x}_{ig}^t - \bar{P}_{ig}^t) + \varepsilon A_i^t. \tag{9}$$

In the proposed equation 9, \bar{g}_i^{t+1} is previously computed Gaussian distribution where $\bar{x}_{ig}^t - \bar{P}_{ig}^t$ is the mean difference of local best of swarm \bar{x}_{ig}^t and the personal best \bar{P}_{ig}^t of each bat. The proposed solution will iteratively evaluate the current best and the local best solution \bar{P}_{ig}^t for each swarm \bar{x}_{ig}^t in the population and check the following condition to use the iterative difference.

$$\bar{x}_{iG}^{new} = \begin{cases} \bar{x}_{ig}^t & \text{if } (\bar{x}_{ig}^t > \bar{x}_{iG}^t) \\ \bar{x}_{ig}^t - \bar{P}_{ig}^t & \text{Otherwise} \end{cases}. \tag{10}$$

Referred to equation 10, the new local best will be selected \bar{x}_{ig}^t if the bats' personal best is less than the swarm local best otherwise, the weighted mean of local best \bar{x}_{ig}^t and global best \bar{P}_{ig}^t will be chosen as new local best.

New N local bests \bar{x}_{iG}^{new} will likely control by the convergence rate, which can be defined by two critical factors loudness \bar{A}_i and pulse emission rate \bar{r}_i^t which can be update thought the following two equations.

$$\bar{A}_i^{t+1} = \alpha \bar{A}_i^t \tag{11}$$

$$\bar{r}_i^t = \bar{r}_i^0 [1 - \exp(-\gamma^t)]. \tag{12}$$

2.2 Optimized Long Short-Term Memory (LSTM)

Recurrent neural network (RNN) has turned out to be the most reliable algorithm for prediction as essential features are extracted automatically from samples of training (Jiang and Schotten 2020). RNN performed well at data processing, and ensured encouraging outcomes for time series prediction while keeping immense information in the internal state (Connor et al. 1994). Nevertheless, it might take much training time due to gradient detonate and evanescence problems (Tomar and Gupta 2020). Hence, in 1997 a long short-term memory RNN structure was designed by Schmidhuber and Hochreiter (Hochreiter and Schmidhuber 1997) to overcome that flaw by administering long-term dependency through multiplicative gates that will handle memory cells and flow of information in the recurrent hidden layer. LSTM's architecture comprises four gates, i.e., input gate, output gate, control gate, and forget gate (Tomar and Gupta 2020).

Input can be defined as:

$$i_t = \sigma(W_i * [h_{t-1}, x_t] + b_i). \tag{13}$$

The information extracted from the above equation can be transferred to the cell. Forget gate decides data that will be ignored from the previous layer's input by utilizing the following equation:

$$f_t = \sigma(W_i * [h_{t-1}, x_t] + b_i). \tag{14}$$

The input from the entire memory cell is controlled by control gate through following equations:

$$\tilde{C} = \sigma(W_c * [h_{t-1}, x_t] + b_c) \tag{15}$$

$$\tilde{C}_t = f_t * \tilde{C}_{t-1} + i_t * \tilde{C}_t \tag{16}$$

Output and hidden layer h_{t-1} is updated as following:

$$O_t = \sigma(W_o * [h_{t-1}, x_t] + b_o) \tag{17}$$

$$h_t = O_t * \tanh(\tilde{C}_t). \tag{18}$$

The interval [-1 to 1] is normalized by using tanh, where W os the weight matrices and σ shows activation function taken as sigmoid.

We feed the learning rate, momentum rate, and dropout rate in each of the LSTM dropout layers to the BA for automatic optimization of the hyperparameters. Each parameter is examined before the classification layer of LSTM to determine BA's best optimal global solution. If the fitness function produces the same values, the proposed algorithm will check in the next generation to see if it avoids premature convergence.

Hyperparameters of each hidden layer h_{t-1} for $t = \{1, 2, 3 \dots N\}$ are optimized by providing global solution \bar{x}_{iG}^{new} obtained using equation 9. The output layer of optimized LSTM can be interpreted as:

$$O_t = \sigma \left(W_o * \left[h_{t-1} \left(\begin{cases} \bar{x}_{ig}^t & \text{if } (\bar{x}_{ig}^t > \bar{x}_{iG}^t) \\ \bar{x}_{ig}^t - \bar{P}_{ig}^t & \text{Otherwise} \end{cases} \right), x_t \right] + b_o \right) \tag{19}$$

where each hidden layer choose global best of the entire population \bar{x}_{ig}^t or mean of personal best and local best of swarm $\bar{x}_{ig}^t - \bar{P}_{ig}^t$. The pseudocode of proposed Algorithm is presented in Algorithm 1.

