Journal of Communicable Diseases
Volume 57, Issue 1 - 2025, Pg. No. 20-32
Peer Reviewed & Open Access Journal

Research Article

An Integrated Multi-Agent Frameworks for COVID-19 Detection Using Machine Learning and Deep Learning Techniques

Rupinder Kaur Walia[1], Harjot Kaur[2]

[1]Research Scholar, [2]Assistant Professor, Department of Computer Science Engineering. Guru Nanak Dev University, Amritsar, Punjab, India
DOI: https://doi.org/10.24321/0019.5138.202503

INFO

Corresponding Author:
Rupinder Kaur Walia, Department of Computer Science Engineering. Guru Nanak Dev University, Amritsar, Punjab, India
E-mail Id:
waliarupinder84@gmail.com
Orcid Id:
https://orcid.org/0000-0001-9304-4050
How to cite this article:
Walia R K, Kaur H. An Integrated Multi-Agent Frameworks for COVID-19 Detection Using Machine Learning and Deep Learning Techniques. J Commun Dis. 2025;57(1):20-32.

Date of Submission: 2025-02-06
Date of Acceptance: 2025-02-27

ABSTRACT

The SARSCoV-2 virus causes the infectious illness known as coronavirus disease (COVID-19). In recent decades, COVID-19 has become the most infectious disease. Millions of individuals worldwide suffer from this disease. Because of the restricted availability and sensitivity of testing kits, doctors and researchers have turned to computer tomography (CT) scans to screen for COVID-19. Recent technological developments and the widespread use of machine learning (ML) and deep learning (DL) techniques have shown high potential in terms of more accurate COVID-19 detection. Therefore, in this paper, we have developed two multi-agent frameworks for COVID-19 detection using the ML and DL algorithms. In the first framework, several ML algorithms, namely, KNN, LR, and SVM, are employed. Further, ensemble learning and hypertuning of the ML algorithms is done using the grid search method. Next, reinforcement learning method Q learning agent is used to update the multi-agent framework. On the other hand, the second multi-agent framework is developed with the help of lightweight ResNet and reinforcement PPO algorithms. Further, in both frameworks, feature selection is done using the LRP-ET method to determine the appropriate features from the dataset. The experimental results are performed on the Google Colab software for the standard dataset. The dataset is split into an 80:20 ratio to train and test the frameworks. The evaluation of both frameworks is done using the various parameters, namely, accuracy, precision, recall, and F-score. The results show that both frameworks outperform the existing models. Finally, the comparison of both frameworks shows that the second framework, which is based on deep learning, performs superiorly over the first framework, which is based on ML, due to efficiently handling complex data.

Keywords: COVID-19, Deep Learning, Ensemble Learning, KNN, LR, Machine Learning, Multi-Agent, Reinforcement Learning, SVM

Introduction

In late 2019, a new disease developed in China, infecting several individuals in a small market. At first, the illness was entirely unknown, but experts determined that its symptoms were comparable to those of the flu and coronavirus infection. Although the exact cause of this widespread illness was initially unidentified, the World Health Organization (WHO) recommended that the virus be named COVID-19 after laboratory testing and analysis of positive sputum by real-time polymerase chain reaction (PCR) test confirmed the infection. The COVID-19 virus rapidly spread across borders, causing harm to the world's health, economics, and welfare. According to data from Worldometers (worldometers.info), up to January 5, 2021, over 86 million individuals worldwide have COVID-19, and more than 1,870,000 of them have died as a result of the illness.[1]

The most common symptoms of the new coronavirus are a high fever, dry cough, dyspnea, myalgia, and headache. However, in many cases, the illness is asymptomatic, which poses an even greater risk to public health. The best way to diagnose COVID-19 is with the reverse transcript polymerase chain reaction (RT-PCR). However, quick and efficient screening of suspected situations is limited by a lack of resources and strict test environment requirements. In addition, there are instances when the RT-PCR examination produces misleading negative results. Unfortunately, the only way to successfully tackle this transmissible illness is to use precise drug/therapy methods and clinical vaccinations, both of which are now unavailable.[2]

One of the most serious illnesses that has seriously threatened human civilization is COVID-19. With the advancement of modern technologies in recent decades, inventive solutions to illness detection, prevention, and control have been developed using smart healthcare equipment and facilities. Particularly, CT and X-rays are two of the most effective imaging modalities for diagnosing COVID-19. When available, CT screening is preferred over X-rays due to its adaptability and three-dimensional pulmonary view, even though X-rays are more widely accessible and affordable. The management of the pandemic depends heavily on these conventional medical imaging techniques. By effectively producing high-quality diagnostic findings and significantly decreasing or eliminating manpower, artificial intelligence (AI), a developing software technology in the field of medical image analysis, has also directly assisted in the fight against the new coronavirus.[3-4] Machine learning and deep learning, two important branches of artificial intelligence, have recently gained popularity in COVID-19 models. These methods efficiently process the dataset and find out the hidden features from it, which helps to classify whether COVID-19 is present or not. In these models, there are four main stages: dataset read, pre-processing of the dataset, feature selection, and classification. In the first stage, the dataset related to COVID-19 is read. After that, in the second stage, pre-processing of the dataset is done to find out if there is any missing or null attribute in the dataset and determine the input and output attributes of the dataset. Further, the feature selection is done to select the most appropriate features from the dataset to train the ML/DL algorithm in the third step. In the fourth step, the ML/DL algorithm is utilized for COVID-19 detection purposes. In this research, we have developed two multi-agent frameworks for COVID-19 detection using ML and DL algorithms. The main contribution of this paper is summarized as follows:

- We have developed a feature selection method by combining the LRP and ET algorithms.
- We have proposed a MA-ML framework by utilizing ensemble learning, the grid search method to hyper-tune the machine learning algorithms, and reinforcement learning.
- We have proposed a MA-DL framework by designing a lightweight ResNet and reinforcement learning method such as Proximal Policy Optimization.
- The performance evaluation of both models shows that the proposed model achieves better accuracy, precision, recall, and F-score over the existing models.

The remaining manuscript has five sections. Section 2 shows the related work done in the COVID-19 detection models using the ML and DL. Section 3 explains the methodologies utilized to design the multi-agent framework for COVID-19 detection using the ML and DL algorithms along with datasets and performance metrics. Section 4 presents the proposed multi-agent frameworks that are designed for COVID-19 detection using the ML and DL. Section 5 shows the simulation results for both frameworks and the comparative analysis between them. Finally, the manuscript is concluded, and future work is defined in Section 6.

