Stochastic Modelling and Computational Sciences

UTILIZING PARTICLE SWARM OPTIMIZATION FOR HYPERPARAMETER TUNING OF MACHINE LEARNING MODELS ON BIG COVID-19 DATA ANALYTICS

Hend S. Salem[1], Mohamed A. Mead[2] and Ghada S. El-Taweel[3]

[1]Department of Computer Sciences Suez Canal University, Ismailia, Egypt
Email: hend shaaban@ci.suez.edu.eg

[2]Department of Computer Sciences Suez Canal University, Ismailia, Egypt
Email: mohamedmead@ci.suez.edu.eg

[3]Department of Computer Sciences Suez Canal University, Ismailia, Egypt
Email: ghada@ci.suez.edu.eg

ABSTRACT

Analyzing big data, especially medical data, helps to provide good health care to patients and face the risks of death. The COVID-19 pandemic has had a significant impact on public health worldwide, emphasizing the need for effective risk prediction models. Machine learning (ML) techniques have shown promise in analyzing complex data patterns and predicting disease outcomes. The accuracy of these techniques is greatly affected by changing their parameters. Hyperparameter opti- mization plays a crucial role in improving model performance. In this study, the Particle Swarm Optimization (PSO) algorithm is employed to search the hyperparameter space efficiently and enhance the machine learning models' predictive abilities by selecting the optimum hyperparameters that can give the best accuracy. The research utilizes a dataset consisting of various clinical and epidemiological features associated with COVID-19 cases. Multiple ML models, such as Decision Trees, Random Forests, Support Vector Machines, and Neural Networks, are employed to capture the intricate relationships within the data.To evaluate the predictive performance of the models, the accuracy metric was employed. The experimental results demonstrate the effectiveness of the proposed approach in predicting COVID- 19 risk. The optimized ML models exhibit superior performance compared to their baseline counterparts, achieving higher results.

Keywords—Big COVID-19 data; Machine Learning; Hyperpa- rameter Optimization; Particle Swarm Optimization; Computational Intelligence

A) INTRODUCTION

The Analysing big data, particularly medical data, assists in providing appropriate health care to patients and dealing with the risks of death. COVID-19, also known as coronavirus disease, is a respiratory infection caused by the SARS-CoV- 2 virus. The sickness was discovered in December 2019 in Wuhan, China, and has since spread globally, resulting in a pandemic that has harmed millions of people worldwide, affected public health, the economy, and daily life. Since the outbreak, COVID-19 has sparked the curiosity of academics from a range of sectors, including medical research, epidemi- ology, public health, and computer science. These scientists collaborated to gain a comprehensive understanding of the virus, its transmission patterns, and viable methods for curbing its spread. In this undertaking, the significance of machine learning and deep learning technologies has been demon- strated. By examining vast datasets associated to COVID-19, these technologies contributed in the development of diagnos- tic tools and predictive models for tracking the evolution and severity of the disease, as well as estimating the number of recovered and deceased cases. Machine learning (ML) is a branch of artificial intelli- gence (AI) with an emphasis on developing systems that can learn and improve on their own without being explicitly pro- grammed [1]. Machine learning algorithms allow computers to recognise patterns in data and make predictions or judgements based on that data. Some uses of AI and machine learning technologies to battle the COVID-19 pandemic are:

- Early detection: AI and machine learning algorithms have been used to develop tools for early detection of COVID- 19, based on symptoms, medical history, and other factors [2].
- Diagnosis: Machine and Deep learning algorithms have been used to analyze chest CT scans and X-ray images to accurately diagnose COVID-19 with a high degree of sensitivity and specificity [3].
- Epidemiological modeling: Machine learning algorithms have been used to develop epidemiological models to predict the infection and spread of COVID-19 [4].
- Contact tracing: AI and machine learning algorithms have been used to develop contact tracing tools that can quickly identify and isolate people who have been in close contact with COVID-19 patients [5].

Overall, the use of machine learning, deep learning, and AI in COVID-19 prevention and control has been crucial in speeding up the development of diagnostic tools, identifying potential treatments, and developing epidemiological models for predicting the spread of the virus.

