Research Article

International Journal of Advanced Technology and Engineering Exploration, Vol 10(102)
ISSN (Print): 2394-5443 ISSN (Online): 2394-7454
http://dx.doi.org/10.19101/IJATEE.2022.10100070

Additive LOG transformation distributed feature embedding convolutional neural learning classifier for early COVID-19 prediction

Kalaiselvi S R[1*] and Vijayabhanu R[2]
Research Scholar, Department of Computer Science, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, Tamil Nadu, India[1]
Assistant Professor, Department of Computer Science, Dr.N.G.P. Arts and Science College, Coimbatore, Tamil Nadu, India[1]
Assistant Professor(SG), Department of Computer Science, Avinashilingam Institute for Home Science and Higher Education for Women, Coimbatore, Tamil Nadu, India[2]

Received: 27-August-2022; Revised: 17-May-2023; Accepted: 20-May-2023

Abstract
The COVID-19 pandemic is a severe disease that has claimed many lives. It is crucial to reduce the mortality rate and take essential steps to provide suitable treatment. This allows the government to strategize and control the spread of the disease or at the very least, uplift the morale of the general public. To classify patients' input and their medical files, various learning methods have been introduced to facilitate COVID-19 prediction. However, due to the extensive dataset, it took a considerable amount of time to train the program, resulting in ineffective predictions, higher infection rates, increased spread, and elevated death rates. The main objective of this research is to accurately predict COVID-19 at an earlier stage and in less time using the additive log transformed distributive feature embedding time-dependent regressive convolutional neural learning classifier (ALTDFETRCNLC). Initially, patient files are collected as input for the dataset. The additive log ratio is transformed using one hot encoding to preprocess and normalize the input data. The Tversky similarity indexed stochastic distributive feature embedding technique is employed to select relevant features efficiently. Finally, the Levenberg-Marquardt convolutional neural learning classifier is utilized to classify COVID-19 predictions. This approach has significantly improved prediction accuracy and considers space complexity. Experimental evaluation is conducted using the proposed ALTDFETRCNLC technique and existing methods, utilizing the COVID-19 dataset with different metrics. The results demonstrate that the ALTDFETRCNLC technique outperforms contemporary and conventional works in terms of prediction accuracy, precision, recall, and F-measure, showing improvements of 4%, 4%, 3%, and 3% respectively. Additionally, the ALTDFETRCNLC technique achieves faster prediction times with an 8% improvement and reduces the error rate and space complexity by up to 8% and 9% compared to existing methods.

Keywords
COVID-19 prediction, Additive log ratio, Tversky similarity index, Time-dependent cox regressive Levenberg–Marquardt convolutional neural learning classifier.

1.Introduction
In the field of healthcare real-time and accurate results plays a vital role. COVID-19 posed itself to not validate the said statement. In today's scenarios pandemic has shaken the entire healthcare domain. With so many uncertainties around, scientists are working hard to ensure better results are acquired to decrease virus extent. So far, the manual processing has not been efficient, therefore, this research work is used for explaining the learning-based COVID-19 prediction.

A deep learning method on long short-term memory (LSTM) ensemble scheme was developed for identifying COVID-19 establishment and the death cases over the world [1]. In this method, the performance of disease prediction time consumption was not focused on. A novel susceptible, infected, recovered, vaccinated, and deceased–deep learning method (SIRVD-DL) was developed in [2] for effective COVID-19. In this, the accuracy performance of COVID-19 was not analyzed effectively. A deep learning method with a convolutional neural network (CNN) and stacked bidirectional gated recurrent unit (Bi-GRU) was

*Author for correspondence

introduced in [3] for identifying and detecting COVID-19. Over here, the performance of time to curb COVID-19 was not minimized. Multi-task Gaussian process (MTGP) regression approach was introduced in [4] using a better forecast of novel COVID-19. This method failed to analyze the deep feature learning for improving the model efficiency. A deep-LSTM ensemble model was developed in [5] to predict COVID-19 with better accuracy. This also was not sufficient to provide better outcomes in a minimum time. A hybrid approach was developed in [6] for COVID-19 prediction. A depth wise separable convolution neural network (DWS-CNN) was introduced in [7] using a deep support vector machine (DSVM) for COVID-19 detection. This was not efficient to attain maximum classification results. An adaptive neuro-fuzzy inference system (ANFIS) was intended by [8] for the early detection of coronavirus disease. This system failed to apply to the large volume of data samples. Integration of three deep learning models was developed in [9] for predicting COVID-19 disease. Even this designed method was unsuccessful in selecting useful features over time and incorporating them within deep learning schemes. Stacking an ensemble with deep neural networks was developed by [10] to predict post-COVID-19 infection. However, it failed to implement efficient feature selection methods using better dimensional data.

1.1Challenges of the previous literature and Motivation of the work

The COVID-19 pandemic is one of the most significant challenges faced by humanity in recent years [2]. By only employing these methods for the prediction cannot capture the time changing pattern of the transmission of this infectious diseases. Researchers have developed various methods for predicting the course of the disease, but analyzing accurate forecasting of its duration has become a critical and challenging task [4]. The reasons can be listed as lack of knowledge, cost of the research, and societal-governmental factors that can highly influence the newly born disease. Any forecasting can play an essential role with little reliability, as there are no real-time data samples available and makes it challenging to predict the spreading of the disease. It should also be understood that any forecasting model can make a big difference in this stage. Therefore, it is essential to find the best machine learning-based prediction model that can provide better forecasting with limited training data. In response to this challenge, a novel model called additive log-transformed distributive feature embedding time-dependent regressive convolutional neural learning classifier (ALTDFETRCNLC) has been proposed and designed to forecast COVID-19 more accurately in a shorter time and at an earlier stage.

1.2Objectives

- To develop a deep time-dependent cox regressive Levenberg–Marquardt convolutional neural learning classifier for forecasting COVID-19 with improved precision, recalling, and f-measure.
- To minimize space complexity and error rate in the proposed ALTDFETRCNLC while forecasting COVID-19.
- To compare the performance of the proposed ALTDFETRCNLC with existing models for forecasting COVID-19.
- To assess the feasibility of implementing the proposed ALTDFETRCNLC for real-time COVID-19 forecasting within a minimum time.

The main contribution of the proposed ALTDFETRCNLC method is explained below:

- The accuracy of COVID-19 prediction is enhanced by a novel approach called ALTDFETRCNLC, which has improved various processes such as data preprocessing, feature selection, and classification. The data is normalized using additive log-ratio transformation and converted into a binary representation using one hot encoding. Relevant features were identified using the Tversky similarity-indexed stochastic distributive feature embedding based on similarity measures. This resulted in accurate disease prediction while minimizing space complexity.
- To improve accuracy and reduce error rates, a deep time-dependent Cox regressive Levenberg–Marquardt convolutional neural learning classifier was employed to analyze the selected features and test disease features. The correlation between the features was measured using Cramér's phi correlation function. Based on the correlation results, COVID-19 was accurately predicted. The Huber loss was then measured for each classified result, and the Levenberg–Marquardt algorithm was used to minimize the loss of classification outcomes. This helped to further minimize the error rate of the disease prediction.
- Finally, simulation using deep prediction methods with various performance metrics was used to evaluate the ALTDFETRCNLC approach.

The rest of the paper is organized into five sections. Following the introduction, section 2 provides a comprehensive review of related works in the field. In section 3, the different processes involved in the ALTDFETRCNLC approach are discussed in detail. Section 4 presents information on the simulation setup and dataset used in the study. The performances of the proposed approach are then explained in section 5, which includes an analysis of various performance metrics. Finally, section 6 presents the conclusion of the article, summarizing the key findings and their implications for COVID-19 prediction.

2.Literature review

The recurrent and CNN schemes were developed in [11] for forecasting COVID-19 confirmed cases without minimizing the error rate performance. Many applications of deep learning schemes were introduced by [12] for prediction of the COVID-19 virus. A federated machine learning model was introduced by [13] for COVID19 prediction, by being a failure in predicting the accurate COVID 19 data's. A Harris hawk's optimization was developed by [14] to differentiate the severity of COVID-19, but it did not incorporate other popular methods to predict COVID19, hence became unsuccessful. The logistic scheme was introduced in [15] for predicting the evolution of the COVID-19 pandemic without enhancing the accuracy. The new method was developed by [16] for COVID-19 by implementing adaptive neuro-fuzzy inference schemes and an improved beetle antennae search (BAS) algorithm, but the error rate in predicting COVID19 was not reduced. Transfer learning was introduced to LSTM networks to predict new COVID diseases [17] without reduction in the infection rates. A novel LSTM deep learning construction was designed in [18] for COVID-19 prediction. The developed architecture however failed to deeply analyze the features for accurate prediction. Internet of things (IoT)-based deep learning method was developed in [19] to predict COVID-19. It was not efficient to handle the huge amount of data. Recurrent neural network models were introduced in [20] for multi-step COVID-19 infection forecasting, without accurately forecasting the spread of COVID-19 diseases.

A deep-LSTM ensemble model was designed in [21] to diagnose COVID-19. The error rate was not minimized in this method. The COVID-19 prediction model with deep learning was designed in [22]. This did not work out as the space complexity was not reduced. Recurrent as well as CNN schemes were introduced in [23] to capture the complex development of COVID-19 occurrences to achieve COVID-19 prediction without improvement in the accuracy. CNN-LSTM hybrid deep learning prediction scheme was developed in [24] to correctly forecast COVID-19. However, the prediction time was not reduced.

An ensemble deep learning approach was designed by [25] for combining bagging ridge (BR) using bi-direction along short-term memory (Bi-LSTM) neural networks employed by base regressors with the Bi-LSTM approach. In this method, the error rate was not reduced. Machine learning methods were introduced in [26] for examining automatic COVID-19 identification but the prediction accuracy was not enhanced. A deep learning framework with a COVID-19 adjustment was developed in [27] for electricity demand forecasting. The designed framework failed to accurately detect the COVID-19 prediction performance. Deep learning algorithm was employed in [28] to recognize the COVID-19 disease here in this model the prediction time was higher.

