

## Covid-19 Forecasting with Deep Learning-based Half-binomial Distribution Cat Swarm Optimization

P. Renukadevi<sup>1,\*</sup> and A. Rajiv Kannan<sup>2</sup>

<sup>1</sup>Department of Computer Science and Engineering, Paavai Engineering College, Namakkal, 637018, Tamil Nadu, India

<sup>2</sup>Department of Computer Science and Engineering, K.S.R. College of Engineering, Tiruchengode, Namakkal, 637215, Tamil Nadu, India

\*Corresponding Author: P. Renukadevi. Email: [pgrenu@gmail.com](mailto:pgrenu@gmail.com)

Received: 09 October 2021; Accepted: 10 January 2022

**Abstract:** About 170 nations have been affected by the COVID-19 epidemic. On governing bodies across the globe, a lot of stress is created by COVID-19 as there is a continuous rise in patient count testing positive, and they feel challenging to tackle this situation. Most researchers concentrate on COVID-19 data analysis using the machine learning paradigm in these situations. In the previous works, Long Short-Term Memory (LSTM) was used to predict future COVID-19 cases. According to LSTM network data, the outbreak is expected to finish by June 2020. However, there is a chance of an over-fitting problem in LSTM and true positive; it may not produce the required results. The COVID-19 dataset has lower accuracy and a higher error rate in the existing system. The proposed method has been introduced to overcome the above-mentioned issues. For COVID-19 prediction, a Linear Decreasing Inertia Weight-based Cat Swarm Optimization with Half Binomial Distribution based Convolutional Neural Network (LDIWCSO-HBDCNN) approach is presented. In this suggested research study, the COVID-19 predicting dataset is employed as an input, and the min-max normalization approach is employed to normalize it. Optimum features are selected using Linear Decreasing Inertia Weight-based Cat Swarm Optimization (LDIWCSO) algorithm, enhancing the accuracy of classification. The Cat Swarm Optimization (CSO) algorithm's convergence is enhanced using inertia weight in the LDIWCSO algorithm. It is used to select the essential features using the best fitness function values. For a specified time across India, death and confirmed cases are predicted using the Half Binomial Distribution based Convolutional Neural Network (HBDCNN) technique based on selected features. As demonstrated by empirical observations, the proposed system produces significant performance in terms of f-measure, recall, precision, and accuracy.

**Keywords:** Binomial distribution; min-max normalization; Cat Swarm Optimization (CSO); COVID-19 forecasting



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

## 1 Introduction

The world is dealing with the Unique Corona Virus (COVID-19) epidemic, which reached in December 2019 in Wuhan, China [1–3]. From December 2019 onwards, most countries worldwide have experienced an increase in COVID-19 cases. Around 353 373 deaths and 5 596 550 cases were reported worldwide, as mentioned in the World Health Organization report dated May 29, 2020. People at old age, low immunity, and medical issues primarily related to lungs are highly prone to COVID 19 disease. Breathing problems, cold, and cough similar to flu are the significant symptoms of COVID-19 [4]. There are no effective therapeutics, drugs, and licensed vaccines for COVID-19 or Severe Acute Respiratory Syndrome Corona Virus 2 (SARS-CoV-2). For controlling the COVID-19 outbreak, various strategies are adopted by different countries because of the absence of pharmaceutical interventions. The most common method used is nationwide lockdown [5].

The local Wuhan government first used it on January 23, 2020, when the city was temporarily closed down to prohibit all public travel and quickly followed by many other cities in Hubei province.

Preserving social isolation is the best technique to reduce human-to-human broadcast of coronavirus disease in the absence of medications or specific antiviral for COVID-19, and therefore stringent lockdowns, quarantines, and curfews have also been incorporated by other nations.

The first coronavirus occurrence in India was reported on January 30, 2020, in the Thrissur region of Kerala, whenever a student returned from Wuhan, China's sprawling capital [6]. The Government of India enforced a complete national lockdown throughout the country on March 25, 2020, for 21 days and one day "Janata Curfew" on March 22, 2020, to fight the coronavirus SARS-CoV-2 epidemic in India. The government of India has prolonged the lockdown owing to massive coronavirus disease dissemination. Preventive steps for Covid19 are to protect oneself by constantly washing one's hands, not rubbing one's lips, nose and ears, and keeping social distance from other people (1 meter or 3 feet). Regularly, confirmed cases have risen in all countries. The transmission of such a virus is risky and needs stricter measures and plans already placed in place in many cities.

In hospitals, accurate COVID-19 cases forecasting is necessary, and it enhances the management strategies in managing infected patients in an optimum manner. Deep learning and machine learning have recently been considered as a leading study subject across a wide variety of applications, particularly in business and academia [7,8].

For predicting COVID-19 future, four algorithms, namely, Support Vector Machine (SVM), Exponential Smoothing (ES), Least Absolute Shrinkage and Selection Operator (LASSO) regression, and linear regression, are applied in [9], and these algorithms are supervised machine learning algorithms. In the prediction of death count, recoveries count, contaminated cases count, other models are outperformed by ES as shown. ES includes information in previous data prediction while managing time-series data. As a result, ES is a powerful tool. In more than 100 countries, for forecasting new and cumulative death and confirmed COVID 19 cases under different intervention conditions in real-time, a Modified Auto-Encoder (MAE) is developed in [10], and it is an Artificial Intelligence (AI) based technique. This technique does not provide satisfactory results concerning prediction accuracy and computation complexity.

In the latest days, advances in computer vision applications such as segmentation, identification, and object detection have indeed been made breakthroughs to the use of deep learning methods in general and convolutional neural networks (CNNs) in particular. Deep learning algorithms are very effective in automating feature-representation learning and removing the time-consuming effort of handcrafted feature engineering. Deep learning and convolutional neural networks (CNNs) use a hierarchical layer of feature representation to replicate the human visual cortex system in terms of structure and functioning. This multi-layer feature representation methodology allowed CNNs to automatically analyse different image features, allowing them to outperform feature approaches.

The main concern of this research work is the classification of COVID-19. Several methodologies have been introduced, but the COVID-19 identification accuracy is not significant. The existing approaches have drawbacks with error rates and inaccurate COVID dataset classification results. The LDIWCSO-HBDCNN method is proposed to improve the overall system performance. The main contribution of this research is data normalization, feature selection, and classification. The proposed technique for the given COVID dataset is employed to produce more accurate findings using effective algorithms.

The remainder of the article is organized as shown: The prediction model from past research is presented in Section 2. The proposed technique is presented in Section 3. The findings and performance analysis are presented in Section 4. Lastly, Section 5 concludes the study and discusses future research possibilities.