Related Work

In this section, a study of existing COVID-19 detection models is done, which are based on machine learning and deep learning algorithms. Abdulkareem et al.[5] presented a COVID-19 detection model based on the three ML algorithms, namely, SVM, RF, and NB. The accuracy achieved by these machine learning algorithms is 94.16% by SVM, 93.33% by RF, and 90.83 by NB, respectively. Further, in their work, feature selection is done using the brute force algorithm, which is a computationally expensive procedure. In,[6] they used the four-ML algorithm in the ensemble learning approach for COVID-19 detection and achieved an accuracy of 94.3%. Further, in their work, feature selection is done using the term frequency (TF)/inverse document frequency (IDF). In,[7] compare the performance of the DT

ISSN: 0019-5138
DOI: https://doi.org/10.24321/0019.5138.202503

and LR algorithms for COVID-19 detection. The result shows that they have achieved an accuracy of 97% by LR and 98% by DT algorithm. However, in their work, no preprocessing of the dataset is done. In ,[8] they used the chi-square correlation coefficient and logistic regression algorithms for feature selection and detection purposes by processing the COVID-19 dataset. Further, they have achieved an accuracy of 98.2%. The authors[9] compared the performance of the LR and SVM on the COVID-19 dataset, which is presented on the Kaggle dataset. The result shows that the LR outperforms SVM. In,[10] we used the ANN algorithm for COVID-19 detection purposes and selected the appropriate features using the phi coefficient correlation algorithm. Further, they have achieved an accuracy of 94%. The authors, Keles et al.,[11] used the two deep learning algorithms, CNN and ResNet, for multi-class classification of COVID, pneumonia, and normal. They have achieved 94.28% accuracy by the CNN model, with 97.61% accuracy by the ResNet model. In,[12] the CNN algorithm is used for COVID-19 detection by processing the dataset based on X-ray images. In their work, the dataset is prepared for CNN using image enhancement and augmentation methods. In their work, they have achieved an accuracy of 98%. The authors, Mohammedqasim et al.,[13] proposed a model by utilizing several deep learning algorithms, namely, CNN, ANN, and RNN. Further, they have pre-processed the dataset to balance it using the SMOTE algorithm and feature selection using the recursive elimination algorithm. Next, they have achieved the highest accuracy of 98%. Aslan et al.[14] used the tree seed algorithm (TSA) and optimized ANN for COVID-19 detection by processing the dataset prepared using the CT images. They have achieved an accuracy of 98.54%.

Next, we have defined the key points as determined from the existing study of COVID-19 detection models based on ML and DL.

- The feature selection from the dataset helps to enhance the efficiency of the ML/DL algorithm for COVID-19 detection. However, in the literature, traditional approaches are used, which are computationally expensive.
- In the existing study, standard ML/DL algorithms are utilized for COVID-19 detection purposes. However, the performance of the ML algorithm is highly dependent on the parameters and bias value of it.
- In the literature, most authors have designed a single-agent COVID-19 detection model.

In this research, these challenges are taken into consideration to design multi-agent frameworks based on ML and DL algorithms. Next, in both frameworks, reinforcement learning is utilized along with ML/DL algorithms. Further, in both frameworks, an optimal feature selection method is designed that is computationally less expensive, and hypertuning of the ML/DL algorithm is done to enhance the efficiency of the model.

Methodology

In this research work, we have presented two models. The first multi-agent framework is based on machine learning, whereas the second framework is based on the deep learning algorithm. In both frameworks, LRP and ET algorithms are combined for feature selection and reinforcement learning algorithms to enhance the accuracy of the model. The detailed description of datasets and algorithms is utilized in both frameworks, and performance metrics are given below.

Dataset: This study considers the Covid-19 dataset that is publicly accessible. The dataset was gathered at Hospital Israelita Albert Einstein in Sao Paulo, Brazil.[15] The gathered samples are described in detail below.

Early in 2020, samples were taken from patients to identify SARS-CoV2. The dataset includes 111 lab results from 5644 different people. In the dataset, around 10% of patients tested positive, of whom 6.5% needed critical care and hospitalization, and 2.5% needed both. There is no information on gender in the dataset. Based on the authors' current research, 18 laboratory results are crucial for understanding COVID-19 illness. Thus, we removed any leftover laboratory characteristics in order to balance the dataset and conduct COVID-19 identification. The number of individuals treated decreased from 5644 to 600 after the balancing procedure, which included 18 laboratory results from 600 patients. This is because some patients are unaware of part of the 18 laboratory results. There are 80 COVID-19 patients and 520 no findings in the balanced dataset.

Algorithms used for COVID-19 Detection

This section gives an overview of the algorithms are employed in feature selection, machine learning, deep learning, and reinforcement learning.

LRP Algorithm: This advanced method lets us determine the significance of various characteristics and how they affect the model's predictions. A DNN's prediction is linked to its input characteristics via LRP. Each characteristic is given a relevance value determined by how much it contributes to the final outcome.[16] LRP may be represented mathematically as equation 1:

$$R_{xj}^{(l)} = \sum_k \frac{a_{jk}^{(l)}.R_k^{(l+1)}}{\sum_i a_{ik}^{(l)}} \tag{1}$$

where x and l represent the feature and layer of the DL model, respectively, $R_{xj}^{(l)}$ and $a_j^{(l)}$ and for weight and biases, respectively, represent the relevance score for each feature in layer l.

ISSN: 0019-5138
DOI: https://doi.org/10.24321/0019.5138.202503

According to the proposed approach, the phase that comes after LRP is devoted to removing the biases and weights that the trained model produced. These biases and weights are crucial because they capture the value of every feature in the disease identification process. A mean-based method is used to calculate each feature's overall relevance weight (equation 2). With this method, the mean, or average, of the weight and bias that are retrieved for each feature is calculated. The formulation is as follows:

$$R_j^l = \left(\frac{R_{xj}^i + b_j^i}{2}\right) \quad (2)$$

where b_j^i is the bias values of a single feature, R_{xj}^i indicates the relevance score for each feature in layer l, and R_j^l is the final relevance weight for each feature xj. To optimize disease detection within the medical domain, this final weightage is intended to preserve the most significant and informative features. It provides a fair and more accurate analysis of each feature's contribution to the disease identification process by averaging the weights given by the retrieved weight and bias. In order to improve the efficiency and accuracy of disease detection, this phase is a crucial part of our proposed feature selection module. It guarantees that only the most important characteristics are kept, making it a useful addition to this area.

ET: Extra-trees use the whole training dataset to create a huge number of distinct decision trees. The method selects a split rule for the root node using a randomly selected collection of characteristics (K) and a partly random cut point.[17] The parent node is divided into two random child nodes using a random split selection. Until you reach a leaf node, this procedure is repeated in each child node. Nodes without children are known as leaf nodes. A majority vote is used to determine the final forecast, which is the sum of the predictions made by each tree. The Gini significance is calculated for each feature throughout the forest's building in order to carry out feature selection. The Gini significance of each attribute determines its ranking, which is downwards. The top k characteristics are then chosen by the user to serve as input for the categorization model.