A novel loss function based on cross-entropy was employed in [29] to improve the CNN algorithm's convergence and the main objective is to enhance the model so that it does not reveal 'Covid' as 'non-Covid'. This models several 'false negatives can put lives at risk. Augmentation techniques were determined in [29] to incremental levels and apply them to the largest open-access benchmark dataset, COVIDx CT-2A. Here in this model challenges remain, including low data diversity in existing datasets, and unsatisfied detection resulting from insufficient accuracy.

A deep sequential prediction model (DSPM) and machine learning-based non-parametric regression model (NRM) was employed [30] to predict the spread of COVID-19. The designed model successfully predicted the spread of COVID-19 with minimum error rates. Novel deep transfer learning techniques named "COV-DLS" were introduced in [31] to improve accuracy. It failed to perform preprocessing. Deep learning method was developed in [32] for precise prediction. Deep learning models were analyzed in [33] to discover normal, influenza, and COVID-19 cases but the relevant feature was not identified.

A hybridization of graph convolutional network (GCN) and gated recurrent units (GRU) models was proposed in [34] for the mRNA degradation field which forecasts the stability/reactivity and degradation risk of mRNA sequences. However, the validation loss was not minimized for a certain number of epochs by designed models. Decision-making-based federated learning network (DMFL_Net) was designed in [35] with minimum time with lower accuracy. Machine learning and deep learning using the Covid-19 pandemic method were introduced in [36] to be crucial in better understanding and dealing with the COVID-19 situation. The accuracy was improved by detecting the COVID-19 disease. The COVID-19 patient classification model was performed in [37] to a combination of patient demographic and comorbidity information.

Deep learning techniques were discussed in [38] to predict Covid-19. In [39], a deep CNN architecture was proposed for the diagnosis of COVID-19 based on chest X-ray image classification. The COVID-19-based classification accuracy was improved and then the loss of learning model was reduced. The supervised machine learning model was introduced in [40] to maximize overall accuracy in identifying patient groups found in the COVID-19 patient. The overall accuracy was improved by the designed method and time complexity was minimized.

Deep learning models were introduced in [41] to enhance accuracy without the selection of the relevant features. Yet another Deep learning approach was presented in [42] that combined recurrent neural network (RNN) and long short-term memory (LSTM) networks. It failed to normalize the data. The faster R-CNN and mask R-CNN methods were presented in [43] to train and test the dataset to categorize patients with COVID-19 and pneumonia infections. In [44], an automatic method was introduced for detecting and predicting COVID-19 patients based on their clinical data. COVID-19 Corona Virus India Dataset; https://www.kaggle.com/datasets/imdevskp/covid19-corona-virus-india-dataset[45].

Various methods are employed for the analysis of abnormal events to provide an accurate prediction. From the above studies, it is imperative that limitations are presented alongside every work like offering minimum accuracy, having only few days of data to consider, focus on short-time perspective analysis, and by taking into consideration only few algorithms for training the data. These features are not sufficient to predict the virus. In order to overcome the said limitations, this research is carried out. Three algorithms with input, hidden, and output layers, neuron, and training data are utilized to increase the accuracy level of the disease prediction. This provides unique novelty in this research compared to the previous ones. The core aim of the proposed work is to progress to higher accuracy in preprocessing, feature selection, and classification to enhance efficiency.

3.Methods

COVID-19 prolongs the negative effect in the medical infrastructure and economic growth of the country. An unpredictable leads to increase the trend of infections. Hence, a novel technique is required for reliable data forecasting. A Deep learning technique called ALTDFETRCNLC is introduced to accurately predict COVID-19 at an earlier stage with minimum time and space complexity. The flow process of the proposed ALTDFETRCNLC is shown in *Figure 1*.

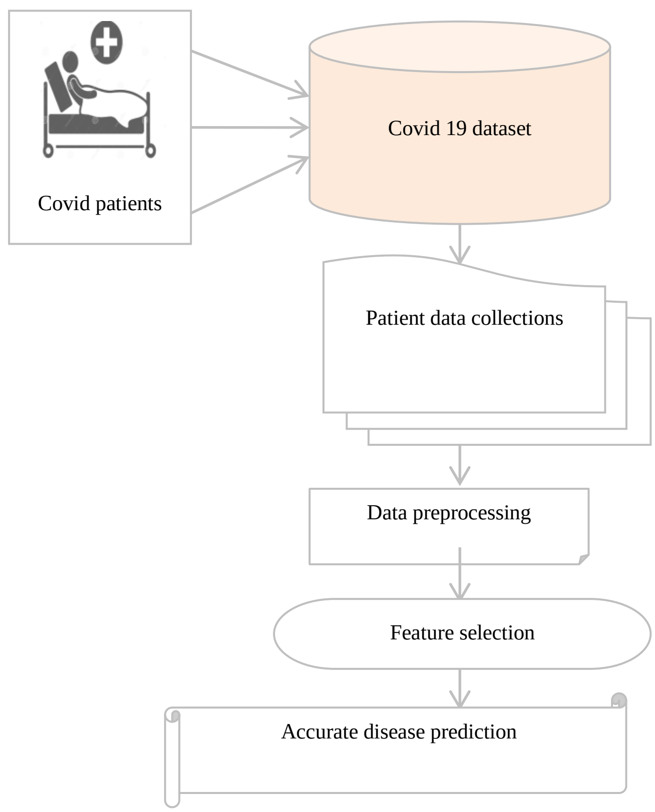

Figure 1 Flow process of the proposed ALTDFETRCNLC

Figure 1 describes a fundamental process of the proposed ALTDFETRCNLC for forecasting COVID-19. The patients data is stored in the dataset which includes the patient data denoted by '$D_i = d_1, d_2, .., d_n$' and the number of features '$f_j =$

$f_1, f_2, \ldots, f_m$'. Whereas, 'n' denotes the number of patient data and 'm' denotes the features. Subsequently, data preprocessing is performed using an additive log ratio transformed by one hot encoder (OHE) to regularize the input data. The feature selection process is thought to be determined for selecting the relevant features by using the Tversky similarity indexed stochastic distributive feature embedding technique. In selected features, classification is attained by using time-dependent Cox regressive Levenberg–Marquardt convolutional neural learning classifier to accurate COVID-19 prediction with better accuracy. The various procedures are developed using deep neural networks. These processes are explained as given below.

3.1Additive log ratio transformed one hot encoding-based data preprocessing

ALTDFETRCNLC first collects patient information from the COVID-19 coronavirus India dataset and performs the data preprocessing to obtain an understandable format which is taken from Kaggle. A dataset is used for accurately calculating diseased patients. Initially, data preprocessing is a method that involves converting raw patient data into a structured format. The obtained raw data is frequently insufficient, inconsistent, and lacks to provide accurate results. Therefore, data preprocessing is performed in the proposed ALTDFETRCNLC for resolving such problems. During the data preprocessing, the normalization process is performed to normalize the different scales of categorical attribute values mapped to an integer value. When there are multiple attributes, but have values on different scales, this may minimize the performance of classification operations. Therefore, normalization is done to obtain attribute values on the same scale. In addition, data preprocessing is utilized to remove the noise for minimizing the space of memory. The flow process of additive log-ratio transformed one hot encoding is revealed below *Figure 2.*

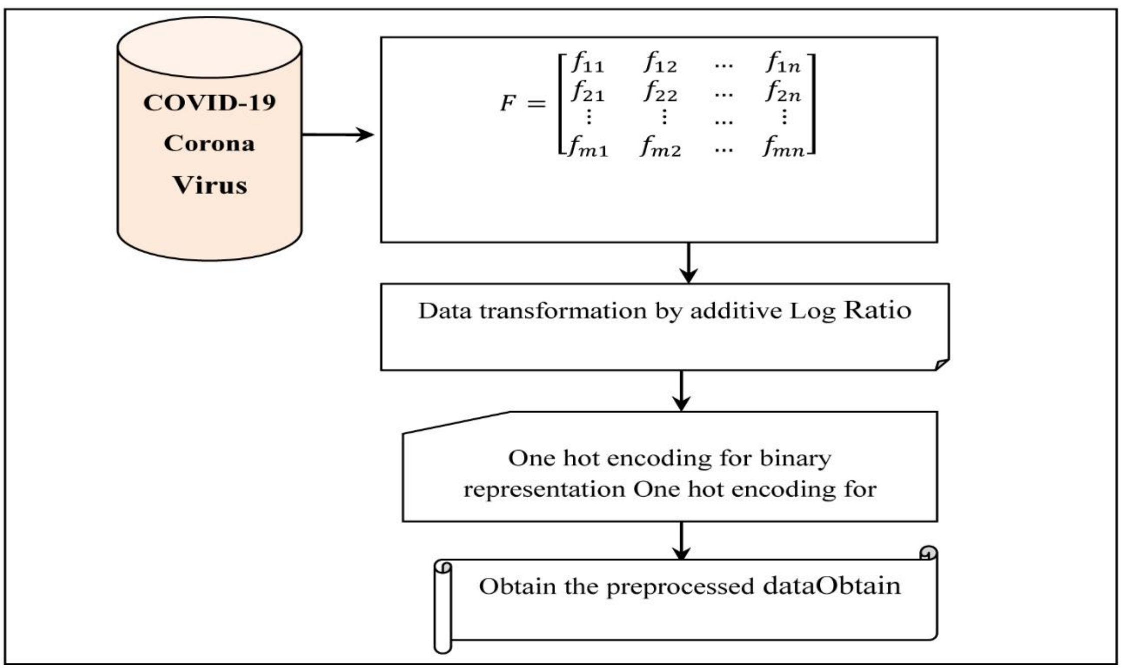

Figure 2 Flow process of Additive Log Ratio transformed One Hot Encoding

Figure 2, additive log-ratio transformed one hot encoding-based data normalization is presented. Let us consider a COVID-19 coronavirus India dataset that consists of features '$F = f_1 f_2, \ldots, f_m$'. First, the data normalization is performed using additive log-ratio transformation and the one-hot encoding technique. Additive log-ratio transformation is a method of rescaling the attributes into the ranges 0 and 1.