We also checked single parameter optimization impact on the proposed technique, and we observed that only learning rate optimization produces a negligible impact on the performance of the proposed LSTM. However, the collective optimization of the learning rate, momentum rate, and dropout rate tends to increase the overall performance of the proposed LSTM.

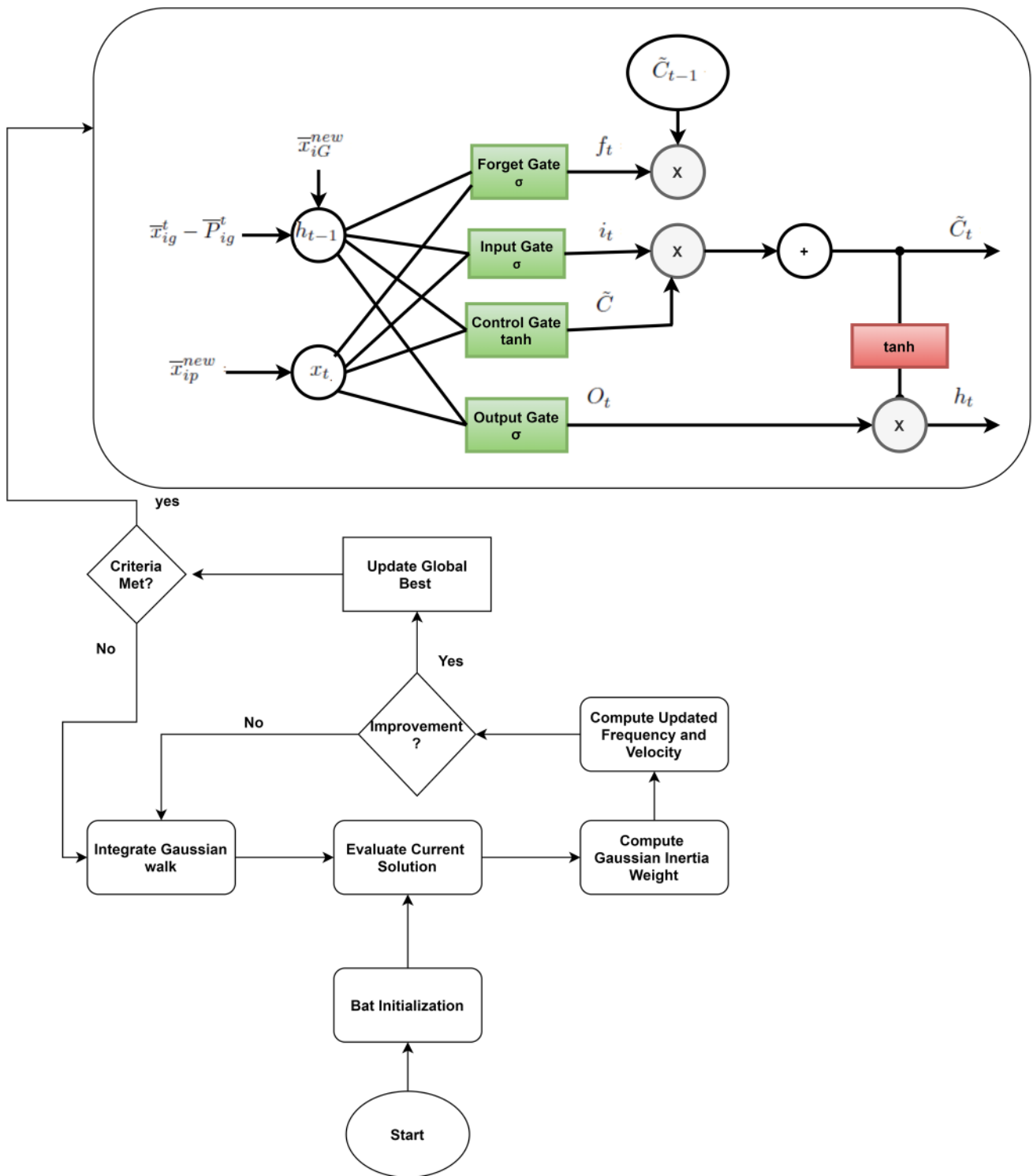


Fig. 2 Proposed architecture of optimized LSTM

3 Experiments

WHO accounted for the outbreak of COVID-19 in states and regions around the world. Several areas of South and North America, in particular, witness the adverse effects of

a massive COVID-19 explosion. The operation of huge air traffic between each state of the USA has entirely encouraged COVID-19 to propagate from its source to the next infected states; individual-to-individual spread has thus been reported among travelers worldwide. The primary goal of this