SVM: This kind of machine learning approach has shown to be successful in recognition of diseases. The supervised learning method determines the largest margins between training data samples and unknown test data by choosing the hyper-plane or the decision boundary denoted by the solution vector w. The two most often used SVM variations are nonlinear SVM with a Radial Basis Function (RBF) kernel and linear SVM.[18] Equation (3) represents the linear SVM binary classifier. In pattern classification, nonlinear SVM with RBF kernel (Equation (4)) has shown highly promising results with a broad range of applications. Now consider the training samples $\{y_i, x_i\}_{i=1}^n$, that has the label y_i $\in$. This shows the feature vector's class in d feature dimensions. Using these metrics, the hyperplane for the model is defined as follows:

$$H(x) = w^T x + b = \sum_{i=1}^n w_i x_i + b_i \quad (3)$$

$$H(x) = \sum_{i=1}^n w_i x_i k(x, v_t) + b \quad (4)$$

LR: Based on statistical data, a logistic regression analysis model shows how the expected variable, which is made up of two or more categories, is related to one or more independent factors on a continuous or category scale.[19] The logistic regression approach has many of the same methods and processes as the linear regression approach. The logistic regression approach utilizes the maximum likelihood estimation (MLE) technique to estimate the parameter values, while the linear regression approach often uses the ordinary least squares (OLS) model. The MLE approach converts observed objects into simple regression coefficients by optimizing the likelihood of categorizing them into relevant groups.

KNN: One of the simplest nonparametric pattern recognition methods is KNN.[20] A label is assigned via the KNN method based on the labels that are most prevalent among its k-nearest neighbors. The primary benefits of the KNN classifier are its ease of use and the fact that it only requires two tuning parameters: k and the distance measure. The KNN algorithm's step-by-step process:

- Set the number of nearest neighbors (k) to its initial value.
- Calculate the distance between the test picture and each training image. Any criterion for distance might be used. For instance, Euclidean distance is usually used and is controlled by the formula

Distance (a, b) = ||a – b|| (5)

In the above equation, are feature space's two different samples.

- Determine the closest neighbor by sorting the distances and using the kth minimal distance.
- Obtain the appropriate labels for the training data below k for the sorted condition.
- Assign the output label to the majority of k-nearest neighbors.
- **ResNet Algorithm:** Eight fundamental layers are there in the proposed lightweight ResNet-based DNN architecture: input, reshape, Conv2D, batch normalization, activation, residual block, global average pooling, and dense layers.
- **Input Layer:** has the ability to accept 100x10 2D data that represents features chosen by the LRP-ET Deep feature selection process.
- **Reshape Layer:** To make the 2D data compatible with Conv2D layers, it is converted into a 4D format (None, 100, 10, 1).
- **Conv2D Layer:** uses 16 3x3 filters to process the reshaped data, collecting temporal elements that are crucial for identifying patterns linked to COVID-19.

ISSN: 0019-5138
DOI: https://doi.org/10.24321/0019.5138.202503

- **Batch Normalization:** improves the network and promotes convergence by normalizing activations inside each mini-batch after convolution.
- **Activation Function:** uses ReLU activation to capture complex correlations and add non-linearity.
- **Residual Blocks:** contains two blocks to improve feature extraction and solve vanishing gradient problems, each containing two Conv2D layers and a shortcut connection.
- **Conv2D Layers:** Convolutional techniques are used to extract features.
- **Shortcut Connection:** aids in training deeper networks and promotes gradient flow.
- **Global Average Pooling:** minimizes the output from residual blocks' spatial dimensions prior to categorization.
- **Dense Layer:** divides the data into two groups according to characteristics that have been retrieved using a SoftMax activation function.

With its simplified architecture and lightweight ResNet-based DL model, this model effectively analyzes data, enhancing accuracy and usefulness for applications such as plant disease identification, real-time traffic sign recognition, and disease detection.