Choosing and setting up appropriate machine and deep learning algorithms can be a challenging task when it comes to the prediction process. The success of these models largely depends on selecting the best configurations [6]. As a result, it is essential to accurately identify the optimal hyperparameters for the model prior to the learning process. Hyperparameter optimization is the process of selecting the optimal hyper- parameters for a machine learning model. Hyperparameters are configuration settings that are not learned during training, but rather set before the training process begins. Examples of hyperparameters include learning rate, batch size, regulariza- tion parameters, and the number of hidden layers in a neural network.

Selecting the appropriate hyperparameters can greatly im- pact the performance of a model. A hyperparameter that is set too low may result in underfitting, while a hyperparameter that is set too high may result in overfitting. Therefore, it is important to find the optimal values of hyperparameters that balance model complexity and performance. Hyperparameters optimization process offers numerous advantages [7] including enhanced performance, faster convergence, improved model generalization, increased robustness, reproducibility, and cost-cutting. Overall, hyperparameter optimisation is an essential aspect in developing machine learning models. It is vital to strike the proper balance between model complexity and performance, as well as to optimise the model's ability to generalise to new and previously unknown data. There are various ways for op- timising hyperparameters, including Grid Search [8], Random Search [9], Bayesian Optimization [10], Genetic Algorithm [8], Gradient-Based Optimization [11], Swarm intelligence [12][13], and Ensemble-based Methods [14].

In general, the selection of hyperparameter optimization method relies on the specific problem and dataset being used. To improve the possibilities of identifying the ideal hyperparameters, it is often recommended to employ a com- bination of different methodologies. Furthermore, adequate validation procedures must be used to ensure that the model does not overfit to the validation data during hyperparameter optimisation.

The following are the key findings of this study:

1) Predicting the COVID-19 risk using well-fine machine learning models.
2) Employing Particle Swarm Optimization(PSO) for model hyperparameters optimization.
3) Training the machine learning models with the optimal hyperparameters obtained from the optimization process.
4) Evaluate the performance and compare the results of the optimized models using PSO to their baseline counter- parts.

The remainder of this paper is organized as follows. Section II provides an overview of related work on machine and deep learning models used with COVID-19 prediction. Section III presents the materials and methods, providing a detailed description of the research approach and the dataset. Section IV outlines the methodology employed in the study. Section V describes the experimental setup and evaluation method as well as presents the findings obtained from the study and offers an in-depth analysis and interpretation of the results. Finally, section VI concludes the paper with a summary of our findings and directions for future research.