Algorithm 1 Pseudocode for proposed BA

Require: Initial fitness values of LSTM parameters: \bar{x}_i^t for $i = \{1, 2, 3 \dots \bar{N}_p\}$

Ensure: Optimal solution \bar{x}_{iG}^{new} for $i = \{1, 2, 3 \dots \bar{N}_p\}$

- 1: Population initialization \bar{x}_i^t Random(0,1)
- 2: Determine initial fitness (\bar{x}_i^t)
- 3: Determine initial global fitness (\bar{x}_{iG}^{new})
- 4: **while** $t \leq termination$ **do**
- 5: **for** $i = 1$ to \bar{N}_p **do**
- 6: Update f_i^t through equation 1.
- 7: Update \bar{v}_{gi}^{t+1} through equation 7.
- 8: Update \bar{x}_{ip}^{t+1} through equation 3.
- 9: **if** $r(0, 1) > \bar{r}_i^t$ **then**
- 10: Update local solution through equations 9-10.
- 11: **end if**
- 12: $f(new)$ \bar{x}_{iG}^{new}
- 13: Generation = Generation + 1
- 14: **if** $f(new) < f(old)$ and $R(0, 1) < \bar{A}_i^t$ **then**
- 15: \bar{x}_{iG}^{new} \bar{x}_{ip}^{t+1} and $f(old)$ $f(new)$
- 16: **end if**
- 17: \bar{x}_{iG}^{new} $\min(\bar{x}_{ip}^t)$
- 18: **end for**
- 19: **end while**

research is the prediction and forecast of epidemic spreading by COVID-19. This examination contains the count of confirmed and recovered cases obtained from the WHO website regularly. We consider the USA for the experiments and employed live dataset updated daily. The utilized dataset is available at (WHO 2020).

The experiments are conducted using specific python packages, namely Keras, TensorFlow, NumPy, and iplot using python language. To compare the performance of the proposed optimized LSTM, we tested other standard forecasting algorithms, i.e., Simple LSTM, GRU, and RNN.

3.1 Results

This study provides an optimized deep-learning model for COVID-19’s time series analysis of the USA. The proposed framework dynamically selects optimal training parameters and determines the execution cycle based on enhanced BA’s global convergence manner.

The forecasting of COVID-19 was achieved in two preliminary stages: data training and evaluation. To compared the proposed variant with existing algorithms, we used five evaluation metrics; namely root mean absolute error (RMSE), mean absolute percentage error (MAPE), standard deviation (Stdev), prediction interval, and accuracy. The following

equations can define RMSE, MAPE, and Stdev:

$$RMSE = \left[\sum_{i=1}^N \frac{(a_i - a_o)^2}{N} \right]^{\frac{1}{2}} \tag{20}$$

where $a_i - a_o$ represents squared difference forecasted and actual values.

$$MAPE = \frac{1}{n} \sum \frac{|e|}{d} \tag{21}$$

where $|e|$ indicates absolute error and d shows demand for each period.

$$Stdev = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (x_i - \bar{x})^2}. \tag{22}$$

In the above equation \bar{x} is mean of i th sample and N indicates total number of instance.

The raw data is pre-processed and standardized in the initial stages and subsequently used to develop the optimized predictive model based on LSTM. The model’s boundary parameters are selected so that the MAPE can be minimized. From a particular stage on, the optimized LSTM with the optimal learning parameters is used in the testing process to predict the extent of COVID-19 cases in the USA.

Table 2 presents the empirical results for confirmed and predicted cases obtained through GRU, RNN, LSTM, and optimized LSTM. RMSE shows the root mean square errors in each network during the training. MAPE is total loss subtracted from precision, where Stdev shows the significant difference between confirmed and predicted COVID-19 cases. Prediction interval represents the difference in response to confirmed cases between each day of the forecasted cases.

We presented a statistical test called Kruskal–Wallis test for the experimental results, comparing the results with other published methods. The average rank, median value, and Z-score obtained through Kruskal–Wallis test for each employed algorithm is presented in Table 5.

Likewise, training and validation loss minimization curves using GRU, RNN, LSTM, and optimized LSTM are illustrated in Figs. 3, 4, 5, and 6. The convergence curves of real and forecasted COVID-19 cases through optimized LSTM in the USA are presented in Fig. 7.