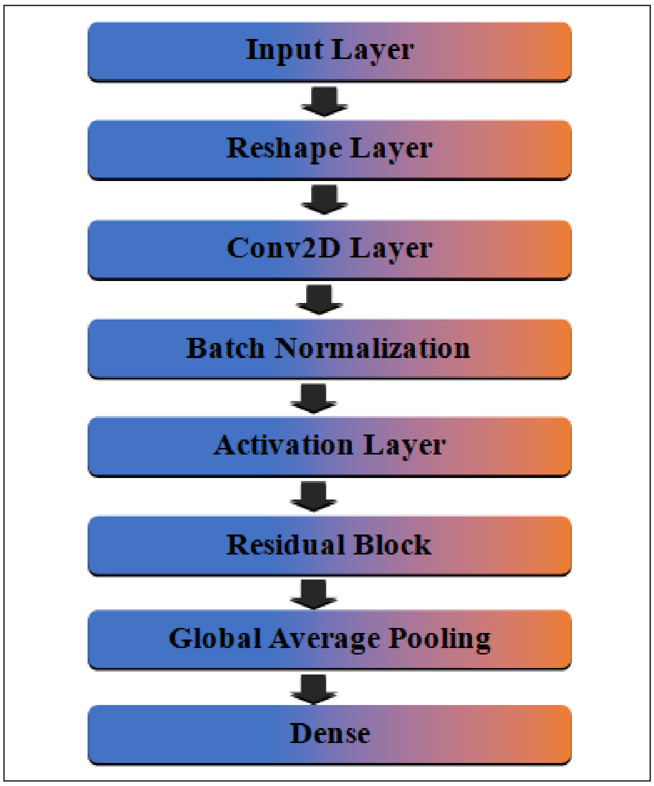

Figure 1.Lightweight ResNet

- **Reinforcement Learning:** To improve the model's performance, two reinforcement learning techniques are used in this study's proposed MA-ML and MA-DL architecture.
- **Deep Q-Learning (DQL):** In reinforcement learning (RL), an agent learns the appropriate behaviour for every environmental condition based on the incentives it receives.[21] From its starting state to its end state, the optimum strategy is the one that maximizes the agent's cumulative rewards. Q-learning is the algorithm that determines which actions are optimal. The foundation of the Q-Learning method is calculating the value for every state-action pair. In other words, it informs the agent in each stage about the behaviour that will provide greater cumulative rewards in the future based on the experience. To keep track of the estimated value for every action-state combination, Q-Learning uses a table called Q-Table. Q-learning with hundreds of states uses an ANN to mimic the Q-Table instead of a physical table. This algorithm's ability to learn from mistakes is what makes it interesting. Bellman's equation enables DQL to connect one state with the others, making it particularly well-suited for identifying the optimal sequences of actions across the states to achieve the highest accumulative rewards. After training, the agent can reliably predict the reward summation from the present condition to the end state.
- **PPO Algorithm:** A PPO agent is a reinforcement learning agent that limits the difference between the distributions of the new and old policies by optimizing a policy using a clipped surrogate objective function. This ensures reliable and constant updates. The following describes how the PPO algorithm works.
- **Collect Data:** Using its current policy, the agent interacts with the environment to collect trajectories, which include actions, states, and rewards.
- **Compute Advantage:** Estimate the advantage function, which analyzes how superior an action is to the norm. Usually, the value function is used to estimate the expected future benefits.
- **Update Policy:** Utilize the gathered information to maximize a clipped surrogate goal function to update the policy. This feature is designed to reduce instability by ensuring that policy changes are not too massive. Exploration and exploitation are balanced by the clipping mechanism, which limits the policy ratio to a range of around 1.
- **Update Value Function:** Additionally, make improvements to the value network's estimates of expected profits in based on the new policy.
- **Iterate:** To constantly enhance the performance of the policy, repeat the steps of data gathering, advantage calculation, and policy/value modifications.

The following stages are also included in the reinforcement learning module.

ISSN: 0019-5138
DOI: https://doi.org/10.24321/0019.5138.202503

- **Initialization:** Setup the agent and environment, taking into account the value and policy operations.
- **Interaction:** The agent follows its policy (PPO) while selecting and carrying out actions.
- **Feedback:** Observe the new situation, and the environment will reward you.
- **Update:** Calculate benefits, modify value functions, and update the policy (PPO).
- **Repeat:** To enhance decision-making and optimize benefits, iterate the procedure.
- **Performance Metrics:** In this section, the performance metrics are defined which are considered to evaluate the developed framework in this research.[22]
- **Accuracy:** It evaluates the classifier's total number of correct predictions. The following formula is used to determine the accuracy:

$$A = \frac{TP+TN}{TP+TN+FP+FN} \tag{6}$$

Precision: The precision measures how many forecasts of the positive class really fall into the positive class. The following formula may be used to determine it:

$$P = \frac{TP}{TP+FP} \tag{7}$$

Recall: It is calculated as the ratio of the total number of positive forecasts to the number of real positive predictions. It is assessed as follows:

$$R = \frac{TP}{TP+FN} \tag{8}$$

F-Score: The F-measure considers both precision and recall problems and produces a single score.

$$F - Score = \frac{2*P*R}{P+R} \tag{9}$$

In Eq. (6-9), TN, TP, FN, and FP denote the following things.

- **True Negative (TN):** when the predicted and actual values are both negative.
- **True Positive (TP):** when both predicted and actual values are positive.
- **False Negative (FN):** when the predicted value is negative while the actual value is positive.
- **False Positive (FP):** When the predicted value is positive, but the real value is negative

Proposed Multi-Agent Frameworks for COVID-19 Detection

This section presents the multi-agent frameworks that are designed for COVID-19 detection using the ML and DL algorithms. In the first framework, the COVID-19 detection model is enhanced by using ensemble learning, fine-tuning the ML parameters using the grid search method, and reinforcement Q-learning agents. On the other hand, in the second framework, deep learning and reinforcement learning, namely, lightweight ResNet and PPO, are employed for the built COVID-19 detection model. Further, in both frameworks, we have developed a feature selection algorithm designed by hybridizing the LRP and ET to eliminate the irrelevant features that have the least impact on the diagnosis model. Also, feature selection helps to overcome the computation time and enhance the performance of the detection model. The detailed description of the proposed multi-agent frameworks is given below.

Proposed Multi-Agent Framework for COVID-19 Detection using ML Algorithms

Figure 2 shows the flowchart of the MA-ML framework is designed for COVID-19 detection. Initially, the standard dataset is read and preprocessing is done using the various approaches, namely, fit transform, handling missing data, and initial correlation-based feature removal to prepare it for the proposed model. Further, the most appropriate feature selection is done using the hybrid combination of the LRP and DT methods due to being computationally less expensive and effectively processing the complex data. In the feature selection method, appropriate features using the LRP and DT algorithms are determined; after that, these features are added. The pseudocode for pre-processing of the dataset and feature selection is shown in Table 1. Next, the dataset is split into an 80:20 ratio to train and test the ML algorithm for COVID-19 detection. In this research, three ML algorithms—SVM, LR, and KNN—are used for COVID-19 detection purposes using the ensemble learning approach.

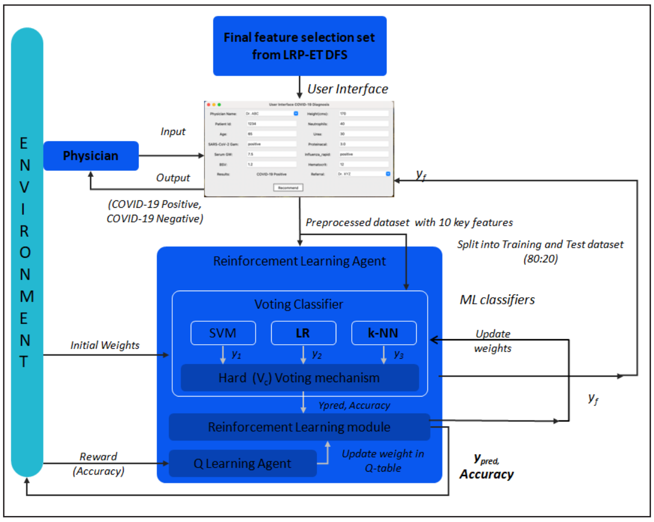

Figure 2.Flowchart of MA-ML Framework for COVID-19 Detection

In the ensemble learning approach, numerous ML algorithms are trained to tackle the same problem and combined in the output to enhance the efficiency of the model. By combining many weak learners (also known as basis models or first-stage models), more complex models may be produced. In most cases, these fundamental models are not sufficient on their own due to either too much bias or too much variance. By combining several weak learners, ensemble approaches aim to reduce their bias and variation

ISSN: 0019-5138
DOI: https://doi.org/10.24321/0019.5138.202503

and create a strong learner with better results. By properly integrating weak models, we may produce more accurate or dependable models. Stacking ensemble models are designed using base models and a meta-learner (or second-stage model) that employs base-model predictions. Predictions are generated using the basic models, which are trained on the training data. To aggregate the base-model predictions, the meta learner is then trained on the choices made by base models using data that hasn't been seen before. To do this, the base learners' inputs and output pairs of data are sent into the meta-learner, which is then trained to predict the ideal result.In this research, a hard voting mechanism is used to determine which ML outperforms over others. Further, hyper-parameter tuning of the ML algorithm is done using the grid search method in order to enhance the efficiency of the model. Finally, a user interface is designed in which the physician must input the patient-related attribute, which is based on the feature selection method. Further, a reinforcement Q-learning algorithm is utilized to update the multi-agent environment. Table 2 shows the pseudocode of ensemble learning, hypertuning, and reinforcement learning for MA-ML model.