Let us consider the feature matrix with 'n' rows and 'm' columns in the dataset. The additive log ratio with standard deviation is applied for normalizing the data as given below in Equations 1 and 2.

$$LR_a = \left(\frac{\log\left| f_{vi-m_f} \right|}{D_s} \right) \quad (1)$$

Where,

$$D_s = \sqrt{\frac{\left(f_{vi-m_f}\right)^2}{n}} \quad (2)$$

Where, LR_a denotes an output of Additive Log Ratio results, f_{vi} denotes a feature value, m_f denotes a mean of the particular feature value, D_s indicates a standard deviation, n denotes the number of samples. The output of LR_a provides the ranges 0 and 1. Following, data decoding is achieved for transferring numerical data within binary coding. The proposed technique uses One Hot Encoding which helps to convert numerical categorical variables into binary vectors. Before implementing the normalized data into an algorithm, make sure that all the numerical attribute values must be encoded.

Let us consider the normalization of features in the array of matrix '*NA*' in Equation 3.

$$NF = \begin{bmatrix} Nf_{11} & Nf_{12} & \dots & Nf_{1n} \\ Nf_{21} & Nf_{22} & \dots & Nf_{2n} \\ \vdots & \vdots & \dots & \vdots \\ Nf_{m1} & Nf_{m2} & \dots & Nf_{mn} \end{bmatrix} \quad (3)$$

Where NFdenotes a normalized feature matrix, $Nf_{11}, Nf_{12} \dots, Nf_{1n}$ are the normalized numerical value of the features. Then, the input numerical feature value is fit into the encoder for obtaining the binary value using Equation 4.

$$OHE \xleftarrow{NBR} \begin{bmatrix} Nf_{11} & Nf_{12} & \dots & Nf_{1n} \\ Nf_{21} & Nf_{22} & \dots & Nf_{2n} \\ \vdots & \vdots & \dots & \vdots \\ Nf_{m1} & Nf_{m2} & \dots & Nf_{mn} \end{bmatrix} \quad (4)$$

Where NBR denotes a numeric to binary representation with Equation 5.

$$OHE \xrightarrow{returns} \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1n} \\ b_{21} & b_{22} & \dots & b_{2n} \\ \vdots & \vdots & \dots & \vdots \\ b_{m1} & b_{m2} & \dots & b_{mn} \end{bmatrix} \quad (5)$$

Whereas, OHEreturns the binary representation 'b' as an output. It aids to reduce the time and space of algorithms. The algorithmic process of additive log-ratio transformed one hot encoding-based data preprocessing is shown in Algorithm 1..

// Algorithm 1: Additive log ratio transformed one hot encoding based data preprocessing

Input**: Dataset D, features $F = f_1 f_2, \dots, f_m$'**

Output**: Obtain the pre-processed data**

Begin

1**: Collect the number of featuresfrom the dataset** 'D'

2: For **each feature in the dataset 'f_i'**

3. **Construct the feature matrix** '*A*'

4. **Measure the log ratio with mean and standard deviation** 'LR_a'

5. **Normalize the data in the ranges from 0 to 1**

6. End for

7. Apply OHE **to a normalized data**

8. Return (binary representation of data '0' or '1')

9. Obtain the preprocessed dataset

End

Algorithm 1 explains data preprocessing. Initially, the number of features and the raw data is gathered. Additive log-ratio transformation to obtain normalized data is applies. OHE is used to transform the numerical data into a binary representation. Thus, data preprocessing is achieved for obtaining a binary representation of data to minimize space complexity.

3.2Tversky similarity indexed distributive feature embedding technique-based Feature selection

Behind data preprocessing, the proposed ALTDFETRCNLC performs feature selection using the Tversky similarity-indexed distributive feature embedding technique. Feature selection is the process of selecting the most relevant features by removing redundant features, and irrelevant or noisy features when developing a predictive model. It is desirable to reduce the computational cost of modeling and, in some cases, to enhance the performance of the model. Generally, the dataset consists of features that slow down the learning process and cause the classifier to provide an accurate classification task. It also deteriorates the performance of accuracy and the teaching rapidity is considerably reduced for implementing many features. Hence, the essential value for choosing relevant as well as essential features within preprocessing phases is to minimize the complexity and increase the accuracy of the classification. Therefore, the proposed technique performs the relevant feature selection. Tversky similarities indexed distributive feature embedding technique is a dimensionality reduction method that helps for visualizing high-dimensional data in low-dimensional space of similar features. They are explained using higher similarity and different features are explained using lesser similarity. Let us consider the number of features distributed throughout a high-dimensional space $F = f_1 f_2, .., f_m$. The similarity between the features is measured using the Tversky indexes given below in Equation 6,

$$\delta = \frac{[f_i \cap f_j]}{K(f_i \cap f_j) + L(f_i - f_j)} \quad (6)$$

Where δ indicates a similarity coefficient, f_i and f_j denotes two features, $f_i \cap f_j$ indicates a mutual dependence between the two features, $f_i - f_j$ indicates a variance between the two features. From (1), K and L designate parameters of the Tversky index ($K, L \geq 0$). Coefficient (δ) offers output results among [0, 1]. Depending on coefficient results, similarity features were correctly identified in Equation 7.

$$\delta = \begin{cases} \beta = 1, \ relevant features \\ \beta = 0, \ irrelevant features \end{cases} \quad (7)$$

Relevant features are chosen for accurate classification, in order to achieve that various features are eliminated. Selected features are shared to the next process resulting in minimizing the time and space consumption of disease prediction.

// Algorithm 2: Tversky similarity indexed stochastic distributive feature embedding technique

Input: Dataset, preprocessed features $F = f_1 f_2, .., f_m$
Output: Select relevant features

Begin
1. Collect the number of features $\{F = f_1 f_2, .., f_m\}$
2. **For each** feature 'f_i' and f_j
3. Measure the similarity '$\beta \ (k_i, k_j)$'
4. **if** ($\delta = 1$)**then**
5. The feature is said to be relevant
6. **Select** relevant features
7. **else**
8. The feature is said to be irrelevant
9. **Remove ir**relevant features
10. **end if**
11. **end for**
End

Algorithm 2 describes the relevant feature selection based on similarity measures. The features are taken over the COVID-19 coronavirus India dataset to find the relevant features. If the similarity coefficient returns '1', then the feature is relevant. Otherwise, the feature is irrelevant. Relevant features are chosen for disease prediction by removing additional features. This process of the proposed technique minimizes time as well as space consumption.

3.3Time-Dependent Cox regressive Levenberg–Marquardt Convolutional neural learning classifier based Covid 19 prediction

The classification process is carried out in the proposed ALTDFETRCNLC using a time-dependent Cox regressive Levenberg–Marquardt convolutional neural learning classifier for disease prediction. It is a Deep learning classifier that helps to deeply analyze the features in multiple layers and provides accurate classification results. For feature analysis, the Cox regression is applied to a convolutional classifier which provides the output results with minimum time. The structure of the time-dependent Cox regressive Levenberg–Marquardt convolutional neural learning classifier is illustrated in *Figure 3*.

Figure 3 demonstrates the concordance correlated Cox regressive convolutional neural learning classifier that holds three types of layers namely: an input layer, an output layer, and more than one hidden layer. The input layer of a deep learning classifier fetches the input (i.e. selected features) and is given into the system for further processing by subsequent layers of artificial neurons. The input layer is positioned at the very beginning of the deep learning network. The output of one layer is fully connected to other successive layers in the feed-forward manner with the equivalent set of weights to form the entire network. The input layer consists of selected features such as patients and these features are learned with hidden layers. In the hidden layer, time-dependent cox regression is utilized to measure Cramér's phi correlation function which helps in precise classification. With every timestep, the Huber loss is estimated for predicted, actual results and the Levenberg–Marquardt algorithm is employed to reduce error. Lastly, the prediction results are achieved at the output layer. These processes are briefly explained as given below.

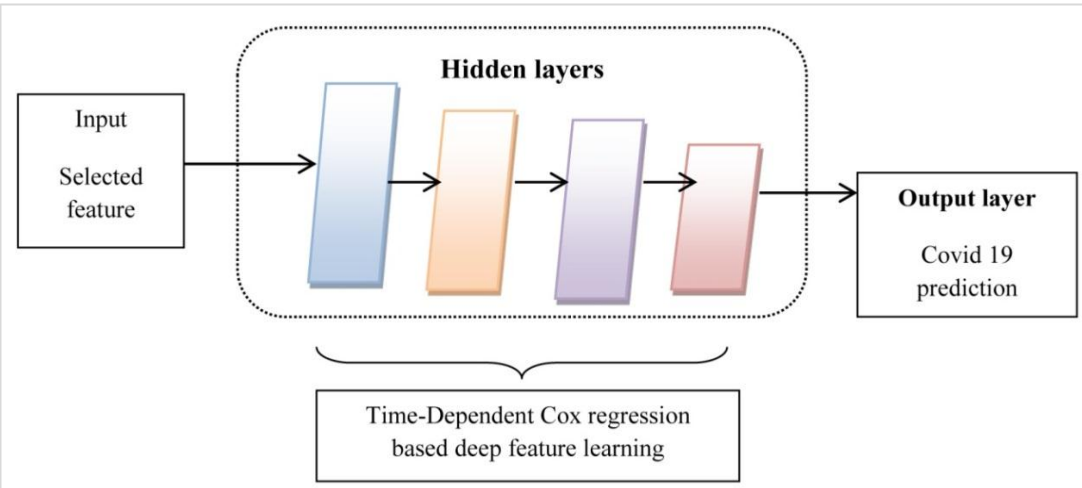

Figure 3 Structure of time-dependent cox regressive Levenberg–Marquardt convolutional neural learning classifier

The input layer receives selected features $f_1, f_2, f_3, \ldots f_k$ and with the patient data

The input layer '$z(t)$' is given below Equation 8,

$$z(t) = D + \left[\sum_{i=1}^{k} f_i(t) * q_{input}\right] \quad (8)$$

Where D indicates bias, whose value is '1', $f_i(t)$denotes input features, q_{input}' represents weight.