A comparison of the proposed optimized LSTM with other standard deep learning forecasting models is tabulated in Table 4. We take the forecasting dates from 1/9/20 to 10/9/20, and to validate the predicted values, we retain previous ten-day cases 22/8/20 to 31/8/20. Referred to Table (4), actual confirmed cases do not appear yet in the USA from 31/8/20 to 1/9/20, predicted shows the forecasted cases through existing

Table 2 Comparison of proposed optimized LSTM with other standard deep learning forecasting models

Model	RMSE	MAPE	Stdev	Prediction interval	Accuracy
GRU	1786.613	30.01539	3261.895	6393.313572	70
RNN	531.3041	8.817398	970.0242	1901.247	91
LSTM	751.2309	12.12951	1371.554	2688.245	88
Optimized LSTM	32.99262	0.483875	60.23602	118.0626	99.52

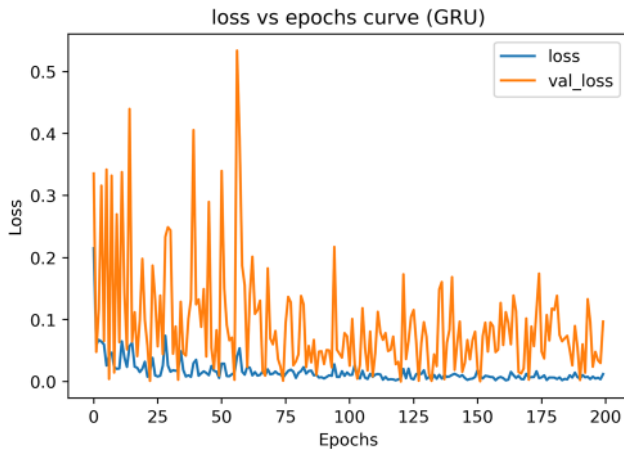


Fig. 3 Training and validation loss minimization curves using GRU

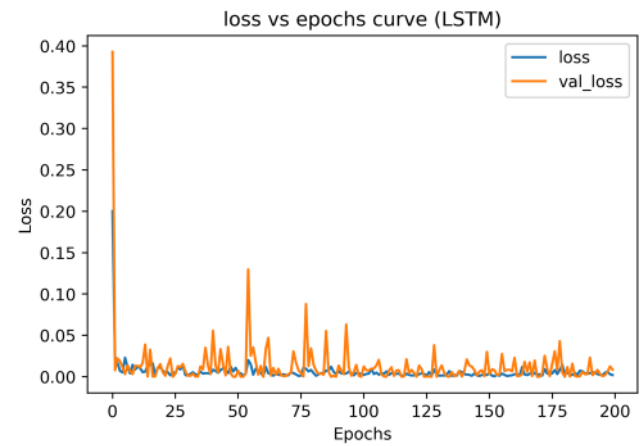


Fig. 5 Training and validation loss minimization curves using LSTM

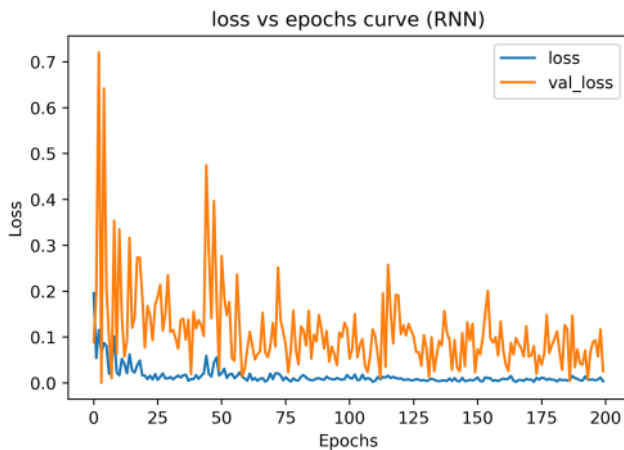


Fig. 4 Training and validation loss minimization curves using RNN

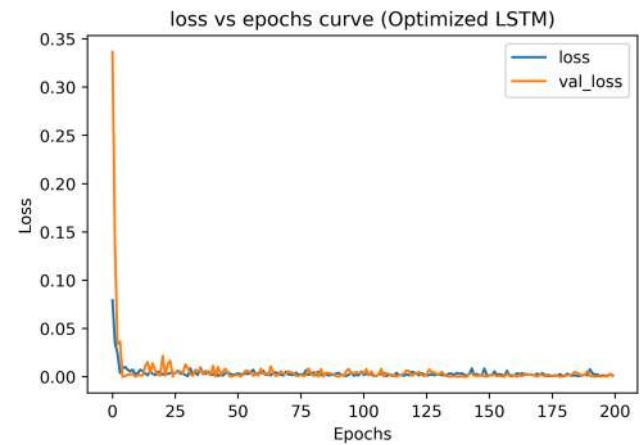


Fig. 6 Training and validation loss minimization curves using optimized LSTM

GRU, RNN, LSTM, and proposed optimized LSTM, respectively.

For validation of the performance of the proposed optimized LSTM, Fig. 8 represents the forecasting curves of several networks compared to the actual number of cases.

Comparison of proposed optimized LSTM with other variants of LSTM and other deep learning models is given in Table 3.