Table 1.Pseudocode for Pre-Processing of the Dataset and Feature Selection for the MA-ML Model

Input : RID : raw input dataset ← read (dataset.xlsx)   #raw COVID-19 dataset with 43 features
Output :PPSFD: Final Pre-processed and Selected Features dataset   #10 selected key features
Procedure : Data Pre-processing Method
begin procedure
1. Implement Pre-Processing Techniques on Dataset
• Implement Fit Transform (data[SARS-COV-2 exam results])
• For every feature column x in RID:
• If missing_data (x) > 85%
▪ Delete Feature Column x   # Handle Missing Data in dataset
• $r = \frac{\Sigma((X-\bar{X})(Y-\bar{Y}))}{\sqrt{\Sigma(X-\bar{X})^2 * \Sigma(Y-\bar{Y})^2}}$ # Calculate correlation among features
where, r is Pearson Correlation Coefficient, X and Y represent two different feature columns and X̄ and Ȳ are mean values of the respective feature columns.
• ρ ← 0.92   # Set threshold for Correlation
▪ if | x | < ρ,
▪ Remove Feature Column x from RID
end procedure
Procedure : LRP-ET Deep Feature Selection
begin procedure
2. Define Training Dataset
• Data Train (x) = new_df (RID)
• Label Train (y) = RID [datset-cov-2_exam_result]

3. Building three layer DNN Model
• DL_Model ← DNN Model
• Train DL_ model (x,y)   # Training DL model with x and y data.
• For each feature x and layer i in DL_model;
○ x.$a_j^{(l)}$ and x. $b_j^{(l)}$ ← (layer.get_weights(), layer.get_bias()) # Extract weights ($a_j^{(l)}$) and biases ($b_j^{(l)}$) of each feature x.
4. Layer-wise Relevance Propagation(LRP) Implementation
• Initialize LRP (DL_model, x)   # Initializing LRP with 2 inputs
• For each feature x_j = i to n :
○ Read ($(a_j^{(l)}, b_j^{(l)})$)   # Getting weight and biases values from DL_model
○ $R_{xj}^{(l)} = \Sigma_K \frac{a_{jk}^{(l)} \cdot R_k^{(l+1)}}{\Sigma_i a_{ik}^{(l)}}$ # Calculate relevance score for each feature x_j in layer l
○ $R_{x_j} = \left(\frac{R_j^i + b_j^l}{2}\right)$ # Calculate final relevance weight for each feature x_j
5. Extra Tree(ET) Implementation
• ET.train(x,y)   # Train ET on x and y data
• F_x = ET. f_x # Calculate feature importance using ET
•   # Combining feature importance and Relevance values to form final feature set.
• PPSFD = .head(10)   # Selecting Top 10 features from data
$C_x = F_x + R_x$
end procedure

ISSN: 0019-5138
DOI: https://doi.org/10.24321/0019.5138.202503

Table 2.Pseudocode for MA-ML Model After Feature Selection

1. Call **LRP-ET Deep Feature Selection** # Call procedure LRP-ET defined in Algorithm1
2. Divide the Preprocessed Dataset PPSFD(X,y) with 10 key features into training and test datasets with ratio (80:20)
3. Define the Voting Environment (VE) for the Reinforcement Learning (RL) Agent with initial weights
4. Initialize hyperparameter-tuned ML Classifiers # k-NN, LR and SVM classifiers with grid search optimization.
5. **clf1=** k Neighboursclassifier()
6. **clf2=** LogisticRegression (max_iter=1000)
7. **clf3=** SVC(probability= True)
8. Create a VotingClassifier (VC) using majority voting mechanism for classifiers clf1,clf2,clf3 in a reinforcement learning environment.
9. Train and Test the Voting Classifier (VC) with initial weights.
10. Define a Q-Learning Agent (QA) # Reinforcement Learning (RL) Agent
11. Initialize Voting Environment and Q-Learning agent
12. Train and Run the Q-Learning agent and update weights in Q_table till best set of weights are generated.
13. Pass the updated weights to Reinforcement Learning module
14. Test Voting Classifier with the updated weights till highest accuracy score is achieved. #Best set of weights
15. FCD (COVID-19 positive, COVID-19 negative) ← y_f#Assign the final predicted and classified output with highest accuracy score.

Proposed Multi-Agent Framework for COVID-19 Detection using DL and Reinforcement Learning Algorithms

Figure 3 shows the ML-DL framework is designed for COVID-19 detection. In this model, standard dataset, preprocessing, and feature selection are done in the same way as in the MA-ML model. After that, the dataset split into 80:20. Further, in this research, a lightweight ResNet algorithm is designed to overcome the limitations of the ResNet algorithm, such as the computationally expensive and overfitting issue. Next, in order to achieve the multi-agent framework, the PPO algorithm is used to update the environment. Table 3 shows the pseudocode for the MA-DL model.