B) LITERATURE REVIEW

Using machine and deep learning for COVID-19 has been a rapidly growing area of research since the start of the pandemic. These techniques have been applied to a wide range of tasks, including diagnosing COVID-19, predicting patient outcomes, analyzing epidemiological trends, and developing treatments. In terms of COVID-19 diagnosis, several studies have investigated the use of machine learning models to distinguish COVID-19 from other respiratory illnesses based on medical imaging and other clinical data. The paper [15] proposed a study that uses a deep learning algorithm to analyze chest CT scans and accurately diagnose COVID-19 with a high degree of sensitivity and specificity. Machine learning has also been used to predict patient outcomes and identify risk factors for severe illness. The study in [16] used machine learning to develop a predictive model for COVID-19 mortality based on patient demographic data, medical history, and symptoms. The authors in [17] utilized a deep learning model to automatically detect the abnormalities in chest CT scans of COVID-19 patients and evaluate its quantitative performance in compar- ison to that of radiology residents. In [3], a deep learning approach to detect COVID-19 coronavirus using X-ray images was proposed. The authors developed a model based on a pre-trained convolutional neural network (CNN) architecture, which was fine-tuned on a dataset of X-ray images of COVID-19 patients, healthy individuals, and those with other lung diseases. The performance of the model was evaluated in terms of sensitivity, specificity, and accuracy. The results showed that the proposed approach achieved high accuracy and sensitivity, demonstrating the potential of deep learning in the diagnosis of COVID-19. The paper [18] introduced a model for predict- ing drug-target interactions (DTIs) that exploits the inherent structural properties of proteins and drugs. The research in [19] used a diversity of the following machine learning algorithms: Support Vector Machine (SVM), Naive Bayes, K-Nearest Neighbour (KNN), Logistic Regression, and Decision Tree, to evaluate a combined dataset of 2500 cases with features of fever, runny nose, body pain, difficulty in breathing, nasal congestion, and sore throat. The authers in [20] presented an artificial intelligence algorithm called DeepCOVID-XR, which is designed to detect COVID-19 on chest radiographs. The algorithm was trained and tested on a large clinical dataset in the United States and achieved high accuracy in detecting COVID-19 cases. The study suggests that DeepCOVID-XR can be used as a screening tool to help clinicians prioritize patients for further testing and treatment. A tailored deep convolutional neural network (CNN) called COVID-Net CT to detect COVID-19 cases from chest CT images was proposed in [21]. The proposed CNN model is trained and evaluated on a large dataset of chest CT images containing COVID-19 and non-COVID-19 cases. The results show that COVID-Net CT achieved high accuracy and outperformed other state-of-the-art models in detecting COVID-19 cases from chest CT images. The paper [4] developed various supervised machine learning models include decision tree, logistic regression, naive Bayes, support vector machine, and artificial neural network to predict the infection of COVID-19 cases in Mexico. The authors used the epidemiology labeled dataset of 263,007 instances and 41 features. After several steps of preprocessing and prediction they ended up with 94.99% as the highest accuracy of decision tree model, 93.34% as the highest sensitivity of support vector machine model, and 94.30% as the highest specificity. Authors in [22] proposed a deep learning based method to predict the risk of the ongoing epidemic. They applied deep learning along with reinforcement learning. Recurrent Neural Network and the Modified Long Short-Term Memory(M-LSTM) were used to construct the prediction model. Moreover, their prediction results were optimized using a deep reinforcement learning algorithm which determined the best activation function for the M-LSTM. The results of this study after optimization proved that the proposed model outperformed other related studies. The work in [23] presented a study to predict the number of newly COVID-19 cases, the number of deaths, and the number of recoveries in the next 10 days by using linear regression (LR), support vector machine (SVM), exponential smoothing (ES), and least absolute shrinkage and selection operator (LASSO). The results showed that the ES had the best performance and the SVM had the worst in the three types of predictions.

C) MATERIALS AND METHODS

a) DATASET

In this study, the ”COVID-19 Dataset” is obtained from Kaggle [24]. This dataset includes 21 distinct features and 1,048,576 distinct cases. The dataset’s features columns are described in Table I. The data types of the features numeric and date.

Table 1: Covid-19 Dataset Descritption

Feature	Type	Description
USMER	Numerical	the patient treated medical units of 1st, 2nd or 3rd level
MEDICAL UNIT	Numerical	type of institution that provided the care
SEX	Numerical	1 - female. 2 - male
PATIENT TYPE	Numerical	type of care the patient received
DATE DIED	Date	The date of death, or 9999-99-99 otherwise
INTUBED	Numerical	whether the patient was connected to the ventilator
PNEUMONIA	Numerical	whether the patient already have air sacs inflammation or not
AGE	Numerical	Age of the patient.
PREGNANT	Numerical	whether the patient is pregnant or not
DIABETES	Numerical	whether the patient has diabetes or not
COPD	Numerical	has Chronic obstructive pulmonary disease or not
ASTHMA	Numerical	whether the patient has asthma or not
INMSUPR	Numerical	whether the patient is immunosuppressed or not
HIPERTENSION	Numerical	whether the patient has hypertension or not
OTHER DISEASE	Numerical	whether the patient has other disease or not
CARDIOVASCULAR	Numerical	whether the patient has heart or blood vessels related disease
OBESITY	Numerical	whether the patient is obese or not
RENAL CHRONIC	Numerical	whether the patient has chronic renal disease or not
TOBACCO	Numerical	whether the patient is a tobacco user
CLASIFFICATION FINAL	Numerical	Covid test results.
ICU	Numerical	whether the patient had been intensive care unit