Next, input is transformed within hidden layers and the feature learning process is carried out using concordance correlated Cox regressive. In the hidden layer, the Convolutional neural learning classifier includes a max-pooling layer to reduce the dimensions of data by combining the input at one layer and transforming it into the next layer.

The time-dependent cox regression model is a machine learning approach that helps measuring the relationships between the time-to-event outcome Y (i.e. output) and a set of explanatory variables (i.e. features$f_1, f_2, f_3, \ldots f_k$). The output of the disease prediction depends on the time. By applying the time-dependent coxregression Equation 9.

$$Y\ (t) = g_o(t).\exp\ (\rho_c.R) \quad (9)$$

Where $Y\ (t)$ denotes a hazards function at times, $g_o(t)$ denotes a covariate vector,ρ_c denotes a regression coefficient, R denotes a Cramér's phi correlation function.

The Cramér's phi correlation is used as h for measuring the connection between two variables in Equation 10.

$$R = \left[\frac{\sum_{i=1}^{k}\sum_{j=1}^{m}|f_k - f_j|^2}{(n-1)+(m-1)}\right] \quad (10)$$

From (10), R denotes a Cramér's phi test result, features 'f_k'denotes a selected feature, 'f_j'denotes a testing disease patterns, n, m are sample sizes. Cramér's phi test returns a value from 0 (no association between the features) to 1 (absolute association between the features) with Equation 11.

$$R = \begin{cases} 1\ ; & f_k isassociatedwith f_j \\ 0\ ; & noassociationbetween f_k and\ f_j \end{cases} \quad (11)$$

The output of the Cramér's phi test R returns+1' indicates that the features are similar such as patient data being properly classified by diseased or normal, and it is selected whereas the value of '0' indicates that the features are not similar. Thus, patient data classification is achieved by reducing time complexity. Finally, the hidden layer is obtained by Equation 12.

$$P\ (t) = \left[\sum_{i=1}^{k} f_i(t) \times q_{input}\right] + [q_{\text{hidden}} \times P_{h-1}] \quad (12)$$

From (12), '$P(t)$' indicates the hidden layer output, 'R_i' denotes the weight, R_{h-1} denotes the preceding hidden layer, '×' denotes a convolutional operator. Finally, the output is transferred to the hidden layer and a modified Huber loss function is applied for minimizing the error rate by Equation 13.

$$L_h = \frac{1}{2}[A_o - P_o]^2 \quad (13)$$

Where 'L_h indicates a Loss, A_o indicates actual results, P_o denotes predicted classification results.

The proposed DL classifier uses the Levenberg–Marquardt algorithm to find a local minimum (i.e., minimum Loss) as given below in Equation 14.

$$F = arg \min L_h \quad (14)$$

Where, the output of the Levenberg–Marquardt algorithm, $argmin$ denotes the argument of the minimum function, L_h indicates a loss.

If the minimal error is attained, then the results are shown in an output layer as follows in Equation 15.

$$Z = \sum_{i=1}^{n} P(t) \times q_{\text{hidden out}} \quad (15)$$

From Equation (15), '$P(t)$' represents the output of classification result, $P(t)$ denotes a hidden layer output, '$q_{\text{hidden out}}$' indicates weight among hidden as well as output layers. Thus, accurate classification is obtained at the output layer. The accurate disease is obtained at the output layer. The algorithmic process of time-dependent cox regressive Levenberg–Marquardt convolutional neural learning classifier-based COVID-19 predictions is shown in Algorithm 3.

Algorithm 3 describes the step-by-step process for COVID-19 prediction using the time-dependent Cox regressive Levenberg–Marquardt convolutional neural learning classifier. First, the numbers of relevant significant features are given as input. With selected training feature values, classification is achieved by analyzing and testing disease patterns and training data using time-dependent Cox regression. Based on time-dependent Cox regression, accurate classification is performed with the help of Cramér's phi correlation. For each timestep, the Huber loss is measured for predicted and actual outcomes. If the minimal error is found using Levenberg–Marquardt algorithm, the prediction outcomes are attained. It aids in increasing the disease prediction accuracy.

Algorithm 3: Time-Dependent Cox regressive Levenberg–Marquardt Convolutional neural learning classifier based Covid 19 prediction

Input: selected features '$f_1, f_2, f_3, \dots . f_k$', Patient data samples $D = D_1, D_2, \dots, D_n$
Output: Increase prediction accuracy

Begin
Step 1. **Collect the relevant** features $f_1, f_2, f_3, \dots . f_k$ in input layer
Step 2: **For** each testing feature 'f_i' --**hidden layer**
Step 3: **For** each training feature 'f_k'
Step 4: Apply Time-Dependent Cox regression
Step 5: **If** $(R = 1)$**then**
Step 6: **patient D**ata sample is classified as a diseased
Step 7: **else**
Step 8: **patient D**ata sample is classified as a normal
Step 9: **end if**
Step 10: **end for**
Step 11: **end for**
Step 12: **For each time step '*t*'**
Step 13: Measure Huber loss 'L_h'
Step 14: Apply –the Levenberg Marquardt algorithm to find the minimum loss
Step 14:*if*(arg min L_h) **then**
Step 15: Obtain accurate classification results at the output layer
Step 16: **else**
Step 17: Repeat step 2
Step 18: **end if**
Step 19: **End for**
Step 20: **End for**
End

4.Results

In an experimental setting, proposed ALTDFETRCNLC and existing methods are implemented in Python high-level generation purpose programming language. To analyze the performance of ALTDFETRCNLC, a COVID-19 coronavirus India dataset is applied for the implementation process. The dataset was considered using COVID-19 Corona Virus India Dataset; https://www.kaggle.com/datasets/imdevskp/covid19-corona-virus-india-dataset [30]. The implementation is conducted with the hardware specification of the Windows 10 Operating system, core i3-4130 3.40GHZ Processor, 4GB RAM, 1TB (1000 GB) Hard disk, ASUSTeK P5G41C-M Motherboard, and Internet Protocol. For data analysis, sampling is used to practice analyzing a subset of two datasets to uncover meaningful information in the larger dataset. It is divided into two sets such as training and testing sets. 70% of patient data is used for training and 30% of data is employed for testing. The class distribution is defined as 0 or 1. '+1' is denoted as the patient data sample is correctly classified as diseased. '0' is represented as a patient data sample correctly classified as normal. In the dataset, the number of patients is taken from the range of 10000 to 100000 for conducting the experiments. It consists of eight.csv files as shown in *Table 1*. From *Tables 2, 3, 4, 5,* and *6,* State level daily.csv, State level latest.csv, national level daily. CSV, District level latest.csv, and Patients data.csv files are considered for conducting the experiments. From the evaluation, the patients affected by COVID-19 e corresponding district and states locations are identified.

Table 1 Details of COVID-19 coronavirus India dataset

S. No.	Content	Description
1	Complete.csv	Cumulative count of day to day state wise A number of cases.
2	District level latest.csv	Latest district level counts
3	National level daily.csv	Day by day national-level numbers
4	Patientdat.csv	Patient-wise data
5	State level daily.csv	State level daily
6	State level latest.csv	Latest state level
7	Test daywise.csv	Day-wise test statistics
8	Test statewise.csv	State-wise test statistics

Table 2 State level daily.csv

S. No.	Features
1	S No
2	Date
3	State
4	confirmed cases
5	deceased cases
6	Recovered
7	State name

Table 3 State level latest.csv

S. No.	Features
1	State
2	Confirmed
3	Recovered
4	Deaths
5	Active
6	Last_updated_Time
7	Migrated_Other
8	State_Code
9	Delta_ Confirmed
10	Delta_ Recovered
11	Delta_ Deaths
12.	State-Node

Table 4 National level daily.csv

S. No.	Features
1	Date
2	Daily confirmed
3	Total confirmed
4	Daily Record
5	Total record
6	Daily decreased
7	Total decreased

Table 5 District level latest.csv

S. No.	Features
1	State name
2	State code
3	Name of the district
4	Number of confirmed cases
5	Number of active cases
6	Number of deceased cases
7	Number of recovered cases
8	Change in confirmed cases
9	Change in active cases
10	Change in deceased cases
11	Change in recovered cases

Table 6 Patient data.csv

S. No.	Features
1	Patient number
2	Patient ID
3	State-wise patient ID

S. No.	Features
4	The date on which the case is announced
5	Age
6	Gender
7	City in which case is detected
8	District in which case is detected
9	State in which case is detected
10	State code
11	Current status (nationality of the patient)
12	Contracted from which patient
13	Nationality
14	Type of transmission
15	Status Change Date
16	Source 1
17	Source 2
18	Source 3

The confusion matrix of the proposed ALTDFETRCNLC is demonstrated in *Table 7.*

A confusion matrix is a fashionable metric utilized for handling the classification issues to binary classification in above *Table 7*. It is measured based on the predicted and actual values. The row values of confusion metrics symbolizes the corresponding true label as well as column values indicates the corresponding predicted labels. The value that appears in each cell shows the prediction labels. The number of patients (i.e, 10000) is taken in the dataset. T_p is true positive, F_p is false positive, F_n is false negative, and T_n is a true negative. *Table 8* given below lists the hyper-parameters and their description employed in our proposed method.