## 2 Literature Review

An enhanced Adaptive Neuro-Fuzzy Inference System (ANFIS) has been introduced by Al-Qaness et al. [11]. This system used the Salp Swarm Algorithm (SSA) and enhanced Flower Pollination Algorithm (FPA). For improving FPA, SSA is applied in general as FPA may get trapped with local optimum. The primary objective of this FPASSA-ANFIS model is to improve the ANFIS performance *via* the computation of ANFIS parameters by using FPASSA.

The official COVID 19 outbreak data given by WHO concerning confirmed cases in the upcoming ten days is used to evaluate this model. The FPASSA-ANFIS model shows significant performance based on the computing time, coefficient of determination (R<sup>2</sup>), RMSRE, MAPE, better performance.

Chimmula et al. [12] used a Long Short-Term Memory (LSTM) network to forecast the present COVID-19 epidemic in Canada and throughout the world. The University of Johns Hopkins and the Canadian Health Authority gathered COVID-19 data for this research and delivered several verified cases by March 31, 2020. The data collection includes the number of fatalities and restored patients every day.

Wavelet transformation is employed to maintain the time-frequency components while simultaneously reducing the random noise in the database. The Long Short-Term Memory (LSTM) networks are then utilized to assess future COVID-19 instances. The COVID-19 data is divided into a training set (80%) for training the models and a test set (20%) for testing the model's results. The system provided a greater prediction accuracy.

Castillo et al. [13] created a hybrid technique that combines the fractal dimension with fuzzy logic to enable effective and accurate COVID-19 time series prediction. The data sets from ten countries worldwide are utilized to build the fuzzy model using time series in a specified timeframe. Following that, further periods were employed to test the viability of the proposed approach for the predicted values of the ten countries. Given the difficulties of the COVID issue, the average forecast accuracy is 98% that may be considered good.

Hu et al. [14] developed the Artificial Intelligence (AI)-inspired algorithms for real-time Covid-19 prediction to assess the size, period, and end time of Covid-19 throughout China. A modified stacked auto-encoder was employed to represent epidemic transmission dynamics. This approach was employed to predict observed Covid-19 instances in China in real time. The database was obtained by WHO between January 11 and February 27, 2020.

The provinces/cities for transmission architecture investigation were organized using latent variables from the auto-encoder and clustering methods. Utilizing multiple-step predicting, the computed average errors of 6-step, 7-step, 8-step, 9-step, and 10-step predictions were 1.64%, 2.27%, 2.14%, 2.08%, and 0.73%, correspondingly.

Elsheikh et al. [15] suggested a deep learning-based prediction system for COVID-19 pandemic in Saudi Arabia. The Long Short-Term Memory (LSTM) network forecasts the total reported cases, total retrieved cases, and total deaths in Saudi Arabia in this investigation as a complete DL model. The model was educated based on the official records. The ideal settings of the model's parameters that optimize

predicting precision have been identified. Seven statistical assessment parameters, namely Coefficient of Residual Mass (CRM), coefficient of variation (COV), Overall Index (OI), Efficiency Coefficient (EC), MAE, coefficient of determination ( $R^2$ ) and RMSE are employed to forecast the model's accuracy.

Kapoor et al. [16] developed a novel forecasting technique for COVID-19 instance detection employing Graph Neural Networks and mobility data. The proposed method is based on a single large-scale Spatio-temporal graph with nodes representing regional human mobility, spatial edges representing inter-regional connection based on human mobility, and temporal edges representing node attributes across time. This technique is tested on the COVID-19 database at the county level in the United States, demonstrating that the graph neural network's rich geographical and temporal data allows the model to learn complicated dynamics. Tab. 1 shows the comparative analysis of existing Covid -19 forecasting approaches. Around 6% minimization in Root Mean Squared Log Error (RMSLE) and improvement in absolute Pearson Correlation to 0.998 from 0.9978 can be achieved as shown in results when compared with baseline models.

**Table 1:** Comparative analysis of existing Covid -19 forecasting approaches

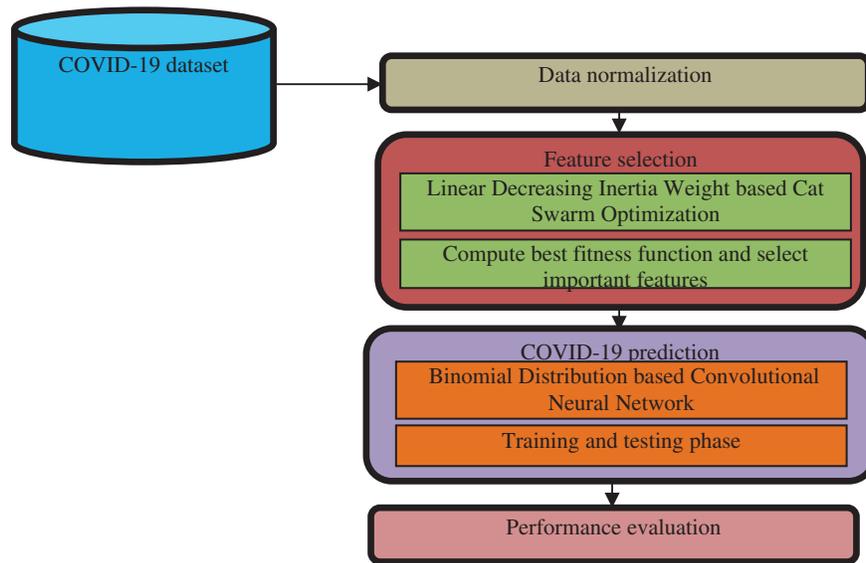
S. No	Authors name	Methods name	Merits	Demerits
1.	Al-Qaness et al. (2020)	Improved Adaptive Neuro-Fuzzy Inference System (ANFIS)	Higher classification accuracy and recall	It has Computational complexity
2.	Chimmula and Zhang (2020)	Long Short-Term Memory (LSTM)	No need of parameter tuning	LSTMs take longer to train LSTMs are easy to overfit
3.	Castillo and Melin (2020)	Fractal dimension and fuzzy logic scheme	It achieves 98% of forecasting average accuracy	Need to update the rules regularly
4.	Hu et al. (2020)	Modified stacked auto-encoder	Higher True positive rate	It required joint training
5.	Elsheikh et al. (2021)	Long Short-Term Memory (LSTM) network	Acceptable. RMSE and MAE)	It has sensitive to different random weight initializations
6.	Kapoor et al. (2020)	Graph Neural Network	It achieves absolute Pearson Correlation development from 0.9978 to 0.998.	The difficulty of showing the problem to the network

### 3 Proposed Methodology

For forecasting COVID-19, Linear Decreasing Inertia Weight based Cat Swarm Optimization with Half Binomial Distribution based Convolutional Neural Network (LDIWCSO-HBDCNN) technique is proposed in this research work. Fig. 1 shows the proposed work's flow diagram.