3.2 Analysis

Table 2 shows that GRU obtained the worst accuracy with 1786.613 RMSE and 3261.895 Stdev, which shows a significant difference between actual and predicted COVID-cases. After GRU, standard LSTM performed better with 2688.245 prediction intervals and 12.12 MAPE. The performance of RNN is relatively good compared to GRU and LSTM with 91 % accuracy and 1371.55 Stdev. Lastly, it can be seen that the proposed version of optimized LSTM outperformed all other deep learning models with 32.99 RMSE better than

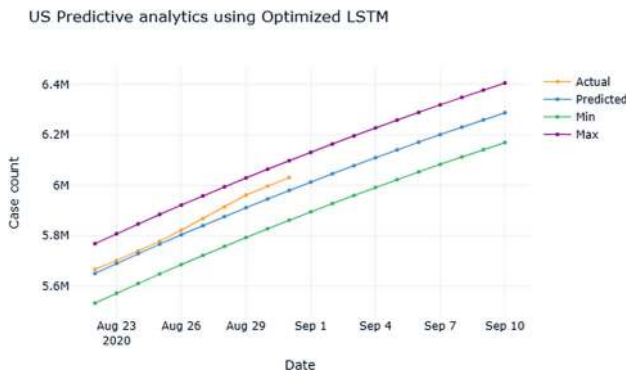


Fig. 7 Convergence of real and forecasted COVID-19 cases through optimized LSTM in the USA

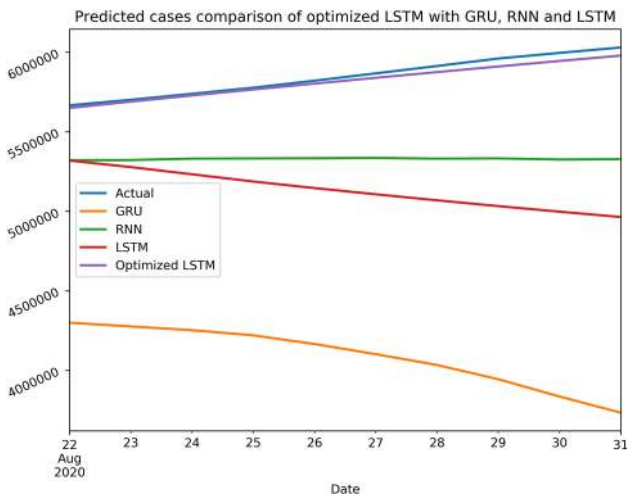


Fig. 8 Predicted cases comparison of optimized LSTM with GRU, RNN, and LSTM

GRU, 0.4838 MAPE better than LSTM, and only 60.23 significant difference among confirmed and predicted cases.

Furthermore, the validation loss in the case of GRU and RNN is not stable throughout the learning process and meets greater than 0.5 and 0.7 (refer Figs. 3 and 4). From Fig. 5, the validation loss of LSTM is stable compared to GRU and RNN throughout the learning process with a greater 0.40. As opposed to GRU, LSTM, and RNN, the proposed model minimized the validation loss up to 0.04 and shows the better capability of loss minimization (refer Fig. 6).

The performance of the proposed optimized LSTM can be confirmed through Fig. 7, where the USA’s actual cases on 31/8/20 were 6030587, and the predictions were 3734918, 5328279 7653031, and 6097641 using GRU, RNN, LSTM, and optimized LSTM, respectively.

From Table 5, it can be observed that the proposed LSTM obtained the best mean rank of 17.0 through Kruskal–Wallis test as compared to others. Advanced algorithms such as NAdam with 41 mean rank, two LSTM variants with 16 and 13 mean ranks, respectively. Similarly, the proposed LSTM

Table 3 Comparison of proposed optimized LSTM with other variants of LSTM and other deep learning models

Model	RMSE	MAPE	Accuracy
LSTM Wieczorek et al. (2020b)	–	–	93.56
NAdam Wieczorek et al. (2020b)	–	–	87.73
RMSprop Wieczorek et al. (2020b)	–	–	87.65
Adam Wieczorek et al. (2020b)	–	–	87.53
Adamax Wieczorek et al. (2020b)	–	–	87.47
Ftrl Wieczorek et al. (2020b)	–	–	40.10
Adagrad Wieczorek et al. (2020b)	–	–	40.10
SGD Wieczorek et al. (2020b)	–	–	9.8
Scenario 1 Chowdhury et al. (2020)	297.89	5425	–
Scenario 2 Chowdhury et al. (2020)	216.48	23.30	–
Scenario 3 Chowdhury et al. (2020)	600.61	38.06	–
LSTM-1 Chimmula and Zhang (2020)	34.83	–	93.4
LSTM-2 Chimmula and Zhang (2020)	45.70	–	92.67
Convolutional LSTM Arora et al. (2020)	–	5.05	–
Stacked LSTM Arora et al. (2020)	–	4.81	–
Bidirectional LSTM Arora et al. (2020)	–	3.22	–
RNN Alakus and Turkoglu (2020)	–	–	84.00
LSTM Alakus and Turkoglu (2020)	–	–	90.34
CNNRNN Alakus and Turkoglu (2020)	–	–	86.24
CNNLSTM Alakus and Turkoglu (2020)	–	–	92.30
CNN Alakus and Turkoglu (2020)	–	–	87.35
ANN Alakus and Turkoglu (2020)	–	–	86.90
Optimized LSTM	32.99	0.48	99.52