Table 7 Confusion matrix

Number of patients =10000	**Classified: Yes**	**Classified: No**	**Total samples**
Actual: No	T_n=350	F_p=1300	135
Actual: Yes	F_n=110	T_p= 8240	365
Total samples	115	385	10000

Table 8 Hyper-parameters and description

S. No.	Hyperparameters	Description
1	Number of hidden layers used	More than one hidden layer is used (the hidden layer from the Convolutional neural learning classifier includes max pooling)
2	Activation function used in hidden layers	Time-Dependent Cox regression, Cramér's phi correlation, and Huber loss functionis used in the hidden layer
3	Activation function used in the output layer	Levenberg–Marquardt algorithm
4	Learning rate	The value of the learning rate used in our work is 0.01
6	Batch size	Batch size in our work refers to the samples from the training dataset. In our work, the batch size is 5000 as samples are considered for simulation.
7	Number of epochs	The number of epochs in our work is 10

The results of the proposed ALTDFETRCNLC and existing methods namely deep-LSTM ensemble model [1], SIRVD-DL [2], Deep learning method [3], and MTGP regression [4] are compared using the different metrics such as

- Accuracy
- Error rate
- Time-space complexity
- Precision
- Recall and
- F measure

Prediction accuracy: It is referred by the proportion of the number of patients correctly predicted as normal or COVID-19 confirmed cases to the total number of patients. It is formulated as follows Equation 16.

$$PAcc = \sum_{i=1}^{n} \frac{P_{AP}}{P_i} \times 100 \quad (16)$$

Where$PAcc$' denotes a prediction accuracy,P_{AP} is the number of patients accurately predicted, P_i denotes the number of patients involved in the simulation process. It is computed by percentage (%).

Error rate: It is calculated by the ratio of the number of patients wrongly predicted as normal or COVID-19 confirmed cases to an entire number of patients. The error rate is expressed as shown in the below Equation 17.

$$ERate = \sum_{i=1}^{n} \frac{PWP_{AC}}{P_i} \times 100 \quad (17)$$

Where, $ERate$ denotes an error rate, PWP_{AC} indicates patients wrongly predicted 'P_i' is the number of patients involved in the simulation process. It is calculated by percentage (%).

Prediction time: It is calculated by the number of times consumed with an algorithm in the prediction of whether the patient is affected by COVID or not. The overall prediction time is formulated in Equation 18.

$$PTime = \sum_{i=1}^{n} P_i \times Time[PSP] \qquad (18)$$

Where, $PTime$ denotes a prediction time,P_i represents the patients involved in simulation '$Time[PSP]$' denotes the time to classify a single patient affected with COVID or not. It is computed in milliseconds (ms).

Space complexity: It is referred to the number of memory space taken to predict whether the patient is infected by COVID or not as in Equation 19.

$$Space_{com} = \sum_{i=1}^{n} P_i \times Mem[PSP] \qquad (19)$$

Where, $Space_{com}$denotes a space consumption, P_i represents the patients involved in simulation '$Mem[PSP]$' denotes a memory space consumed to classify a single patient being infected with COVID or not. It is computed in Megabytes (MB).

Precision: It is measured based on a number of true positives as well as false positives. Therefore, the precision is calculated as follows in Equation 20.

$$Pr = \left(\frac{T_p}{T_p + F_p}\right) \times 100 \qquad (20)$$

Where Prsymbolizes Precision and it is measured in percentage (%).

Recall: It is measured to determine the number of true positives as well as false negatives during the prediction. It is formulated as in Equation 21.

$$R_c = \left(\frac{T_p}{T_p + F_n}\right) \times 100 \qquad (21)$$

Where 'C_R 'specifies a recall and it is measured in percentage (%).

F measure: It is measured as the mean value of both precisions as well as recall. It is computed using the following mathematical Equation 22.

$$\text{F measure} = \left[2 \times \frac{P_r * R_c}{P_r + R_c}\right] \times 100 \qquad (22)$$

Where, F measure computed based on precision P_r and recall 'R_c'. It is measured in terms of percentage (%).

Figure 4 provides the experimental analysis of prediction accuracy versus the number of patients from 10000 to 100000. The prediction accuracy is calculated by ALTDFETRCNLC as well as the deep-LSTM ensemble model [1] and SIRVD-DL [2], Deep learning method [3], and MTGP regression [4]. Among four methods, the ALTDFETRCNLC provides improved performance when compared to existing [1–4] respectively. For example, the number of input patients is considered to be 10000. The accuracy of ALTDFETRCNLC is observed as 98.6% and the accuracy of the deep-LSTM ensemble model [1], SIRVD-DL [2], and Deep learning method [3] was found to be 96.25%, 94.2%, 97.4%, and 94.23% respectively. The number of input patients is considered to be 20000. The accuracy of ALTDFETRCNLC is observed as 98.1% and the accuracy of deep-LSTM ensemble model [1], SIRVD-DL [2], and Deep learning method [3] were found to be 95.5%, 93.75%, 96.6% and 94.8% respectively. Consequently, different performance outcomes are attained. The observed outcomes of the ALTDFETRCNLC are compared with existing techniques. Finally, ten evaluation outcomes provides the output of proposed ALTDFETRCNLC that has enhanced the disease prediction accuracy as 4%,6%, 3% and 5%compared with [1–4] respectively.

As revealed in *Figure 4,* the numbers of patients are presented in the horizontal axis, and the performance of accuracy of various techniques is observed on the 'vertical axis. The performance outcomes denote accuracy of ALTDFETRCNLC is improved than the other three conventional deep learning methods. This is because of the time-dependent Cox regressive Levenberg–Marquardt convolutional neural learning classifier. The deep learning classifier receives the number of relevant significant features given to the input layer. The time-dependent Cox regression is applied to deep learning for detecting features with disease patterns with the help of Cramér's phi correlation. Based on correlation measures, accurate classification results are observed at the output layer.

Table 9 given below lists the error rate for four different methods, ALTDFETRCNLC as well as deep-LSTM ensemble model [1] and SIRVD-DL [2], Deep learning method [3], and MTGP regression [4].

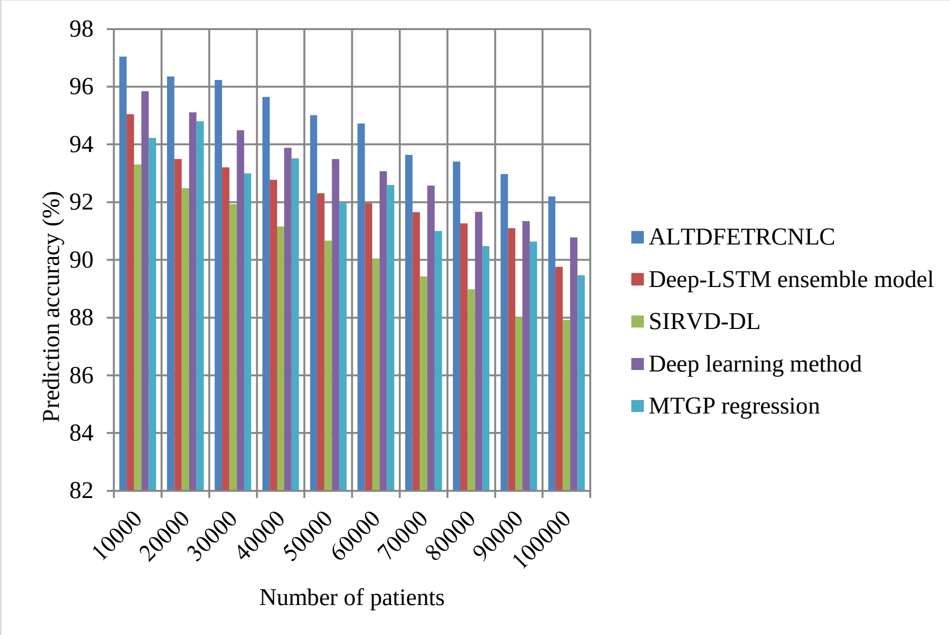

Figure 4 Graphical Illustration of prediction accuracy

Table 9 Error rate

Number of patients	**Error rate (%)**				
	ALTDFETRCNLC	**Deep-LSTM ensemble model**	**SIRVD-DL**	**Deep learning method**	**MTGP regression**
10000	1.4	3.75	5.8	2.6	2.4
20000	1.9	4.5	6.25	3.4	3
30000	2.33	6	7.33	4.7	4.2
40000	2.62	6.25	8.25	5.2	4.9
50000	2.8	6.5	8.76	5.5	5.3
60000	3	6.9	9.25	5.7	5.5
70000	3.07	6.99	9.4	6	5.9
80000	3.22	7.375	9.82	6.2	6
90000	3.44	7.55	10.44	6.8	6.4
100000	3.69	7.78	10.63	7	6.8

Table 9 presents the experimental assessment of error rate with patients such as ALTDFETRCNLC and existing methods namely the deep-LSTM ensemble model [1], SIRVD-DL [2], and Deep learning method [3]. The performance results designate that the overall performance of the error rate using the proposed ALTDFETRCNLC was found to be lesser when compared to other Deep learning methods. Let us consider 10000 numbers of patients in the first iteration and compute the error rate using the ALTDFETRCNLC as 1.4% whereas the error rate of [1–3] are 3.75%, 5.8%, 2.6% and 2.4% respectively. Likewise, the overall performance of ten results notices that the average of ten results is reduced to 57%, 68%, 48%and 44%with [1–4]. The reason behind Proportional Cox regressive Mahout Deep Recurrent Perception Neural Learning Classifier (PCRMDRPNLC) technique is to apply the time-dependent Cox regressive Levenberg–Marquardt convolutional neural learning classifier. For each output result, Huber loss is measured and Levenberg–Marquardt algorithm is applied for finding the minimum loss. As a result, the error rate of data classification is minimized