#### 3.1 Input

The input corresponds to the COVID-19 forecasting dataset.



**Figure 1:** Suggested system flow diagram

From <https://www.kaggle.com/davidbnn92/weather-data-for-covid19-data-analysis>, this dataset is collected. Information like fog, temperature, fatalities, confirmed cases, date, long, lat, region, state, id are added in this dataset. The weather data is obtained from the National Oceanic and Atmospheric Administration & Global Surface summary of the Day (NOAA GSOD) dataset. Recent measurements are included using continuous updates.

**3.2 Data Normalization Using Min-Max Normalization**

A mathematical function converts the numerical values into new range values in the normalization process. The covid-19 dataset is normalized using min-max normalization in this proposed research work. Data are generally normalized using a standard technique called min-max normalization. From the dataset, within the specified maximum and minimum range, values are normalized, and using the following expression, every value is replaced.

$$v' = \frac{v - \min_A}{\max_A - \min_A} (new\_max_A - new\_min_A) + new\_min_A \tag{1}$$

where,

A indicates Attribute data,

Min(A) is a minimum absolute value of A, Max(A) is a maximum absolute value of A

v' indicates every entry's new value in data

v indicates every entry's old value in data

new\_max(A) indicated maximum value of range, new\_min(A) indicated min value of range (i.e, mandatory boundary value series)

**3.3 Feature Selection Using Linear Decreasing Inertia Weight Based Cat Swarm Optimization (LDIWCSO) Algorithm**

The Linear Decreasing Inertia Weight based Cat Swarm Optimization (LDIWCSO) algorithm selects normalized features. A new Cat Swarm Optimization algorithm has been developed according to the cat's

behavior. This algorithm has two search modes, Tracing Mode (TM) and Seeking Mode (SM). A cat's capacity to keep aware of its environment when at rest is modelled in seeking mode. Tracing and catching the targets of cats is emulated in tracing mode. Optimization problems are solved using these modes.

Features are given as input in this proposed research work. Here, every feature is assigned with a fitness value, velocity, and position in a specified population. A point with respect to possible feature set represented by position. In a  $D$ -dimensional space, distance variance is represented using velocities. The solution set's quality is represented using fitness value. Fitness function corresponds to classification accuracy in this proposed work.

$$\text{Fitness function} = \text{Max}(\text{Accuracy}) \quad (2)$$

Features are distributed to tracing or seeking mode randomly by CSO in every generation. Then altered every feature's position. At last, the feature's position with the highest accuracy value indicates the best feature. Fig. 2 shows Cat Swarm Optimization (CSO) system.

#### *CSO Process*

The cat's percentage (features) distributed to tracing mode is represented using Mixture Ratio (MR). As much time is spent on testing by cats, the MR value will be minimal. Following outlines this process as

- (1) Within a  $D$ -dimensional space,  $N$  features are initialized with velocities and positions.
- (2) Features are distributed to tracing or seeking mode; features count in two modes defines MR.
- (3) Every feature's accuracy value is measured.
- (4) In every iteration, they are searched based on the cat's mode. The following subsection, describes the process involved in these two modes.
- (5) If the terminal requirements are met, the algorithm is terminated; it returns to step (2) for the next cycle.

#### *Seeking Mode*

There are four operations in seeking mode; namely, Counts of Dimension to Change (CDC), Seeking Range of the selected Dimension (SRD), Self-Position Consideration (SPC), and Seeking Memory Pool (SMP). Seeking memory's pool size is defined by SMP. For instance, the SMP value of 5 represents that every cat can store 5 result sets as candidates.

SPC is a Boolean value. One position within a memory retains the current solution set if SPC is true and not changed. Seeking range's minimum and maximum values are defined by SRD. In the seeking process, dimensions count to be changed is represented using CDC. The steps of the seeking process are discussed below.

- (1) Current feature position's SMP copies are generated. One of these copies retains the current feature position if SPC is true and becomes a candidate immediately. Before becoming candidates, other features need to be changed. Otherwise, searching is performed by all SMP copies and its results in its position change.
- (2) By altering CDC's dimension percent, the position of every copy to be changed is changed randomly. The CDC's dimensions percent is selected initially. The SRD percent's current value is decreased or increased by changing every dimension chosen randomly. After changes are made, copies become a candidate.
- (3) Every candidate's fitness value is computed through the fitness function.
- (4) The likelihood of each candidate being chosen is calculated. The chosen probability ( $P_i$ ) equals one, and if all candidates have the same fitness value, else, using expression (3),  $P_i$  is computed. Range of  $i$  lies between 0 and SMP;  $P_i$  gives candidates selection probability;  $P_i$  Overall fitness values

maximum value is represented as  $FS_{max}$ , and minimum value is represented as  $FS_{min}$ ;  $\Phi\Sigma_t$  gives candidates fitness value  $FS_i$

If the objective is to find a solution set having maximum fitness

Value, then  $FS_b = FS_{min}$ ; else  $FS_b = FS_{max}$ . As a result of this function's probability computation, better candidates have a larger chance of getting chosen, and vice versa, as shown below:

$$P_i = \frac{|FS_i - FS_b|}{FS_{max} - FS_{min}} \tag{3}$$

(5) One candidate is selected randomly based on selected probability (Pi). The current feature is moved to this position after choosing the candidate.

Tracing Mode. Cats target tracing is represented using tracing mode. Here the same strategy is followed.

(1) Expression (4) is used for updating velocities.