Table 4 Comparison of proposed optimized LSTM with other standard deep learning forecasting models

Date	GRU	RNN	LSTM	Optimized LSTM
1/9/20	3619310	5305265	4932695	6012715
2/9/20	3506454	5304446	4903980	6045536
3/9/20	3344938	5304747	4876812	6077771
4/9/20	3139538	5304128	4851221	6109418
5/9/20	2912792	5303244	4827170	6140511
6/9/20	2693745	5302879	4804581	6171062
7/9/20	2472994	5301480	4783279	6201055
8/9/20	2310167	5301190	4763229	6230511
9/9/20	2206934	5299707	4744368	6259414
10/9/20	2070085	5299569	4726624	6287779

outperformed other published results by obtaining the best positive Z-score of 163.

We can conclude that using the proposed optimized framework can help the USA and other governments predict the actual cases with 99 % accuracy and take precautionary measures in advance.

Table 5 Kruskal–Wallis test: proposed LSTM vs recent state-of-the-art algorithms

Model	Median	Ave rank	Z
Adagrad Wieczorek et al. (2020b)	40.100	2.5	−1.33
Adam Wieczorek et al. (2020b)	87.530	9.0	0.00
Adamax Wieczorek et al. (2020b)	87.470	8.0	−0.20
ANN Alakus and Turkoglu (2020)	86.900	6.0	−0.61
CNN Alakus and Turkoglu (2020)	87.350	7.0	−0.41
CNNLSTM Alakus and Turkoglu (2020)	92.300	13.0	0.82
CNNRNN Alakus and Turkoglu (2020)	86.240	5.0	−0.82
Ftrl Wieczorek et al. (2020b)	40.100	2.5	−1.33
LSTM-1 Chimmula and Zhang (2020)	93.400	15.0	1.22
LSTM-2 Chimmula and Zhang (2020)	92.670	14.0	1.02
LSTM Alakus and Turkoglu (2020)	90.340	12.0	0.61
LSTM Wieczorek et al. (2020b)	93.560	16.0	1.43
NAdam Wieczorek et al. (2020b)	87.730	11.0	0.41
RMSprop Wieczorek et al. (2020b)	87.650	10.0	0.20
RNN Alakus and Turkoglu (2020)	84.000	4.0	−1.02
SGD Wieczorek et al. (2020b)	9.800	1.0	−1.63
Optimized LSTM	99.520	17.0	1.63

4 Conclusion

This research offers the optimized LSTM to forecasts COVID-19 cases in the USA. Many machine learning and deep learning approaches are available to forecast confirmed cases, but they lack both the optimized temporal aspect and nonlinearity. To overcome this issue, we applied the BA for the optimization of LSTM. Besides, we implemented an enhanced BA variant to tackle BA's premature convergence and local minima problems. The proposed version of BA used Gaussian adaptive inertia weight to control the individual velocity in the swarm. In addition, we replace the random walk with the Gaussian walk to observe the local search. The robust local search mechanism assists LSTM hyperparameter optimization during the training process. The proposed optimized LSTM is compared with GRU, RNN, and LSTM. Empirical results reveal that optimized LSTM minimized MAPE by 0.48, which is far better than the existing algorithms.

In future work, we intend to adopt other evolutionary models such as the Genetic Algorithm and Differential evolution algorithm in the regression-based deep learning model for multivariate forecasting of a pandemic.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

Informed consent Informed consent was obtained from all individual participants included in the study.