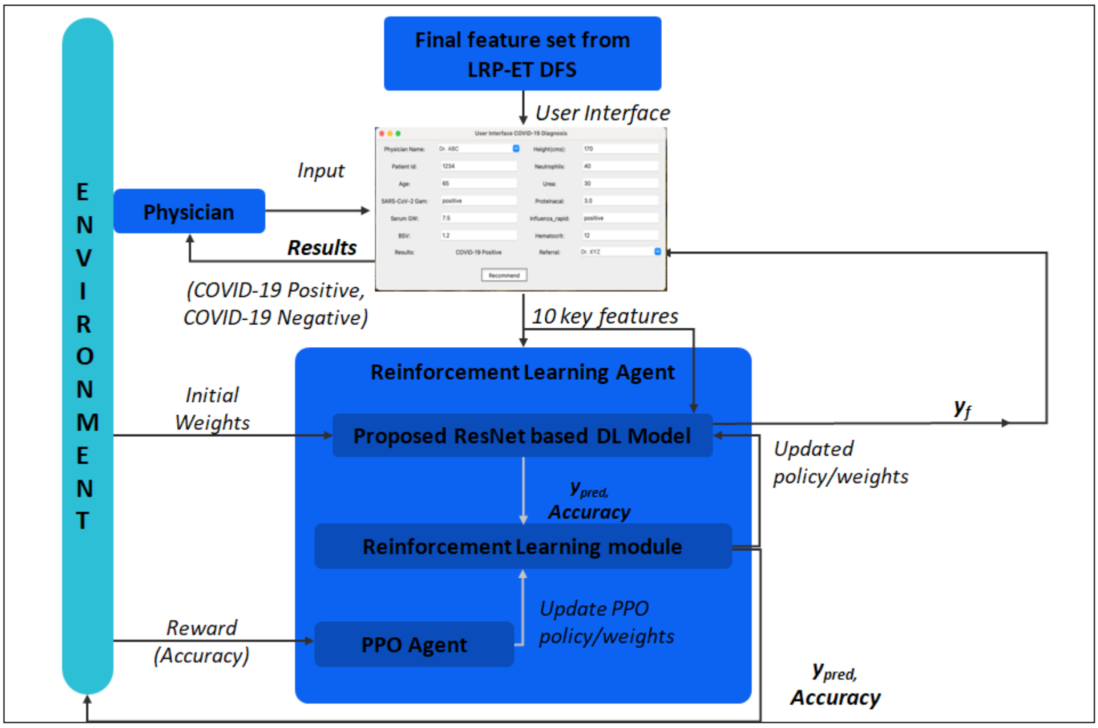

Figure 3.Flowchart of MA-DL Framework for COVID-19 Detection

ISSN: 0019-5138
DOI: https://doi.org/10.24321/0019.5138.202503

Table 3.Pseudocode for the MA-DL Framework

1. Call **LRP-ET** Deep Feature Selection # Call procedure LRP-ET defined in Algorithm1
2. Divide the Preprocessed Dataset PPSFD(X,y) with 10 key features into training and test datasets with ratio (80:20)
3. Define environment for the Reinforcement Learning Agent with initial weights/policy.
4. Initialize the proposed ResNet Based DL Model.
   - P_model= Pro_ResNet (input, num_fillters=16, kernel_size=3, strides=1, activation= 'relu')
5. Train the proposed DL model with initial weights/policy
6. Test the proposed model with initial weights/policy
7. Define PPO Agent (Proximal Policy Optimization) Agent #RL Agent
8. Initialize a hyperparameter tuned PPO agent with an initial/ custom policy
9. Train and Run the PPO agent and update weights till best set of weights are generated
10. Pass the updated weights policy to Reinforcement Learning Module
11. Test the proposed DL Model with the updated weights/policy till highest accuracy score is achieved. #Best set of weights.
12. FCD (COVID-19 positive, COVID-19 negative) ← y_f#Assign the final predicted and classified output with highest accuracy score.

Results and Discussion

In this section, we have presented the simulation results for multi-agent frameworks are developed for COVID-19 detection. In both frameworks, same dataset is utilized and code is written in python language and simulated on google colab with the system configuration of i7 processor, 8GB RAM, and 500GB of hard disk. Further, the developed frameworks are evaluated various performance metrics and compared with the existing models.

Simulation Results for Proposed Multi-Agent Framework for COVID-19 Detection using ML Algorithms

In this section, the simulation results for the proposed multi-agent framework on ML and reinforcement learning is explained. Table 4 shows the simulation setup configuration is sets for simulate the proposed MA-ML framework. In this table, feature selection, ML algorithm, and reinforcement learning algorithm parameter values are defined which are initialised for experimental purposes.

Figure 4 shows the confusion matrix is evaluated for the proposed MA-ML framework. The confusion matrix represents the four scenarios, namely, TP, FP, FN, and TN, as explained in the performance metrics. The results show that the proposed model is accurately predicts 1020 cases in the TP, 1 case in the FP, 27 case in the FN, and 81 cases in the TN, respectively. In the figure, lower values of FP and FN show that the proposed model is efficiently detect the COVID-19.

Figure 5 shows the ROC curve is plotted for the MA-ML framework between TPR vs. FPR. The result shows that the proposed model achieves the value in between 0.8-0.9 when compared to the threshold value (0.5) which reflects that the proposed model is efficiently distinguish the different classes.

Table 5 shows the comparative analysis of the proposed MA-ML framework with the existing models based on RF, Bernoulli NB, and SVM[5]. The result shows that the proposed model achieves the high accuracy, precision, recall, and F-score over the existing models due to optimal feature selection, followed by ensemble learning and hyper-tuning of the ML parameters.

Simulation Results for Proposed Multi-Agent Framework for COVID-19 Detection using DL and Reinforcement Learning Algorithms

Table 6 shows the simulation setup configuration of the second framework is developed using the DL and reinforcement learning. In this table, training and testing ratio, optimization, loss, metrics, no. of classes, epochs, and batch size value is defined which is taken for simulation purposes. Further, Table 7 shows the layer-wise simulation setup configuration of the DL algorithm.

Figure 6 shows the confusion matrix is determined for the framework is based on DL. The confusion matrix is evaluated by considering the four scenarios, TP, FP, FN, and FN. The figure shows that the proposed framework is efficiently predict the cases when compared to the true label. In the Figure, TP value is 105, TN value is 1021, FP value is 0, and FN value is 3.

Figure 7 shows the ROC curve for the MA-DL framework. The ROC curve is plotted to find out how efficiently the model is differentiated the different classes. The result shows that the proposed model achieves the value near to 1.0 value when compared to the threshold value (0.5). Thus, the proposed model provides excellent performance for COVID-19 detection.

Table 8 shows the comparative analysis based on various parameters, namely, accuracy, precision, recall, and F-score with the existing models based on RF, Bernoulli, and SVM[5]. The results indicate that the proposed model achieves the high accuracy value of 99.73, precision value of 100, recall value of 99.73, and F1-score value of 98.59.

Table 9 shows the comparative analysis of the proposed frameworks based on MA-ML and MA-DL. The results show that the MA-DL model outperforms over the MA-ML model due to effectively handle the large dataset.