(2) Current feature's position is updated using expression(5) as mentioned below:

$$v_{k,d}^{t+1} = v_{k,d}^t + r_1 \times c_1 \times (x_{best,d}^t - x_{k,d}^t) \tag{4}$$

$d = 1, 2, \dots, D,$

$$x_{k,d}^{t+1} = x_{k,d}^t + v_{k,d}^t \tag{5}$$

where,

$x_{k,d}^t$  and  $v_{k,d}^t$  are current feature  $k$ 's position and velocities at iteration  $t$

$x_{best,d}^t$  gives the best solution set from *feature k* in population

$c_1$  is constant

$r_1$  is an arbitrary number that lies between 0 and 1

*Linear Decreasing Inertia Weight based Cat Swarm Optimization (LDIWCSO) Algorithm*

One of the most effective algorithms for computing the best global solution is conventional CSO. But, for producing an acceptable solution, CSO may consume a long time in rare cases. So, the algorithm's convergence and performance may get affected by this. A new parameter, inertia weight ( $w$ ), is added in the expression used for updating positions to solve this issue. An algorithm's tracing mode, a new velocity update form is used. A Linear Decreasing Inertia Weight factor is introduced in this work and it has a linear decrement to  $w_{min}$  from  $w_{max}$  with respect to iteration's increment and they indicates final and initial values. Reformed expressions are mentioned below.

$$v_{k,d}^{t+1} = w * v_{k,d}^t + r_1 \times c_1 \times (x_{best,d}^t - x_{k,d}^t) \tag{6}$$

$$W = w_{max} - \frac{w_{max} - w_{min}}{iter_{max}} \times k \tag{7}$$

$$v_{k,d}^{t+1} = \left( w_{max} - \frac{w_{max} - w_{min}}{iter_{max}} \times k \right) * v_{k,d}^t + r_1 \times c_1 \times (x_{best,d}^t - x_{k,d}^t) \tag{8}$$

where,

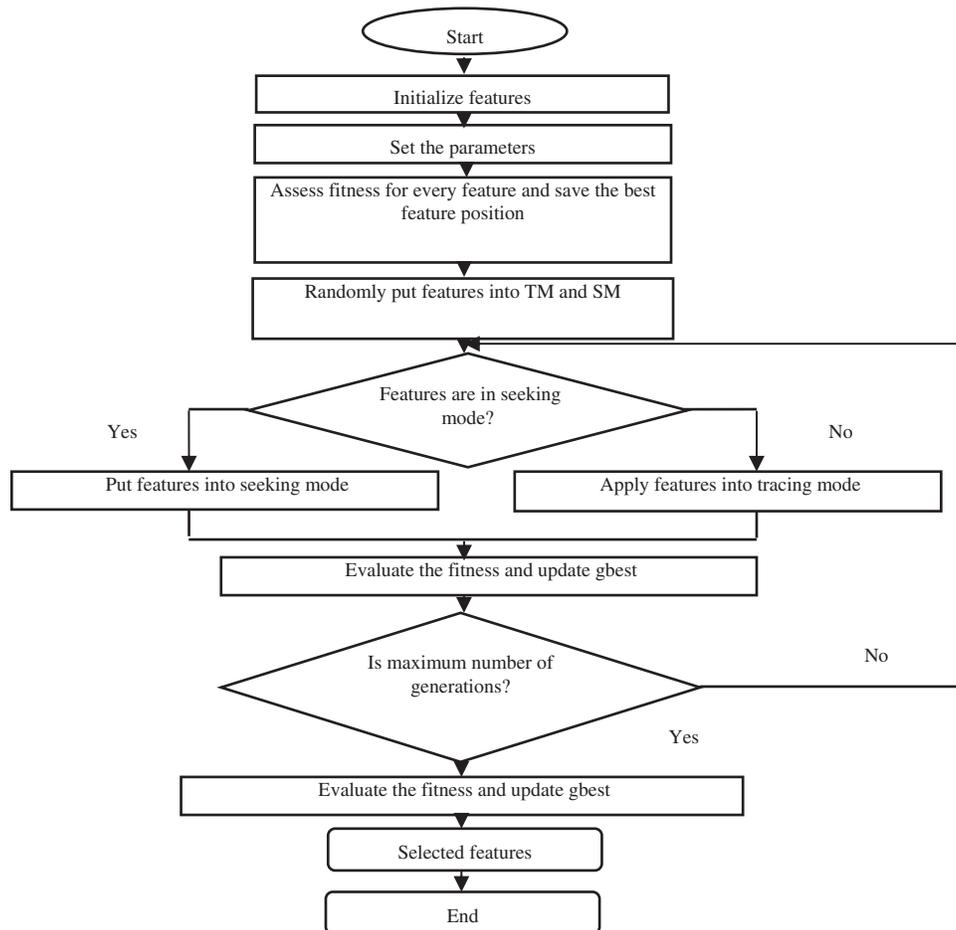
$w$  gives inertia weight

$w_{max}$  represents maximum weight value

$w_{min}$  represents minimum weight value

$iter_{max}$  indicates maximum iteration

$k$  is a Constant



**Figure 2:** Flow diagram of the cat swarm optimization (CSO) method

---

**Algorithm 1:** Cat Swarm Optimization (CSO) Algorithm

---

1. Start
  2. In dataset, features count is initialized
  3. Velocity and parameters are initialized
  4. While ( $t \leq it_{max}$ )
  5. For every cat
  6. Fitness value is computed
  7. Next, best cat is selected and saved into memory.
  8. If feature ( $X_i$ ) = *seeking mode*
  9. Seeking process is initiated
- 

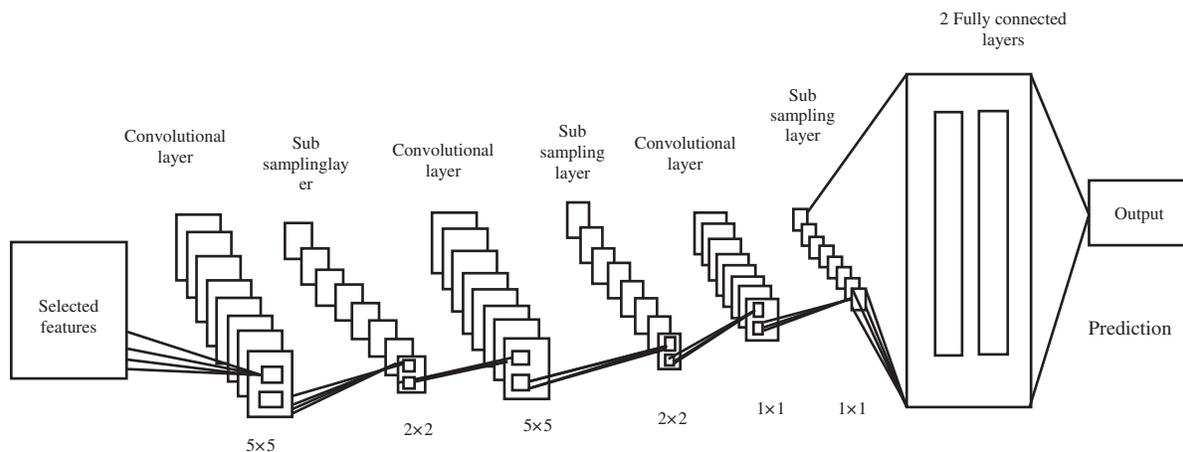
(Continued)

**Algorithm 1 (continued).**

10. Else
11. Tracing process is initiated
12. Termination condition is checked; if satisfied; program is terminated; otherwise, Step 4 to 11 are repeated.
13. Fitness function is computed and cat with best finest value is preserved.
14. Selected features
15. Stop

**3.4 Classification Using Half Binomial Distribution Based Convolutional Neural Network (HBDCNN)**

This research work uses Half Binomial Distribution based Convolutional Neural Network (HBDCNN) for classification. The CNN [17,18] is a robust deep learning algorithm. This network shows excellent efficiency in different areas, especially in computer vision. There are three types of layers in CNN: fully connected, sub-sampling, and convolution layers [19,20]. A convolutional neural network's (CNN) typical structure is depicted in Fig. 3. The following section explains every layer type.