Figure 5 represents the COVID-19 time with the number of patients. Time is calculated by the number of times consumed by the four methods namely ALTDFETRCNLC and deep-LSTM ensemble model [1], SIRVD-DL [2], and Deep learning method [3], MTGP regression [4]. It is observed among the four methods, ALTDFETRCNLC consumes lesser time consumption for disease prediction. Let us consider 10000 patients for experimentation. The prediction time of the proposed ALTDFETRCNLC is 8200ms. The time consumption of disease prediction using [1−3] was found to be 8800ms, 9200ms, 8400ms and 9800ms respectively. The different performance outcomes are attained for all the techniques. The overall analysis of the result demonstrates that the disease prediction time of ALTDFETRCNLC is reduced by 6%, 9%, 4%, and 3% compared to the existing technique. But comparatively, ALTDFETRCNLC was achieved for minimizing the time consumption. The reason being the relevant feature selection using the Tversky similarity-indexed stochastic distributive feature embedding technique. Then the Tversky similarity index is utilized to find relevant features. Relevant and irrelevant features were identified and selected. Relevant features are chosen for disease prediction as well additional features are eliminated from the dataset. It aids in minimizing the time of disease prediction (*Table 10*).

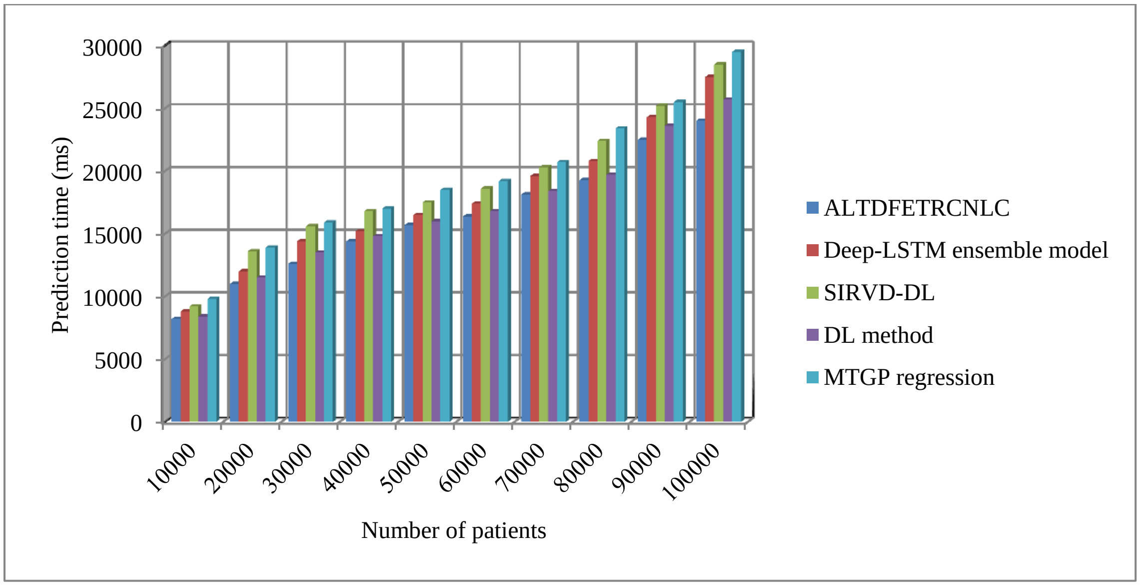

Figure 5 Prediction time

Table 10 Space complexity

Number of patients	Space complexity (MB)				
	ALTDFETRCNLC	**Deep-LSTM ensemble model**	**SIRVD-DL**	**Deep learning method**	**MTGP regression**
10000	23	26	23	25	24
20000	32	35	32	34	33
30000	36	39	36	37	35
40000	44	48	44	46	45
50000	50	55	50	53	52
60000	54	60	54	58	56
70000	57.4	63	57.4	60	58
80000	60	66.4	60	63	62
90000	64.8	68.4	64.8	66.8	65
100000	68	71	68	69	68.4

The performance analysis of space complexity of four different methods ALTDFETRCNLC and deep-LSTM ensemble model [1], SIRVD-DL [2], and Deep learning method [3], MTGP regression [4] are illustrated in *Table 8* for the number of patients 10000, 20000, 30000…100000. Let us consider that the number of patients is 10000 in the experimentation. The space consumption for predicting the disease is $23MB$ using ALTDFETRCNLC whereas the memory consumption of the other four methods are $26MB$, $28MB$, $25MB$ and $24MB$respectively. The observed results substantiate that the proposed ALTDFETRCNLC achieve lesser memory consumption. The overall analysis of the result shows that the space consumption of ALTDFETRCNLC is minimized by 8%, 13%, 5% and 2% compared with existing methods.

Figure 6 exhibits the overall performance of precision using four different methods namely ALTDFETRCNLC and deep-LSTM ensemble model [1], SIRVD-DL [2], and Deep learning method [3]. For each method, ten different results are observed and shown in *Figure 6*. The performance of recall using the ALTDFETRCNLC is higher when compared to conventional methods [1–3] respectively. This is proved through some statistical measures. Let us consider 10000 patients in the first iteration and calculate the precision. The observed performance of recall value using ALTDFETRCNLC and existing deep-LSTM ensemble model [1], SIRVD-DL [2], and Deep learning method [3] and [4]are 97.6%, 95.45%, 93.91%, 96.5% and 92.4% respectively. Likewise, different results are observed for each method with respect to the number of patients. Finally, the obtained performance results are compared to existing methods. The overall analysis results indicate that the precision is considerably improved by 3% using ALTDFETRCNLC when compared to [1], 5% when compared to existing [2], 2% when compared to existing [3], and 5% when compared to existing [4] respectively.

Table 11 reveal recalls the four different methods, ALTDFETRCNLC as well as deep-LSTM ensemble model [1] and SIRVD-DL [2], Deep learning method [3] and MTGP regression [4].

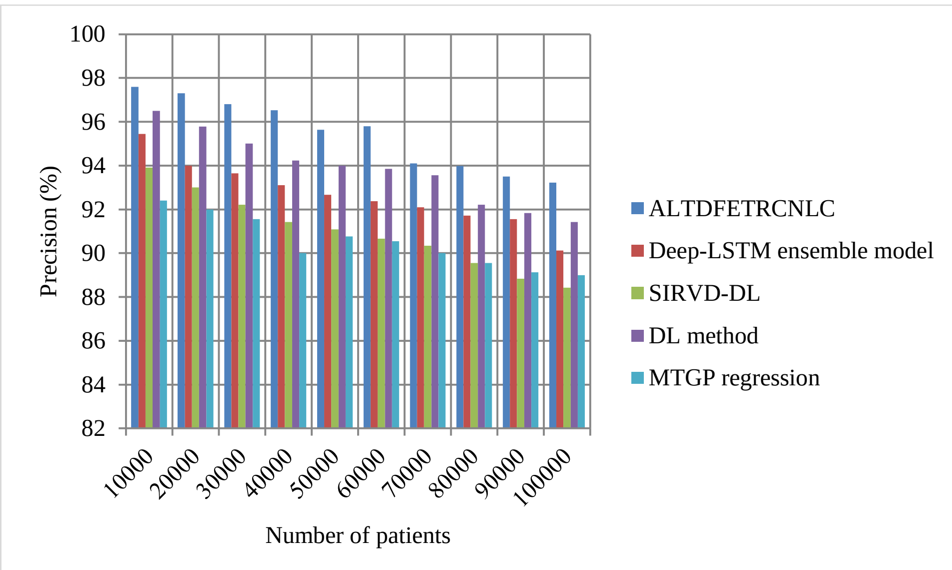

Figure 6 Performance of precision

Table 11 Recall

Number of patients	Recall (%)				
	ALTDFETRCNLC	**Deep-LSTM ensemble model**	**SIRVD-DL**	**Deep learning method**	**MTGP regression**
10000	96.5	94.67	92.72	95.22	94
20000	95.43	93	92	94.45	93.2
30000	95.67	92.79	91.65	94	93.7
40000	94.8	92.45	90.88	93.55	92
50000	94.4	91.97	90.28	93	92.6
60000	93.7	91.58	89.45	92.32	91
70000	93.2	91.22	88.55	91.62	90.37
80000	92.85	90.84	88.43	91.14	90
90000	92.45	90.66	87.21	90.86	89.4
100000	91.2	89.42	87.43	90.15	88.64

Table 11 provides the performance assessment of recall with four methods namely the ALTDFETRCNLC and deep-LSTM ensemble model [1], SIRVD-DL [2], and Deep learning method [3].

The observed results indicate that the overall performance of recall using the proposed ALTDFETRCNLC was found to be improved when compared to other methods. Let us consider 10000 numbers of patients in the first iteration and the observed recall results using the ALTDFETRCNLC is 96.5% whereas the observed recall results of [1–3] are 94.67%, 92.72%, 95.22% and 94% respectively. Likewise, other different performance results are observed for a number of patients. After that, the performance results of the proposed ALTDFETRCNLC are compared to the performance of existing methods. The overall results indicate that the average of ten comparison results indicates that the performance of recall is found to be increased by 2% when compared to [1] and 5% when compared to [2], 2% when compared to [3] and 3% when compared to [4] respectively. This is owing to the implementation of the time-dependent cox regressive Levenberg–Marquardt convolutional neural learning classifier. The Deep learning technique accurately analyzes features with the testing metrics. In addition, the false positive rate is minimized by finding a minimal error. This in turn increases the recall.