Table 4.Simulation Setup Configuration of the MA-ML Framework

Parameter	Values
Training and Testing Ratio	80:20
KNN	Euclidean
SVM	Linear
LR	lbfgs
Voting Classifier	Hard
Hyper-tuning	Grid Search

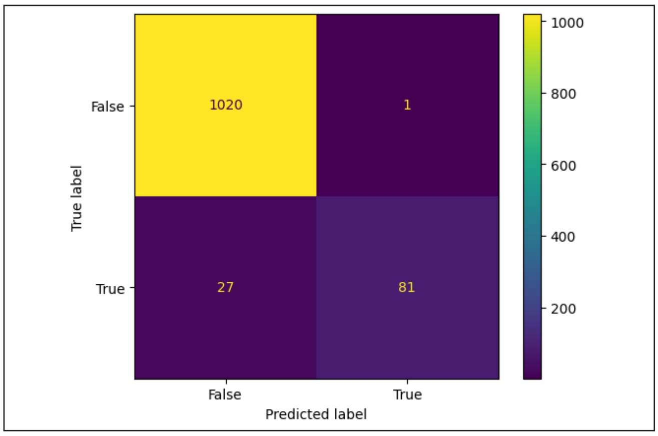

Figure 4.Evaluation of MA-ML Method using the Confusion Matrix

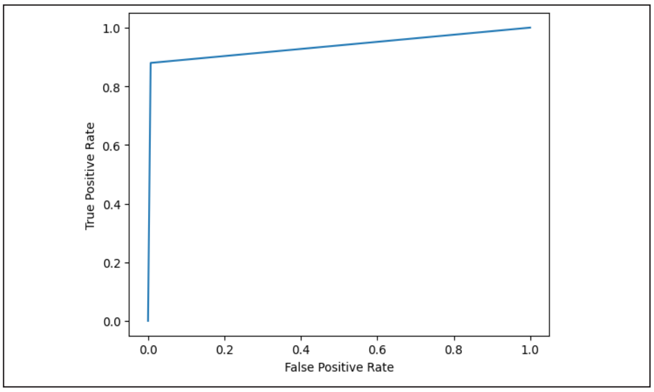

Figure 5.Evaluation of MA-ML Method using the ROC Curve

Table 5.Comparative Analysis with the Existing Models

Models	Accuracy	Precision	Recall	F-Score
RF [5]	94.16	95	94	93
Bernoulli NB [5]	92.5	86	93	89
SVM [5]	95	95	95	94
Proposed Model	97.5	98.7	97.5	97.51

Table 6.Simulation Setup Configuration for MA-DL Framework

Training Parameters	Values
Training data	80%
Test data	20%
Optimizer	Adam
Loss	Binary cross entropy
Metrics	Accuracy
No. of Classes	2
No. of Epochs	8
Batch size	32

Table 7.Simulation Setup Configuration for MA-DL Framework

Layer Type	Output Shape	Number of Parameters	Description
Input	None, 100, 10	0	Input layer that accepts 2D data of size 100x10.
Reshape	None, 100, 10, 1	0	Reshapes the input to 4D format for convolutional layers.

ISSN: 0019-5138
DOI: https://doi.org/10.24321/0019.5138.202503

Conv2D	None, 100, 10, 16	160	Initial convolutional layer with 16 filters of size 3x3.
Batch Normalization	None, 100, 10, 16	64	Batch normalization to normalize activations.
Activation ReLU	None, 100, 10, 16	0	ReLU activation function to introduce non-linearity.
Residual Block	None, 100, 10, 16	Varies	Two Conv2D layers with shortcut connection and ReLU.
Global Avg Pooling	None, 16	0	Global Average Pooling reduces spatial dimensions to 1x1.
Dense	None, 2	34	Output layer with 2 units and SoftMax activation.

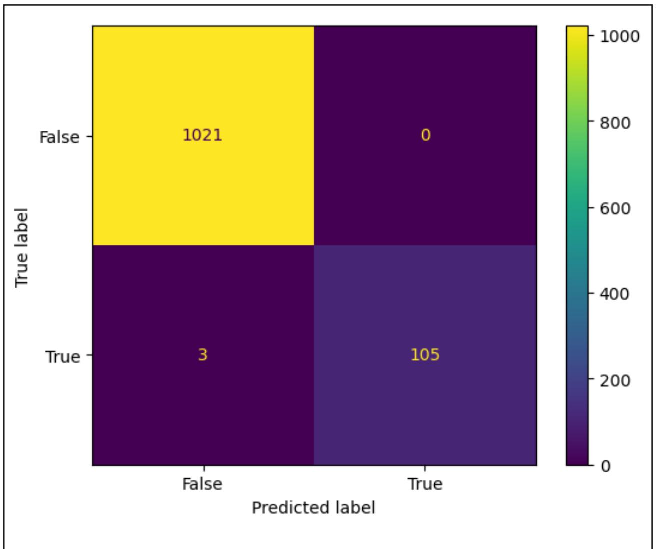

Figure 6.Confusion Matrix for MA-DL Framework

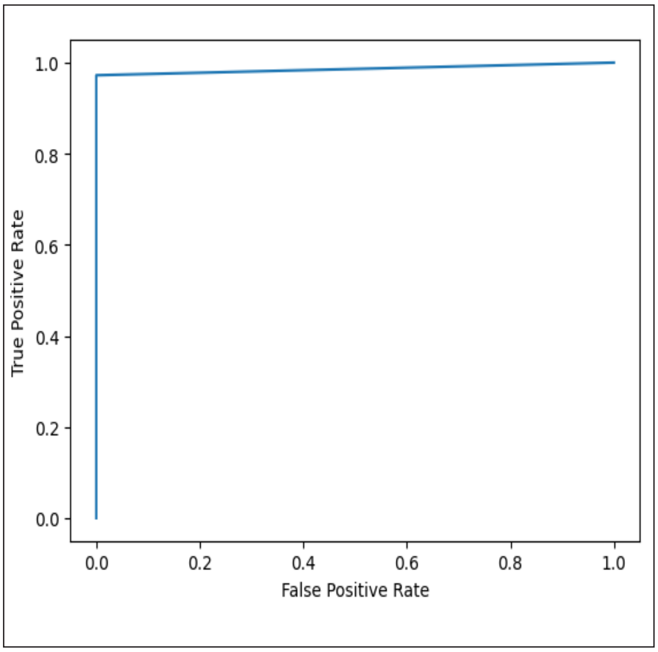

Figure 7.ROC Curve for MA-DL Framework

Table 8.Comparative Analysis of the MA-DL Framework with the Existing Models

Algorithm	Accuracy	Precision	Recall	F1-Score
RF [5]	94.16	95	94	93
Bernoulli NB [5]	92.5	86	93	89
SVM [5]	95	95	95	94
Proposed MA-DL Model	99.73	100	99.73	98.59

Table 9.Comparative Analysis of Proposed MA-ML and MA-DL Framework for COVID-19 Detection

Framework	Accuracy	Precision	Recall	F-Score
Proposed MA-ML Model	97.5	98.7	97.5	97.51
Proposed MA-DL Model	99.73	100	99.73	98.59

ISSN: 0019-5138
DOI: https://doi.org/10.24321/0019.5138.202503

Discussion

In this section, the key points are defined which are found out from this research.