**Figure 3:** Convolutional neural network

*Convolution Layer*

selected features are given as input in this proposed work. With a kernel (filter), these input features are convolved in this convolution layer. The  $n$  output features are generated using this convolution result between kernel and input feature. In general, the filter is a convolution matrix kernel. The feature map with  $i \times i$  size refers to output features that convolve input and kernel.

Multiple convolutional layers are present in the CNN architecture. The feature vector is given as an input to the subsequent convolutional layers, and it produces the same at the output. In every convolution layer, there is filter bunch  $n$ . The input is convolved with all these filters, and the number of filters in the convolution process is equal to depth ( $n^*$ ) of the resulting feature map. Every filter map is a specific feature at a given input point.

The  $l$ -th convolution layer's output is represented as  $C_i^{(l)}$ , and it has feature maps. It is expressed as,

$$C_i^{(l)} = B_i^{(l)} + \sum_{j=1}^{a_i^{(l-1)}} K_{ij}^{(l-1)} * C_j^{(l-1)} \quad (9)$$

where, bias matrix is given by  $B_i^{(l)}$ , kernel or convolution filter is represented as  $K_{ij}^{(l-1)}$  and its size is  $a^*a$ . In the same layer, the  $i$ -th feature map is connected with the layer  $(l-1)$ . Feature maps are available in output  $C_i^{(l)}$  layer. In expression (10), input space corresponds to first convolutional layer  $C_i^{(l-1)}$ , that is  $C_i^{(0)} = X_i$ . A feature map is generated using the kernel. For convolutional layer output's nonlinear transformation, activation function is applied after convolution layer.

$$Y_i^{(l)} = Y(C_i^{(l)}) \quad (10)$$

where,

$Y_i^{(l)}$  represents activation function's output

$C_i^{(l)}$  gives the input that it receives.

The rectified linear units (ReLU), tanh, and sigmoid normally utilize activation functions. The ReLU activation function is employed here and it is represented by,

$$Y_i^{(l)} = \max(0, Y_i^{(l)}) \quad (11)$$

This function is normally employed in DL models to reduce nonlinear effects and interactions. When given a negative input, ReLU converts the output to 0, and when given a positive input, it returns the same input. Faster training is a significant advantage of this function compared with other functions. The error derivative will be very small in the saturation region in this function. So, there will be weight update vanishing. This is termed a vanishing gradient problem.

#### *Sub Sampling or Pooling Layer*

The extracted feature map's dimensionality from the previous convolution layer is reduced spatially using this layer. This is a significant objective of this layer. Between feature maps and masks, this sub-sampling operation is performed. Various sub-sampling techniques like maximum pooling, sum pooling, averaging pooling are produced. Beyond convex hull of inputs  $\{X_p^c\}_{p \in R_q}$ ,  $Y_q^c$  is produced for CNN's better recognition.

Here, for simplicity, subscript  $q$  (output position) and superscript  $c$  (channel) are omitted. First, a binomial distribution is used for modeling local neuron activations  $\{X_p\}_{p \in R}$ , and this distribution has a standard deviation  $\sigma_x$  and mean  $\mu_x$ .

$$\bar{X} \tilde{N}(\mu_x, \sigma_x) \iff \bar{X} = \mu_x + \varepsilon \sigma_x \quad (12)$$

where,

$$\mu_x = n X_S$$

$$\sigma_x^2 = (X_S - \mu_x)^2$$

$$\varepsilon \tilde{N}(0, 1), \varepsilon \in (-\infty, +\infty)$$

$N$  indicates iterations count

$X_S$  represents input features successfully beyond specific class

The expression (12) is modified according to the local pooling knowledge and it restrict the output to fall below mean  $\mu_x$  as,

$$Y = \mu_x + |\varepsilon|\sigma_x, \varepsilon \sim N(0, 1) \Leftrightarrow Y = \mu_x + \eta \sigma_x, \eta \sim N_h(1), \eta \in [0, +\infty) \quad (13)$$

The prior probabilistic model corresponds to half- binomial distribution  $N_h(\sigma_0)$  and has  $\sigma_0 = 1$ . For producing Y stochastically without the use of pooling parameter, formulated the fixed half- binomial pooling as shown in expression (13).

#### *Fully Connected Layer*

The CNN's final layer is a traditional feed forward network and there may be more than one hidden layers. The output layer employs the Softmax activation function..

$$Y_i^{(l)} = f(z_i^{(l)}) \quad (14)$$

where

$$z_i^{(l)} = \sum_{j=1}^{m_i^{(l-1)}} w_{ij}^{(l)} y_j^{(l-1)} \quad (15)$$

where, weights are represented as  $w_{ij}^{(l)}$ , and for making every class representation, they are tuned using a complete fully connected layer and transfer function is indicated by  $f$  and nonlinearity is represented using this. Nonlinearity is incorporated into the neurons of the fully connected layer. In pooling and convolution layers, it is built separately. In a specific period across India, death cases are confirmed using Half Binomial Distribution based Convolutional Neural Network (HBDCNN) output.

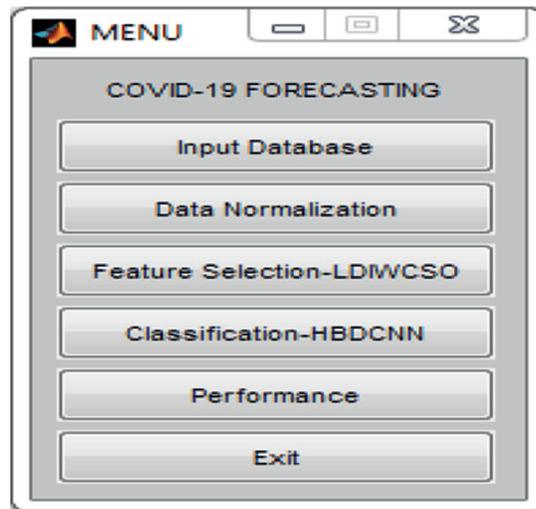
## 4 Experimental Results

Experimental evaluation is conducted for proposed and available research techniques in MATLAB. From <https://www.kaggle.com/davidbnn92/weather-data-for-covid19-data-analysis>, collected the COVID-19 forecasting data in the research work. The available CNN, LSTM and suggested HalfBinomial Distribution Convolutional Neural Network (HBDCNN) performance are compared in terms of error rate, f-measure, recall, precision, accuracy values.