Figure 7 depicts the performance results of the F measure versus the number of patients taken from the dataset. The F measure is computed to precision and recall. The observed result specifies that the ALTDFETRCNLC provides better performance when compared to existing methods by improving the precision and recall. Finally, the overall results of the proposed ALTDFETRCNLC are compared to the results of existing methods. The overall analysis of ten comparison results indicates that the overall performance of the F measure is significantly improved by 3%, 5%,2%, and 3% when compared to existing [1–4] respectively.

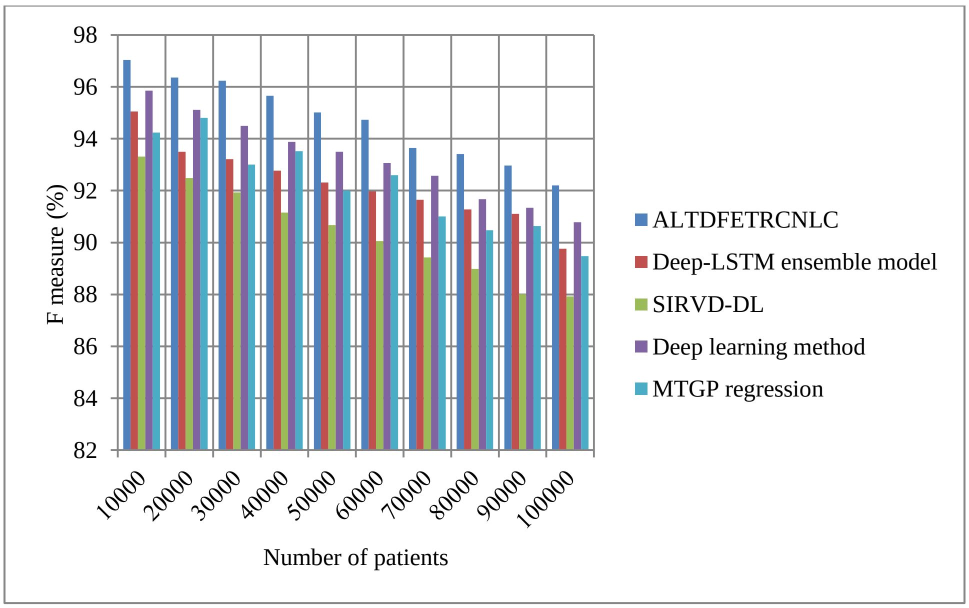

Figure 7 Performance of F measure

Comparison of different feature combination methods

For selecting features for the COVID-19 prediction, the Tversky similarity-indexed stochastic distributive feature embedding technique is employed in ALTDFETRCNLC. Next, the proposed ALTDFETRCNLC, Conventional deep-LSTM ensemble model [1], SIRVD-DL [2], Deep learning method [3], and MTGP regression [4] were compared for feature combination. There may be redundancies among these feature combinations, which can affect performance. Therefore, each of these four feature combinations was dimensionally reduced and the classification results of the validation set were estimated using the dataset, and the results were revealed in *Table 12.*

The performance is compared in diverse parameters for different feature combinations. For example, the accuracy based on different features is compared to age, gender, and glucose level achieved by 97% of accuracy. The best classification was found by comparing experimental results which provided precision values of 95%, recall values of 94%, F measure values of 95%, error rate values of 3%, space complexity values of 48ms, and prediction time values of 16219 MB. The dataset dimensionality, space, and computation time were minimized by selecting features based on Tversky similarity.

Table 12 Performance comparison of different feature combinations

Methods	Parameter Name						
	Accuracy (%)	**Precision (%)**	**Recall (%)**	**F measure (%)**	**Error rate (%)**	**Time complexity (ms)**	**Space complexity (MB)**
ALTDFETRCNLC	97	95	94	95	3	16219	48
Deep-LSTM ensemble model	94	93	92	92	6	17650	53
SIRVD-DL	91	91	90	90	9	18770	56
DL method	95	94	93	93	5	16840	51
MTGP regression	90	92	91	92	6	19340	49

5.Discussion

This study compares the proposed technique with the deep-LSTM ensemble model [1], SIRVD-DL [2], Deep learning method [3], and MTGP regression [4] using the COVID-19 India dataset. The objective is to accurately predict COVID-19 by classifying patient data. Data preprocessing involves additive log-ratio transformation and one-hot encoding to normalize the input data. The feature selection process utilizes the Tversky similarity indexed stochastic distributive feature embedding technique to select relevant features. Classification is achieved using a time-dependent Cox regressive Levenberg–Marquardt convolutional neural learning classifier, resulting in accurate COVID-19 prediction with higher accuracy.

An experimental evaluation is conducted to assess the performance of the proposed technique. The results demonstrate significant improvements in prediction accuracy, precision, recall, F-measure, error rate, space complexity, and prediction time. The key findings are as follows:

The proposed ALTDFETRCNLC technique achieves higher prediction accuracy by 5%, precision by 4%, recall by 3%, and F-measure by 3% compared to the deep-LSTM ensemble model [1], SIRVD-DL [2], Deep learning method [3], and MTGP regression [4]. Additionally, the ALTDFETRCNLC technique reduces prediction time by 6%, error rate by 54%, and space complexity by 7% compared to existing methods.

A complete list of abbreviations is shown in *Appendix I.*

5.1 Limitation

Our proposed model utilizes a CNN approach for making predictions. While it does demonstrate improved prediction performance compared to other contemporary approaches, the achieved prediction accuracy may still not be significantly better. Furthermore, for certain datasets, a lower accuracy rate is observed. Considering the objectives, comparative study, and study limitations, it becomes evident that the proposed model may not cater to all types of COVID-19 cases unless the prediction accuracy for all datasets reaches 98-99% or higher.

In cases where the number of features in the dataset varies, the proposed model fails to achieve superior classification performance. This leads to increased time required for efficient data classification and prediction. Additionally, there are certain parameters that have not been sufficiently focused on, such as specificity, feature selection time, and preprocessing time.

6.Conclusion and future work

The COVID-19 pandemic poses a global health risk, infecting individuals worldwide. It has led to increased mortality rates and has also had a significant impact on economic stability. To address this, it is crucial to mitigate the mortality rate, allowing the government to implement other necessary measures. ALTDFETRCNLC is specifically designed for early detection of COVID-19, providing accurate data results.

The process begins with data preprocessing, which involves normalizing the data using data normalization techniques. Relevant features are then selected using the Tversky similarity-indexed stochastic distributive embedding technique. Finally, a time-dependent cox regressive Levenberg–Marquardt convolutional neural learning classifier is employed for COVID-19 classification, resulting in improved accuracy.

Simulation is conducted using various performance parameters, including accuracy, error rate, time, and space complexity. The comprehensive performance analysis demonstrates that the proposed ALTDFETRCNLC achieves higher accuracy in COVID-19 prediction while requiring less time compared to conventional techniques. In future work, it is recommended to explore CNN methods for identifying COVID-19 patients through classification techniques, using different datasets.

Acknowledgment
None.

Conflicts of interest
The authors have no conflicts of interest to declare.

Author's contribution statement
Kalaiselvi S R: Conceptualization, investigation, writing – original draft, writing – review and editing-data collection, conceptualization, writing – original draft, analysis, and interpretation of results. **Vijayabhanu R:** Study conception, design, data collection, supervision, investigation of challenges, and draft manuscript preparation.

References
[1] Zaki F, Afifi F, Gani A, Anuar NB. Granular network traffic classification for streaming traffic using incremental learning and classifier Chain. Malaysian Journal of Computer Science. 2022; 35(3):264-80.

[2] Liao Z, Lan P, Fan X, Kelly B, Innes A, Liao Z. SIRVD-DL: A COVID-19 deep learning prediction model based on time-dependent SIRVD. Computers in Biology and Medicine. 2021; 138.

[3] Ahuja S, Shelke NA, Singh PK. A deep learning framework using CNN and stacked Bi-GRU for COVID-19 predictions in India. Signal, Image and Video Processing. 2022:1-8.

[4] Ketu S, Mishra PK. Enhanced Gaussian process regression-based forecasting model for COVID-19 outbreak and significance of IoT for its detection. Applied Intelligence. 2021; 51:1492-512.

[5] Ketu S, Mishra PK. India perspective: CNN-LSTM hybrid deep learning model-based COVID-19 prediction and current status of medical resource availability. Soft Computing. 2022; 26:645-64.

[6] Castillo Ossa LF, Chamoso P, Arango-López J, Pinto-Santos F, Isaza GA, Santa-Cruz-González C, et al. A hybrid model for COVID-19 monitoring and prediction. Electronics. 2021; 10(7):1-13.

[7] Le DN, Parvathy VS, Gupta D, Khanna A, Rodrigues JJ, Shankar K. IoT enabled depthwise separable convolution neural network with deep support vector machine for COVID-19 diagnosis and classification. International Journal of Machine Learning and Cybernetics. 2021:1-4.

[8] Iwendi C, Mahboob K, Khalid Z, Javed AR, Rizwan M, Ghosh U. Classification of COVID-19 individuals using adaptive neuro-fuzzy inference system. Multimedia Systems. 2021:1-5.

[9] Abbasimehr H, Paki R. Prediction of COVID-19 confirmed cases combining deep learning methods and Bayesian optimization. Chaos, Solitons & Fractals. 2021; 142:1-14.

[10] Gupta A, Jain V, Singh A. Stacking ensemble-based intelligent machine learning model for predicting post-COVID-19 complications. New Generation Computing. 2022; 40(4):987-1007.

[11] Verma H, Mandal S, Gupta A. Temporal deep learning architecture for prediction of COVID-19 cases in India. Expert Systems with Applications. 2022; 195:1-11.

[12] Shorten C, Khoshgoftaar TM, Furht B. Deep learning applications for COVID-19. Journal of Big Data. 2021; 8(1):1-54.