b) HYPERPARAMETERS OF MACHINE LEARNING MODELS

Selecting and configuring the machine and deep learning algorithms is a challenging task in the prediction process. The performance of numerous models is dependent on picking the appropriate model configurations [25]. Consequently, it is crit- ical to successfully define the model’s ideal hyperparameters prior to model learning. Hyperparameters are the parameters of the machine learning algorithms that shape the structure of the machine learning models and they are set by the user before the learning phase [26]. Some hyperparameters are used to configure the model, such as the learning rate used to train a neural network or the penalty parameter in the SVM model, while others, such as the activation function in a neural network and the kernel type in the SVM model, are used to select the algorithm that minimises the cost function. The hyperparameters are classified as discrete, continuous, or categorical hyperparameters [27]. Table II provides a list of general hyperparameters that are used to build machine learn- ing models. The process of determining the ideal combination of hyperparameter settings that maximises the overall model’s performance and accuracy is known as hyperparameter tuning [28]. Because of its direct influence on the accuracy of the analysis and prediction tasks, the hyperparameter tuning pro- cess is regarded as a game changer for any machine learning system [42] 43][44][45]

Table 2: Hyperparameters For Machine Learning Models

Machine Learnign Models	Hyperparameter
Boosted Random Forest	n estimators max depthcriterion min samples splitmin samples leafmax features
Decision Tree	Criterion max depth min samples splitmin samples leaf max features
SVM	C kernel
KNN	N neighbors
MLPANN	number of hidden layers, loss, optimizer, activation, learning rate, dropout rate, epochs, batch size, early stop patience

c) PARTICLE SWARM OPTIMIZATION (PSO)

PSO algorithm is a type of Evolutionary Algorithms (EA) that are inspired by nature to search and find the best solutions as an optimization problem. The PSO operates on spreading a swarm of interactive particles (agents) in the problem' search space. Each particle expresses a candidate solution for the problem. Each particle moves in the search space by tracking its position and velocity which are updated each state to keep track the optimal particles[29]. The main steps of the PSO algorithm is explained in Algorithm 1.

Because of the ease of use, high performance, and the ability to find the optimal solution, PSO technique is engaged in many analytics and prediction applications. For example, health- care systems [30][31][32], transportation systems [33][34][35], wastewater treatment [36][37], and Education [38][39][40].

Algorithm 1 The Standard algorithm of Particle Swarm optimization (PSO)

for i 1 **to** N_p **do**

Initialize the particle's position $P_i(0) = U(LB, UB)$, where LB and UB represent the lower and upper bounds of the search space

Initialize $pbest(i, 0)$ $P_i(0)$

Initialize $gbest$ to the minimal value of the swarm as $gbest \leftarrow argmin(f[P_i(0)])$ Initialize velocity $V_i \sim U(-|UB - LB|, |UB - LB|)$

REPEAT

foreach *particle i 1* **to** N_p **do**

Pick random numbers: r_1, r_2 $U(0, 1)$. Update velocity $V_i(t+1) = \omega V_i(t) + c_1 r_1(pbest(i, t) \quad P_i(t)) + c_1 r_1(gbest(t)\ P_i(t))$ Update position $P_i(t + 1) = P_i(t) + V_i(t + 1)$ **if** $f[P_i(t)] < f[pbest(i, t)]$ **then**

Set best position as $pbest(i, t) \leftarrow P_i(t)$ **if** $f[P_i(t)] < f[gbest(t)]$ **then** $gbest(t) \leftarrow P_i(t)$;

Move to next iteration t $t + 1$

until *termination criteria is met*;

Output: $gbest(t)$ that holds the best found solution

d) METHODOLOGY

A. The Proposed Approach

The overall prediction framework is depicted in Figure 1. The prediction procedure is divided into five stages: Covid-19 data collecting, preprocessing, prediction, performance evaluation, and visualisation of outcomes. The data collection phase involves gathering the Novel COVID-2019 dataset, which is then utilized to assess the machine learning models. Preprocessing phase involves changing, purifying, updating, and preparing the dataset for machine learning model training. Prediction phase includes the prediction of the COVID-19 risk, multiple PSO-based refined machine learning models were used on the training dataset.