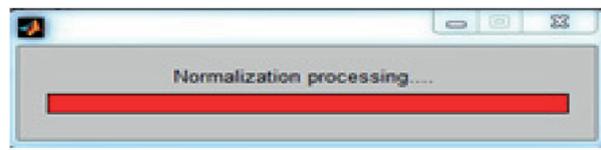
Fig. 4 shows the process of the proposed LDIWCSO-HBDCNN process. COVID-19 forecasting database is given as input in this proposed research work. The min-max normalization technique is used to normalize this input. Fig. 5 shows this process. Optimum features are selected using Linear Decreasing Inertia Weight based Cat Swarm Optimization (LDIWCSO) algorithm and it is illustrated in Fig. 6.

In the prediction of death and confirmed cases across India, the training process of the neural network is represented in Fig. 7. Further, the decision may be incorrect (false) or correct (true), which is illustrated in Fig. 8. So, the decision is false under anyone the following four possible classes, False-Negative (FN), False-Positive (FP), True-Negative (TN) and True-Positive (TP).

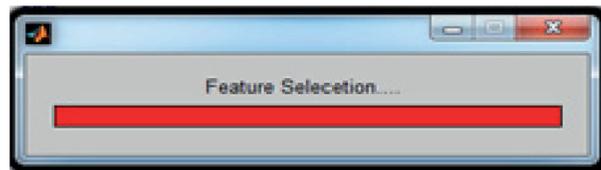
The performance of the existing CNN, LSTM models, and the proposed Half Binomial Distribution Convolutional Neural Network (HBDCNN) are compared in terms of error rate, f-measure, recall, precision, accuracy values are represented in Tab. 2.



**Figure 4:** Lineardecreasing inertia weight based cat swarm optimization with half binomial distribution based convolutional neural network method



**Figure 5:** Normalization process



**Figure 6:** Feature selection

#### 4.1 Accuracy

Accuracy is a simple ratio of accurately forecasted observations to total observations. Therefore, it is a nearly straightforward performance metric.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}} \quad (16)$$

The Proposed HBDCNN technique and available CNN and LSTM techniques are compared in terms of accuracy, and its results are shown in Fig. 9. The LDIWCSO algorithm is used in this work for selecting optimum features. For covid-19 prediction, HBDCNN is utilized here, and this prediction is made using selected features. This produces better accuracy. Proposed HBDCNN delivers around 92.57% accuracy value, CNN attains 90.74%, and 87.92% of accuracy is produced by LSTM as indicated in experimental results. Compared with CNN and LSTM techniques, higher accuracy value is produced by the proposed HBDCNN, as shown in test findings.

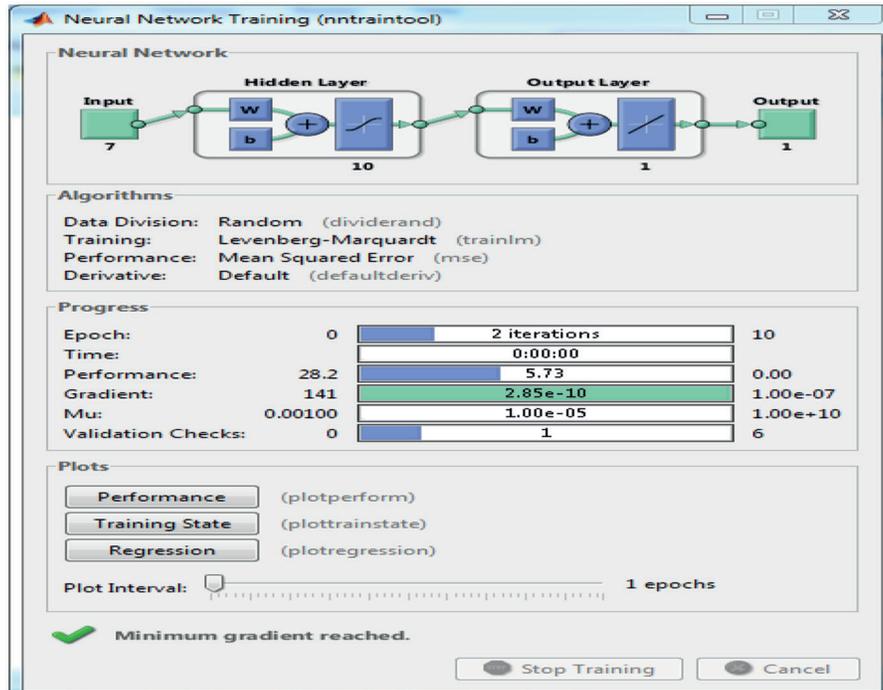


Figure 7: Neural network training

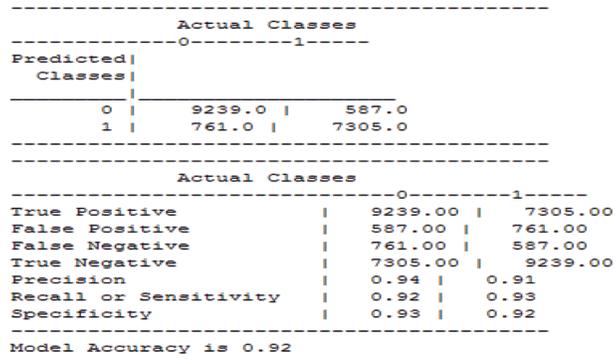
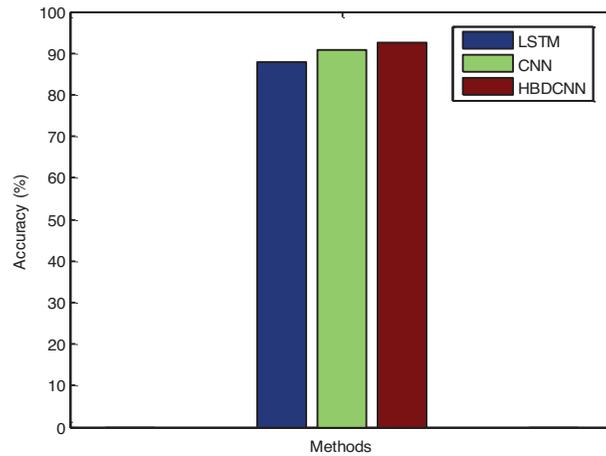