References

- Abbasimehr H, Paki R (2021) Prediction of COVID-19 confirmed cases combining deep learning methods and Bayesian optimization. *Chaos Solitons Fractals*. <https://doi.org/10.1016/j.chaos.2020.110511>
- Abrams ME, Johnson KA, Perelman SS, Zhang Ls, Endapally S, Mar KB, Thompson BM, McDonald JG, Schoggins JW, Radhakrishnan A, et al (2020) Oxysterols provide innate immunity to bacterial infection by mobilizing cell surface accessible cholesterol. *Nat Microbiol*, pp 1–14
- Abualigah L (2020) Group search optimizer: a nature-inspired meta-heuristic optimization algorithm with its results, variants, and applications. *Neural Comput Appl*, pp 1–24
- Abualigah L, Diabat A, Mirjalili S, Abd Elaziz M, Gandomi AH (2021) The arithmetic optimization algorithm. *Comput Methods Appl Mech Eng* 376:113609
- Alakus TB, Turkoglu I (2020) Comparison of deep learning approaches to predict covid-19 infection. *Chaos Solitons Fractals*, p 110120
- Alshinwan M, Abualigah L, Shehab M, Abd Elaziz M, Khasawneh AM, Alabool H, Al Hamad H (2021) Dragonfly algorithm: a comprehensive survey of its results, variants, and applications. *Multimedia Tools Appl*, pp 1–38
- Arora P, Kumar H, Panigrahi BK (2020) Prediction and analysis of COVID-19 positive cases using deep learning models: a descriptive case study of india. *Chaos Solitons Fractals* 139:110017. <https://doi.org/10.1016/j.chaos.2020.110017>
- Arqub OA (2017) Adaptation of reproducing kernel algorithm for solving fuzzy fredholm-volterra integrodifferential equations. *Neural Comput Appl* 28(7):1591–1610
- Arqub OA, Al-Smadi M (2020) Fuzzy conformable fractional differential equations: novel extended approach and new numerical solutions. *Soft Comput*, pp 1–22
- Arqub OA, Al-Smadi M, Momani S, Hayat T (2017) Application of reproducing kernel algorithm for solving second-order, two-point fuzzy boundary value problems. *Soft Comput* 21(23):7191–7206
- Bárcena M, Oostergetel GT, Bartelink W, Faas FG, Verkleij A, Rottier PJ, Koster AJ, Bosch BJ (2009) Cryo-electron tomography of mouse hepatitis virus: insights into the structure of the coronavirus. *Proc Natl Acad Sci* 106(2):582–587
- Basu A, Chakraborty S (2020). Faculty opinions recommendation of a pneumonia outbreak associated with a new coronavirus of probable bat origin. <https://doi.org/10.3410/f.737304963.793575060>
- Benvenuto D, Giovanetti M, Vassallo L, Angeletti S, Ciccozzi M (2020) Application of the ARIMA model on the COVID-2019 epidemic dataset. *Data Brief* 29:105340. <https://doi.org/10.1016/j.dib.2020.105340>
- Cascella M, Rajnik M, Cuomo A, Dulebohn SC, Di Napoli R (2020) Features, evaluation and treatment coronavirus (covid-19). In: *StatPearls* [internet], StatPearls Publishing
- Chimmula VKR, Zhang L (2020) Time series forecasting of covid-19 transmission in canada using lstm networks. *Chaos Solitons Fractals*, p 109864
- Chowdhury AA, Hasan KT, Hoque KKS (2020) Analysis and prediction of covid-19 pandemic in bangladesh by using long short-term memory network (lstm) and adaptive neuro fuzzy inference system (anfis)