[13] Kallel A, Rekik M, Khemakhem M. Hybrid-based framework for COVID-19 prediction via federated machine learning models. The Journal of Supercomputing. 2022; 78(5):7078-105.

[14] Ye H, Wu P, Zhu T, Xiao Z, Zhang X, Zheng L, et al. Diagnosing coronavirus disease 2019 (COVID-19): efficient Harris hawks-inspired fuzzy K-nearest neighbor prediction methods. IEEE Access. 2021; 9:17787-802.

[15] Bhardwaj R. A predictive model for the evolution of COVID-19. Transactions of the Indian National Academy of Engineering. 2020; 5(2):133-40.

[16] Zivkovic M, Bacanin N, Venkatachalam K, Nayyar A, Djordjevic A, Strumberger I, Al-Turjman F. COVID-19 cases prediction by using hybrid machine learning and beetle antennae search approach. Sustainable Cities and Society. 2021; 66:1-22.

[17] Gautam Y. Transfer learning for COVID-19 cases and deaths forecast using LSTM network. ISA Transactions. 2022; 124:41-56.

[18] Lucas B, Vahedi B, Karimzadeh M. A spatiotemporal machine learning approach to forecasting COVID-19 incidence at the county level in the USA. International Journal of Data Science and Analytics. 2023; 15(3):247-66.

[19] Ahmed I, Ahmad A, Jeon G. An IoT-based deep learning framework for early assessment of COVID-19. IEEE Internet of Things Journal. 2020; 8(21):15855-62.

[20] Chandra R, Jain A, Singh Chauhan D. Deep learning via LSTM models for COVID-19 infection forecasting in India. PloS one. 2022; 17(1):1-28.

[21] Shastri S, Singh K, Kumar S, Kour P, Mansotra V. Deep-LSTM ensemble framework to forecast Covid-19: an insight to the global pandemic. International Journal of Information Technology. 2021; 13:1291-301.

[22] Sinha T, Chowdhury T, Shaw RN, Ghosh A. Analysis and prediction of COVID-19 confirmed cases using deep learning models: a comparative study. In advanced computing and intelligent technologies: proceedings of ICACIT 2021 (pp. 207-18). Springer Singapore.

[23] Abedin MZ, Moon MH, Hassan MK, Hajek P. Deep learning-based exchange rate prediction during the COVID-19 pandemic. Annals of Operations Research. 2021:1-52.

[24] Sakthivel R, Thaseen IS, Vanitha M, Deepa M, Angulakshmi M, Mangayarkarasi R, et al. An efficient hardware architecture based on an ensemble of deep learning models for COVID-19 prediction. Sustainable Cities and Society. 2022; 80:1-13.

[25] Solayman S, Aumi SA, Mery CS, Mubassir M, Khan R. Automatic COVID-19 prediction using explainable machine learning techniques. International Journal of Cognitive Computing in Engineering. 2023; 4:36-46.

[26] Cui Z, Wu J, Lian W, Wang YG. A novel deep learning framework with a COVID-19 adjustment for electricity demand forecasting. Energy Reports. 2023; 9:1887-95.

[27] Mareeswari V, Vijayan R, Sathiyamoorthy E, Ephzibah EP. A narrative review of medical image processing by deep learning models: origin to COVID-19. International Journal of Advanced Technology and Engineering Exploration. 2022; 9(90):623-43.

[28] Motwani A, Shukla PK, Pawar M, Kumar M, Ghosh U, Alnumay W, et al. Enhanced framework for COVID-19 prediction with computed tomography scan images using dense convolutional neural network and novel loss function. Computers and Electrical Engineering. 2023; 105:1-15.

[29] Xu Y, Lam HK, Jia G, Jiang J, Liao J, Bao X. Improving COVID-19 CT classification of CNNs by learning parameter-efficient representation. Computers in Biology and Medicine. 2023; 152:1-10.

[30] Ayris D, Imtiaz M, Horbury K, Williams B, Blackney M, See CS, et al. Novel deep learning approach to model and predict the spread of COVID-19. Intelligent Systems with Applications. 2022; 14:1-12.

[31] Kumar V, Zarrad A, Gupta R, Cheikhrouhou O. COV-DLS: prediction of COVID-19 from X-rays using enhanced deep transfer learning techniques. Journal of Healthcare Engineering. 2022; 2022:1-3.

[32] Kafieh R, Arian R, Saeedizadeh N, Amini Z, Serej ND, Minaee S, et al. COVID-19 in Iran: forecasting pandemic using deep learning. Computational and Mathematical Methods in Medicine. 2021; 2021:1-16.

[33] Aftab M, Amin R, Koundal D, Aldabbas H, Alouffi B, Iqbal Z. Classification of COVID-19 and influenza patients using deep learning. Contrast Media & Molecular Imaging. 2022; 2022:1-11.

[34] Muneer A, Fati SM, Akbar NA, Agustriawan D, Wahyudi ST. iVaccine-deep: prediction of COVID-19 mRNA vaccine degradation using deep learning. Journal of King Saud University-Computer and Information Sciences. 2022; 34(9):7419-32.

[35] Malik H, Naeem A, Naqvi RA, Loh WK. DMFL_Net: a federated learning-based framework for the classification of COVID-19 from multiple chest diseases using X-rays. Sensors. 2023; 23(2):1-23.

[36] Paul SG, Saha A, Biswas AA, Zulfiker MS, Arefin MS, Rahman MM, et al. Combating Covid-19 using machine learning and deep learning: Applications, challenges, and future perspectives. Array. 2022.

[37] Wang L, Yin Z, Puppala M, Ezeana CF, Wong KK, He T, et al. A time-series feature-based recursive classification model to optimize treatment strategies for improving outcomes and resource allocations of COVID-19 patients. IEEE Journal of Biomedical and Health Informatics. 2021; 26(7):3323-9.

[38] Xu L, Magar R, Farimani AB. Forecasting COVID-19 new cases using deep learning methods. Computers in Biology and Medicine. 2022; 144:1-7.

[39] Reshi AA, Rustam F, Mehmood A, Alhossan A, Alrabiah Z, Ahmad A, et al. An efficient CNN model for COVID-19 disease detection based on X-ray image classification. Complexity. 2021; 2021:1-2.

[40] Huyut MT. Automatic detection of severely and mildly infected COVID-19 patients with supervised machine learning models. Innovation and Research in BioMedical Engineering. 2023; 44(1):1-12.

[41] Arora P, Kumar H, Panigrahi BK. Prediction and analysis of COVID-19 positive cases using deep learning models: a descriptive case study of India. Chaos, Solitons & Fractals. 2020; 139:1-9.

[42] Alassafi MO, Jarrah M, Alotaibi R. Time series predicting of COVID-19 based on deep learning. Neurocomputing. 2022; 468:335-44.

[43] Sahin ME, Ulutas H, Yuce E, Erkoc MF. Detection and classification of COVID-19 by using faster R-CNN and mask R-CNN on CT images. Neural Computing and Applications. 2023:1-5.

[44] Moujahid H, Cherradi B, Al-Sarem M, Bahatti L, Eljialy AB, Alsaeedi A, et al. Combining CNN and grad-cam for COVID-19 disease prediction and visual explanation. Intelligent Automation & Soft Computing. 2022; 32(2): 723-45.

[45] https://www.kaggle.com/datasets/imdevskp/covid19-corona-virus-india-dataset. Accessed 20 April 2023.

Ms. Kalaiselvi S R currently serves as an Assistant Professor in the Department of Computer Science at Dr.N.G.P. Arts and Science College in Coimbatore, Tamil Nadu, India. Additionally, she is a research scholar in the Department of Computer Science at Avinashilingam Institute for Home Science and Higher Education for Women, also located in Coimbatore, India. She has an impressive publication record, with three papers published in UGC Journals and one paper published in Scopus. Furthermore, she has presented a total of four papers at international conferences.
Email: srkalai2010@gmail.com

Dr. Vijayabhanu R is an Assistant Professor-SG in the Department of Computer Science at Avinashilingam Institute for Home Science and Higher Education for Women, located in Coimbatore. She obtained her Ph.D. degree from the same institution in 2014. Her research expertise lies in the areas of Data Mining, Image Processing, and Soft Computing. She has made significant contributions to the academic field. She has published three book chapters and authored a book. Additionally, she has published 30 research papers in international journals and presented 12 papers at international conferences.
Email: vijayabhanu_cs@avinuty.ac.in

Appendix I

S. No.	Abbreviation	Description
1	ALTDFETRCNLC	Additive Log Transformed Distributive Feature Embedding Time-dependent Regressive Convolutional Neural Learning Classifier
2	ANFIS	Adaptive Neuro-Fuzzy Inference System
3	Bi-GRU	Bidirectional Gated Recurrent Unit
4	Bi-LSTM	Bidirectional Long Short-Term Memory
5	BAS	Beetle Antennae Search
6	BR	Bagging Ridge
7	CNN	Convolutional Neural Network
8	COV-DLS	Novel Deep Transfer Learning Techniques
9	DMFL_Net	Decision-Making-Based Federated Learning Network
10	DWS-CNN	Depth wise Separable Convolution Neural Network
11	DSVM	Deep Support Vector Machine
12	GCN	Graph Convolutional Network
13	GRU	Gated Recurrent Units
14	IoT	Internet of Things
15	LSTM	Long short-Term Memory
16	MTGP	Multi-Task Gaussian Process
17	MB	Megabytes
18	NRM	Non-Parametric Regression Model
19	OHE	One Hot Encoder
20	PCRMDRPNLC	Proportional Cox regressive Mahout Deep Recurrent Perception Neural Learning Classifier
21	SIRVD-DL	Susceptible, Infected, Recovered, Vaccinated, and Deceased – Deep Learning Method
22	RNN	Recurrent Neural Network