Figure 8: Performance outcome

Table 2: Performance analysis

Methods/metrics	LSTM	CNN	HBDCNN
Accuracy	87.9220	90.7445	92.5777
Precision	87.7028	90.5500	92.4192
Recall	87.8875	90.7531	92.5654
F-measure	87.7950	90.6514	92.4922
Error rate	12.0780	9.2555	7.4223



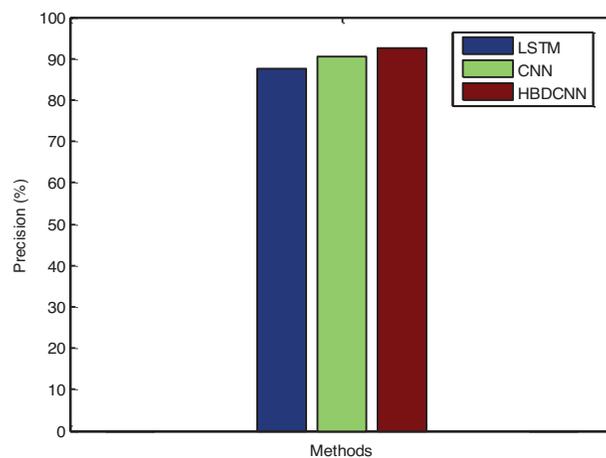
**Figure 9:** Accuracy comparison

#### 4.2 Precision

Precision value is defined as the ratio of accurately classified as positive observations to total expected positive observations and it is expressed as,

$$Precision = \frac{TP}{TP + FP} \quad (17)$$

The Proposed HBDCNN technique and available CNN and LSTM techniques are compared in terms of precision, and its results are shown in Fig. 10. The methods are plotted on the x-axis, and for the y-axis, precision values are plotted. Proposed HBDCNN produces around 92.41% precision value, CNN produces 90.55%, and 87.70% is produced by LSTM as indicated in experimental results. When compared with available techniques, a high precision value is produced by proposed HBDCNN as in test findings.



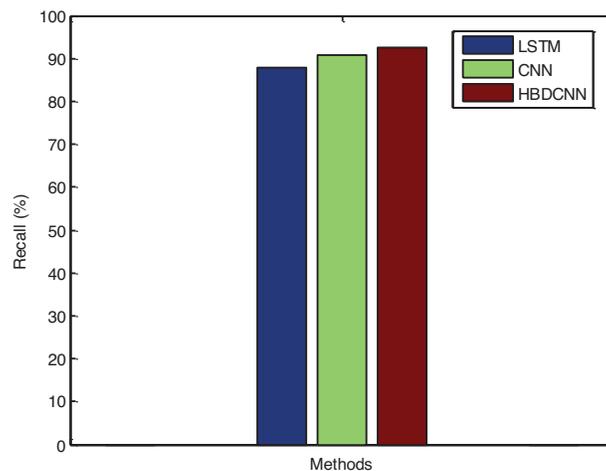
**Figure 10:** Precision comparison

### 4.3 Recall

In covid-19 prediction, recall is a ratio of actual true instances that are retrieved successfully.

$$Recall = \frac{TP}{TP + FN} \quad (18)$$

The Proposed HBDCNN technique and existing CNN and LSTM techniques are compared in terms of recall, and its results are shown in Fig. 11. It is observed from the graphical evaluation that the proposed HBDCNN attains 92.56% recall value, whereas the existing CNN and LSTM produced 90.75% and 87.88% recall results. Thus, the proposed work outperformed the existing techniques in terms of Recall.



**Figure 11:** Recall comparison

### 4.4 F1 Score

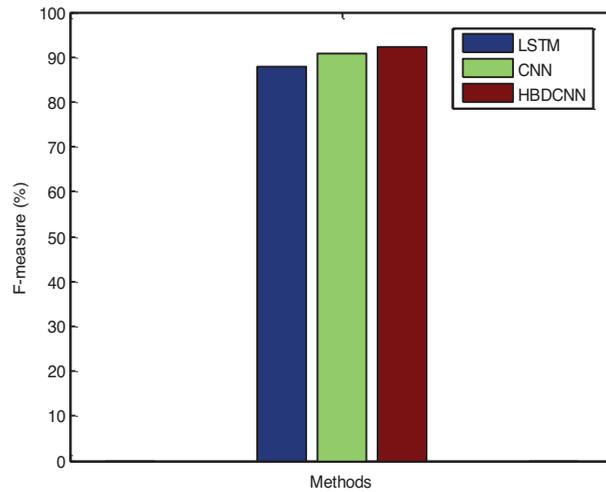
Recall and precision's weighted average produces F1 score. Classifier performance can be rated using this as a statistical measure. False negatives and false positives are taken into account for this score. F1 score is expressed as,

$$F - measure = 2 \frac{Precision * Recall}{Precision + Recall} \quad (19)$$

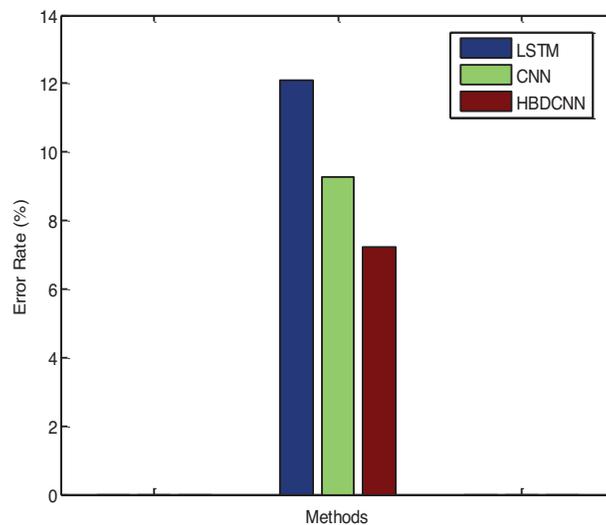
The Proposed HBDCNN technique and available CNN and LSTM techniques are compared in terms of F-measure, and its results are shown in Fig. 12. The graphical results showed that the proposed HBDCNN approach attains 92.49% F-measure. Alternatively, the existing CNN and LSTM approach attained 90.65% and 87.79% F-measure respectively. Compared with the existing techniques, a higher f-measure value is produced by the proposed HBDCNN, as shown in experimental results.