- The hybridization of LRP and ET algorithm for feature selection enhances the classification rate in the proposed frameworks.
- The development of the multi-agent framework helps to enhance the robustness and adaptability required to stay ahead within the system for COVID detection.

Conclusion and Future Work

In this study, we have designed two multi-agent frameworks for COVID-19 detection using the ML and DL algorithms. Initially, in both frameworks, pre-processing of the dataset and appropriate feature selection using the LBP-ET method are done to prepare it for the ML/DL model. Further, in the first framework, three ML algorithms are used to design an ensemble learning approach, and hypertuning is done to fine-tune the parameters of the ML algorithms using the grid search method. In the second framework, a lightweight ResNet algorithm is designed to reduce the computational complexity. Next, in both frameworks, a user interface is designed with the help of reinforcement learning. In the first framework, Q-Learning, whereas in the second interface, the PPO algorithm is used. The simulation evaluation shows that the MA-DL framework outperforms MA-ML due to effectively handling the dataset. In the future, real-time datasets will be taken into consideration. Followed by exploring the other ML and DL algorithms for designing the COVID-19 detection frameworks.

Conflict of Interest: None

Source of Funding: None

Generative AI and AI-Assisted Technologies in the Writing Process: None

References

1. Ghaderzadeh, M. and Asadi, F., 2021. Deep learning in the detection and diagnosis of COVID-19 using radiology modalities: a systematic review. Journal of healthcare engineering, 2021(1), p.6677314.
2. Islam, M.M., Karray, F., Alhajj, R. and Zeng, J., 2021. A review on deep learning techniques for the diagnosis of novel coronavirus (COVID-19). Ieee Access, 9, pp.30551-30572.
3. Alyasseri, Z.A.A., Al-Betar, M.A., Doush, I.A., Awadallah, M.A., Abasi, A.K., Makhadmeh, S.N., Alomari, O.A., Abdulkareem, K.H., Adam, A., Damasevicius, R. and Mohammed, M.A., 2022. Review on COVID-19 diagnosis models based on machine learning and deep learning approaches. Expert systems, 39(3), p.e12759.
4. Mondal, M.R.H., Bharati, S. and Podder, P., 2021. Diagnosis of COVID-19 using machine learning and deep learning: a review. Current Medical Imaging, 17(12), pp.1403-1418.
5. Abdulkareem, K.H., Mohammed, M.A., Salim, A., Arif, M., Geman, O., Gupta, D. and Khanna, A., 2021. Realizing an effective COVID-19 diagnosis system based on machine learning and IOT in smart hospital environment. IEEE Internet of things journal, 8(21), pp.15919-15928.
6. Khanday, A.M.U.D., Rabani, S.T., Khan, Q.R., Rouf, N. and Mohi Ud Din, M., 2020. Machine learning based approaches for detecting COVID-19 using clinical text data. International Journal of Information Technology, 12, pp.731-739.
7. Arista, A., 2021. Comparison Decision Tree and Logistic Regression Machine Learning Classification Algorithms to determine Covid-19. Sinkron: jurnal dan penelitianteknikinformatika, 6(1), pp.59-65.
8. Nopour, R., Shanbehzadeh, M. and Kazemi-Arpanahi, H., 2022. Using logistic regression to develop a diagnostic model for COVID-19: A single-center study. Journal of education and health promotion, 11(1), p.153.
9. Charan, G.V.S. and Kumar, N.S., 2022. Analysis and Comparison for Innovative Prediction Technique of COVID-19 using Logistic Regression algorithm over Support Vector Machine Algorithm with Improved Accuracy. Journal of Pharmaceutical Negative Results, pp.461-469.
10. Shanbehzadeh, M., Nopour, R. and Kazemi-Arpanahi, H., 2022. Developing an artificial neural network for detecting COVID-19 disease. Journal of education and health promotion, 11(1), p.2.
11. Keles, A., Keles, M.B. and Keles, A., 2024. COV19-CNNet and COV19-ResNet: diagnostic inference Engines for early detection of COVID-19. Cognitive Computation, 16(4), pp.1612-1622.
12. Akter, S., Shamrat, F.J.M., Chakraborty, S., Karim, A. and Azam, S., 2021. COVID-19 detection using deep learning algorithm on chest X-ray images. Biology, 10(11), p.1174.
13. Mohammedqasim, H. and Ata, O., 2022. Real-time data of COVID-19 detection with IoT sensor tracking using artificial neural network. Computers and Electrical Engineering, 100, p.107971.
14. Aslan, M.F., Sabanci, K. and Ropelewska, E., 2022. A new approach to COVID-19 detection: An ANN proposal optimized through tree-seed algorithm. Symmetry, 14(7), p.1310.
15. Alakus, T.B. and Turkoglu, I., 2020. Comparison of deep learning approaches to predict COVID-19 infection. Chaos, Solitons & Fractals, 140, p.110120.
16. Aminizadeh, S., & Lee, J. H. (2024). Opportunities and challenges of artificial intelligence and distributed systems to improve the quality of healthcare service. Artificial Intelligence in Medicine, 149, 102779.

17. Alfian, G., Syafrudin, M., Fahrurrozi, I., Fitriyani, N.L., Atmaji, F.T.D., Widodo, T., Bahiyah, N., Benes, F. and Rhee, J., 2022. Predicting breast cancer from risk factors using SVM and extra-trees-based feature selection method. Computers, 11(9), p.136.
18. Abayomi-Alli, O.O., Damaševičius, R., Maskeliūnas, R. and Misra, S., 2022. An ensemble learning model for COVID-19 detection from blood test samples. Sensors, 22(6), p.2224.
19. Wibowo, F.W., 2021, February. Prediction modelling of COVID-19 outbreak in Indonesia using a logistic regression model. In Journal of Physics: Conference Series (Vol. 1803, No. 1, p. 012015). IOP Publishing.
20. Kaur, T. and Gandhi, T.K., 2022. Classifier fusion for detection of COVID-19 from CT scans. Circuits, systems, and signal processing, 41(6), pp.3397-3414.
21. Miralles-Pechuán, L., Jiménez, F., Ponce, H. and Martínez-Villaseñor, L., 2020, October. A methodology based on deep q-learning/genetic algorithms for optimizing covid-19 pandemic government actions. In Proceedings of the 29th ACM international conference on information & knowledge management (pp. 1135-1144).
22. Jafar Zaidi, S.A., Chatterjee, I. and Brahim Belhaouari, S., 2022. COVID-19 Tweets Classification during Lockdown Period Using Machine Learning Classifiers. Applied Computational Intelligence and Soft Computing, 2022(1), p.1209172.

ISSN: 0019-5138
DOI: https://doi.org/10.24321/0019.5138.202503