- Connor J, Martin R, Atlas L (1994) Recurrent neural networks and robust time series prediction. *IEEE Trans Neural Netw* 5(2):240–254. <https://doi.org/10.1109/72.279188>
- Cucinotta D, Vanelli M (2020) Who declares covid-19 a pandemic. *Acta Bio-Medica Atenei Parmensis* 91(1):157–160
- Cui J, Li F, Shi ZL (2019) Origin and evolution of pathogenic coronaviruses. *Nat Rev Microbiol* 17(3):181–192
- Dutta S, Bandyopadhyay SK, Kim TH (2020) Cnn-lstm model for verifying predictions of covid-19 cases. *Asian J Res Comput Sci*, pp 25–32
- Elsheikh AH, Saba AI, Abd Elaziz M, Lu S, Shanmugan S, Muthuramalingam T, Kumar R, Mosleh AO, Essa F, Shehabeldeen TA (2021) Deep learning-based forecasting model for covid-19 outbreak in saudi arabia. *Process Safety Environ Protect* 149:223–233
- Gao J, Wang H, Shen H (2019) Task failure prediction in cloud data centers using deep learning. In: 2019 IEEE International Conference on Big Data (Big Data), IEEE. <https://doi.org/10.1109/bigdata47090.2019.9006011>
- Grenfell R, Drew T (2020) Here's why its taking so long to develop a vaccine for the new coronavirus. *Science Alert Archived from the original on 28*
- Hochreiter S, Schmidhuber J (1997) Long short-term memory. *Neural Comput* 9(8):1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- Huang C, Wang Y, Li X, Ren L, Zhao J, Hu Y, Zhang L, Fan G, Xu J, Gu X et al (2020) Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China. *Lancet* 395(10223):497–506
- Jiang W, Schotten HD (2020) Deep learning for fading channel prediction. *IEEE Open J Commun Soc* 1:320–332
- Kavadi MDP, Patan R, Ramachandran M, Gandomi AH (2020) Partial derivative nonlinear global pandemic machine learning prediction of covid 19. *Chaos Solitons Fractals*, p 110056
- Knight GM, Dharan NJ, Fox GJ, Stennis N, Zwerling A, Khurana R, Dowdy DW (2016) Bridging the gap between evidence and policy for infectious diseases: how models can aid public health decision-making. *Int J Infect Dis* 42:17–23. <https://doi.org/10.1016/j.ijid.2015.10.024>
- Li L, Huang T, Wang Y, Wang Z, Liang Y, Huang T, Zhang H, Sun W, Wang Y (2020) Covid-19 patients clinical characteristics, discharge rate, and fatality rate of meta-analysis. *J Med Virol* 92(6):577–583
- Lu R, Zhao X, Li J, Niu P, Yang B, Wu H, Wang W, Song H, Huang B, Zhu N et al (2020) Genomic characterisation and epidemiology of 2019 novel coronavirus: implications for virus origins and receptor binding. *Lancet* 395(10224):565–574
- Momani S, Abo-Hammour ZS, Alsmadi OM (2016) Solution of inverse kinematics problem using genetic algorithms. *Appl Math Inf Sci* 10(1):225
- WH Organization, et al (2020) Covid-19 in suspected human cases: interim guidance, 17 January 2020. WHO
- Panch T, Mattie H, Celi LA (2019) The inconvenient truth about AI in healthcare. *NPJ Digital Med*. <https://doi.org/10.1038/s41746-019-0155-4>
- Pathan RK, Biswas M, Khandaker MU (2020) Time series prediction of covid-19 by mutation rate analysis using recurrent neural network-based lstm model. *Chaos Solitons Fractals*, p 110018
- Perwaiz U, Younas I, Anwar AA (2020) Many-objective bat algorithm. *PLoS ONE* 15(6):e0234625
- Pinter G, Felde I, Mosavi A, Ghamisi P, Gloaguen R (2020) Covid-19 pandemic prediction for hungary; a hybrid machine learning approach. *Mathematics* 8(6):890
- Prasanth S, Singh U, Kumar A, Tikkiwal VA, Chong PH (2021) Forecasting spread of covid-19 using google trends: a hybrid gwo-deep learning approach. *Chaos Solitons Fractals* 142:110336
- Rauf HT, Malik S, Shoaib U, Irfan MN, Lali MI (2020a) Adaptive inertia weight bat algorithm with sugeno-function fuzzy search. *Appl Soft Comput* 90:106159. <https://doi.org/10.1016/j.asoc.2020.106159>
- Rauf HT, Shoaib U, Lali MI, Alhaisoni M, Irfan MN, Khan MA (2020b) Particle swarm optimization with probability sequence for global optimization. *IEEE Access* 8:110535–110549. <https://doi.org/10.1109/access.2020.3002725>
- Tomar A, Gupta N (2020) Prediction for the spread of COVID-19 in india and effectiveness of preventive measures. *Sci Total Environ* 728:138762. <https://doi.org/10.1016/j.scitotenv.2020.138762>
- Tuli S, Tuli S, Tuli R, Gill SS (2020) Predicting the growth and trend of covid-19 pandemic using machine learning and cloud computing. *Internet of Things*, p 100222
- Wang C, Horby PW, Hayden FG, Gao GF (2020) A novel coronavirus outbreak of global health concern. *Lancet* 395(10223):470–473
- WHO (2020) Git-hub live data. https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_confirmed_global.csv
- Wieczorek M, Siłka J, Woźniak M (2020a) Neural network powered COVID-19 spread forecasting model. *Chaos Solitons Fractals* 140:110203. <https://doi.org/10.1016/j.chaos.2020.110203>
- Wieczorek M, Siłka J, Woźniak M (2020b) Neural network powered covid-19 spread forecasting model. *Chaos Solitons Fractals*, p 110203
- Yang J (2020) Inhibition of sars-cov-2 replication by acidizing and ma lyase-modified carbon nanotubes combined with photodynamic thermal effect. *J Exploratory Res Pharmacol*, pp 1–6
- Zhou P, Yang XL, Wang XG, Hu B, Zhang L, Zhang W, Si HR, Zhu Y, Li B, Huang CL et al (2020) A pneumonia outbreak associated with a new coronavirus of probable bat origin. *Nature* 579(7798):270–273

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