### 4.5 Error Rate

Error rate performance of LSTM, CNN and HBDCNN techniques are illustrated in Fig. 13. The error rate of the proposed HBDCNN approach is observed to be 7.4%. Alternatively, the error rate attained by CNN is 9.25% and that of LSTM is observed to be 12.07%. Thus, it is clearly observed that the proposed approach resulted in significant performance with a minimal error rate compared to the existing CNN and LSTM techniques.



**Figure 12:** F-measure comparison



**Figure 13:** Error rate comparison

## 5 Conclusion

Linear Decreasing Inertia Weight based Cat Swarm Optimization with Half Binomial Distribution based Convolutional Neural Network (LDIWCSO-HBDCNN) is proposed in this work to predict death and confirmed COVID cases. Data is normalized initially using min-max normalization. LDIWCSO algorithm selects the optimum features, which will enhance classification accuracy. The HBDCNN is used for predicting covid-19 according to selected features. Accuracy value around 92.57%, precision value about 92.41%, recall value about 92.56%, f-measure results about 92.49%, and error rate value about 7.4% is obtained using the proposed system as shown in experimental results. The future enhancement of this work would be to use various recent optimization algorithms like Harmony Search (HS) and analyze its performance.

**Acknowledgement:** We thank anonymous referees for their helpful suggestions.

**Funding Statement:** The authors received no specific funding for this study.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

- [1] A. H. Elsheikh, A. I. Saba, M. Abdelaziz, S. Lu, S. Shanmugan *et al.*, “Deep learning-based forecasting model for COVID-19 outbreak in saudi Arabia,” *Process Safety and Environmental Protection*, vol. 149, no. 11, pp. 223–233, 2021.
- [2] N. Chintalapudi, G. Battineni and F. Amenta, “COVID-19 virus outbreak forecasting of registered and recovered cases after sixty day lockdown in Italy: A data driven model approach,” *Journal of Microbiology, Immunology and Infection*, vol. 53, no. 3, pp. 396–403, 2020.
- [3] A. F. Lukman, R. I. Rauf, O. Abiodun, O. Oludoun, K. Ayinde *et al.*, “COVID-19 prevalence estimation: Four most affected African countries,” *Infectious Disease Modelling*, vol. 5, no. 10, pp. 827–838, 2020.
- [4] C. H. Yan, F. Faraji, D. P. Prajapati, C. E. Boone and A. S. DeConde, “Association of chemosensory dysfunction and Covid-19 in patients presenting with influenza-like symptoms,” *International Forum of Allergy & Rhinology*, vol. 10, no. 7, pp. 806–813, 2020.
- [5] D. Almaghaslah, G. Kandasamy, M. Almanasef, R. Vasudevan and S. Chandramohan, “Review on the coronavirus disease (COVID-19) pandemic: Its outbreak and current status,” *International Journal of Clinical Practice*, vol. 74, no. 11, pp. 01–09, 2020.
- [6] S. Khajanchi and K. Sarkar, “Forecasting the daily and cumulative number of cases for the COVID-19 pandemic in india,” *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 30, no. 7, pp. 01–17, 2020.
- [7] S. He, S. Tang and L. Rong, “A discrete stochastic model of the COVID-19 outbreak: Forecast and control,” *Mathematical Biosciences and Engineering*, vol. 17, no. 4, pp. 2792–2804, 2020.
- [8] İ. Kırbaş, A. Sözen, A. D. Tuncer and F. Ş. Kazancıoğlu, “Comparative analysis and forecasting of COVID-19 cases in various European countries with ARIMA, NARNN and LSTM approaches,” *Chaos Solitons & Fractals*, vol. 138, no. 9, pp. 1–7, 2020.
- [9] F. Rustam, A. A. Reshi, A. Mehmood, S. Ullah, B. On *et al.*, “COVID-19 future forecasting using supervised machine learning models,” *IEEE Access*, vol. 8, no. 5, pp. 01–11, 2020.
- [10] Z. Hu, Q. Ge, S. Li, L. Jin and M. Xiong, “Evaluating the effect of public health intervention on the global-wide spread trajectory of Covid-19,” *medRxiv*, pp. 1–17, 2020.
- [11] M. A. Al-Qaness, A. A. Ewees, H. Fan and M. Abd El Aziz, “Optimization method for forecasting confirmed cases of COVID-19 in china,” *Journal of Clinical Medicine*, vol. 9, no. 3, pp. 01–15, 2020.
- [12] V. K. R. Chimmula and L. Zhang, “Time series forecasting of COVID-19 transmission in Canada using LSTM networks,” *Chaos Solitons & Fractals*, vol. 135, no. 5, pp. 01–06, 2020.
- [13] O. Castillo and P. Melin, “Forecasting of COVID-19 time series for countries in the world based on a hybrid approach combining the fractal dimension and fuzzy logic,” *Chaos Solitons & Fractals*, vol. 140, no. 8, pp. 01–12, 2020.
- [14] H. Zixin, Q. Ge, S. Li, L. Jin and M. Xiong, “Artificial intelligence forecasting of covid-19 in china,” arXiv preprint arXiv:2002.07112, pp. 1–20, 2020.
- [15] A. H. Elsheikh, A. I. Saba, M. Abdelaziz, S. Lu, S. Shanmugan *et al.*, “Deep learning-based forecasting model for COVID-19 outbreak in saudi Arabia,” *Process Safety and Environmental Protection*, vol. 149, no. 11, pp. 223–233, 2021.
- [16] A. Kapoor, X. Ben, L. Liu, B. Perozzi, M. Barnes *et al.*, “Examining covid-19 forecasting using spatio-temporal graph neural networks,” in *Conf. '20*, San Diego, CA, USA, pp. 1–6, 2020.
- [17] S. Albawi, T. A. Mohammed and S. Al-Zawi, “Understanding of a convolutional neural network,” in *Int. Conf. on Engineering and Technology (ICET)*, Antalya, Turkey, pp. 1–6, 2017.
- [18] M. Sarıgül, B. M. Ozyildirim and M. Avci, “Differential convolutional neural network,” *Neural Networks*, vol. 116, no. 5, pp. 279–287, 2019.
- [19] S. Indolia, A. K. Goswami, S. P. Mishra and P. Asopa, “Conceptual understanding of convolutional neural network—a deep learning approach,” *Procedia Computer Science*, vol. 132, no. 1, pp. 679–688, 2018.
- [20] T. Goel, R. Murugan, S. Mirjalili and D. K. Chakrabarty, “OptCoNet: An optimized convolutional neural network for an automatic diagnosis of COVID-19,” *Applied Intelligence*, vol. 51, no. 3, pp. 1351–1366, 2021.