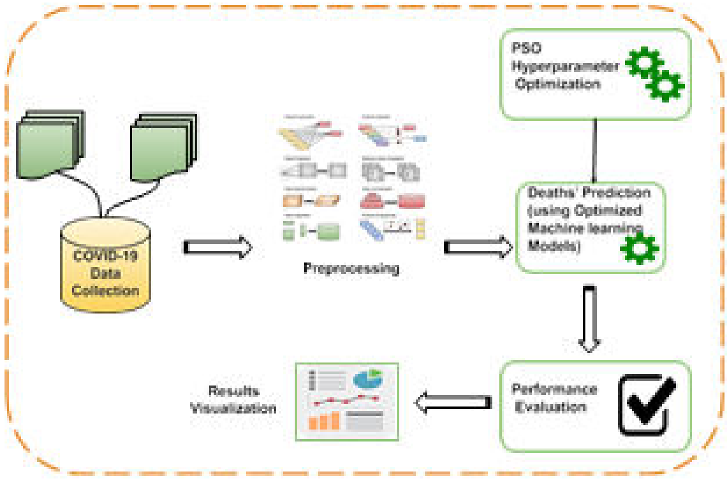


Figure 1: . Schematic architectural diagram of the proposed COVID-19 prediction framework

During the performance evaluation phase, the results are scrutinized and assessed to ascertain their effectiveness. Result visualisation phase entails converting the prediction outcomes into a graphical represen- tation. Figure 2 shows the detailed steps of the prediction process [41][42].

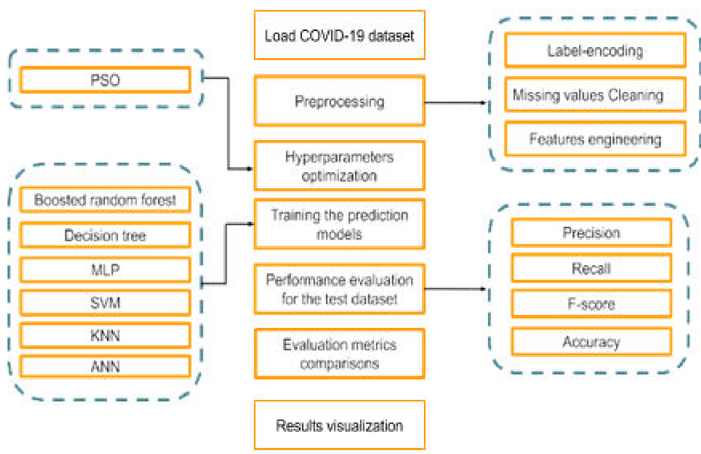


Figure 1: Block diagram of COVID-19 prediction process

B. Data Preprocessing

The COVID-19 Dataset had been preprocessed by three tasks as follows:

1. Label-encoding: One Hot Encoding was used to en- code each column with a numeric value(0 or 1) except columns 'MEDICAL UNIT', 'AGE', and 'CLASIFFI- CATION FINAL' [43][44].
2. Cleaning the missing values: the rows containing values 97 and 99 were missing, so the data was cleaned to preserve rows that contain 1 and 2 values only. Feature engineering: columns 'DATE DIED', 'IN- TUBED', and 'ICU' were summed together resulting a label column 'AT RISK' that expresses whether the patient is at risk from COVID-19 or not [45][46][47].
3. Handling imbalancing classes by using Synthetic Mi- nority Over-sampling Technique (SMOTE): SMOTE is a statistical technique used to increase the number of cases in a dataset in a balanced way [41]. It works by creating synthetic samples from the minority class instead of creating copies. This is achieved by randomly selecting a point from the minority class and computing the k- nearest neighbors for this point. The synthetic

points are then created by choosing one of the k-nearest neighbors and forming a linear combination with the original point. This method helps in balancing the dataset by augmenting the minority class, which improves the performance of the classification algorithms. Unlike simple oversampling techniques that replicate existing samples, SMOTE gen- erates new instances that are plausible and within the feature space of the minority class. This leads to a more diverse and representative sample of the minority class, reducing the likelihood of overfitting that is common with simple oversampling. By synthesizing new examples, SMOTE can improve the decision boundary derived by the classifier, making it more robust and accurate in distinguishing between the classes. Figure 3 portraits the affect of SMOTE on the imbalancing dataset. In the ”Original Dataset” chart in Figure 3 (a), we see a significant imbalance between the two classes, with the majority class (label 0) having 937,891 instances compared to 87,261 instances in the minority class (label 1). This kind of imbalance can skew the performance of machine learning models, often leading to a bias towards the majority class. The ”Balanced Dataset with SMOTE” chart in Figure 3 (b) shows the class distribution after applying SMOTE. Both classes are now equal with 937,891 instances each. By synthesizing new samples in the minority class, SMOTE has created a balanced dataset, which can potentially improve the performance of a classifier. With this balance, a model can better learn the characteristics of both classes, leading to improved generalization and a more accurate prediction on unseen data.

C. Hyperparameters Optimization of Machine learning Mod- els using PSO

In this study, the PSO algorithm is utilized to optimize the hyperparameters of the machine learning models. Recent studies have demonstrated that swarm intelligence, particularly PSO, is capable of achieving highly effective results in hyper- parameter optimization tasks, particularly when dealing with large-scale datasets. The findings of this study also provide evidence of the efficacy of PSO in optimizing hyperparameters for improved performance. The procedure of hyperparameter optimization process by using the PSO is mentioned in Algo- rithm 2 [48][49].

Algorithm 2 PSO for Machine Learning HyperparameterOptimization.

Input:

Number of Particles (N) Maximum Iterations ($MaxIter$)Inertia Weight (ω)

Learning Factors (c_1, c_2)

Hyperparameter Boundaries ($ParamLimits$)

Result:

Global Best set of Hyperparameters of a given ML Model($BestGlobal$)

for $i \leftarrow 1$ **to** N **do**

$particle[i].pos \leftarrow random\_pos\_within(ParamLimits) particle[i].vel \leftarrow random\_init\_velocity()$

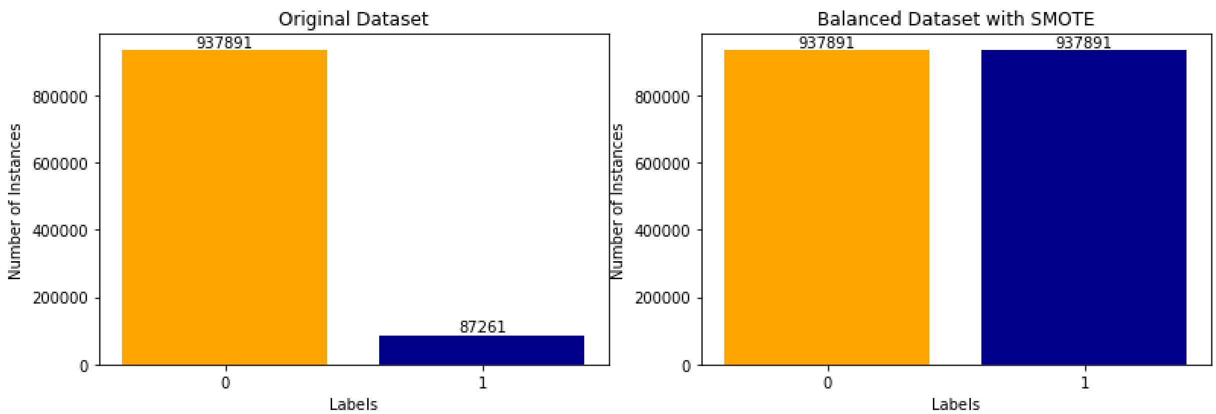
$particle[i].pbest \leftarrow particle[i].pos$

$particle[i].pbest\_score \leftarrow evaluate(particle[i].pos)$

$BestGlobal \leftarrow$ Best scoring particle among $particle[i].best$
$BestGlobalScore \leftarrow$ Score of $BestGlobal$
for $iter = 1$ *to* $MaxIter$ **do**
  foreach *particle* **do**
    foreach *dimension* d **do**
      $rand1, rand2 \leftarrow$ Uniform random numbers in $[0, 1]$
      $personal \leftarrow c_1 \times rand1 \times (particle.best[d] - particle.pos[d])$
      $social \leftarrow c_2 \times rand2 \times (BestGlobal[d] - particle.pos[dimension])$
      $particle.vel[d] \leftarrow \omega \times particle.vel[d] + personal + social$
      $particle.pos$ Update position with $particle.vel$
      Enforce bounds on $particle.pos$ within $ParamLimits$
      $current\ score \leftarrow evaluate(particle.pos)$
      if $current\ score > particle.best\ score$ **then**
        $particle.best \leftarrow particle.pos$ $particle.best\ score \leftarrow current\ score$
      if $current\ score > BestGlobalScore$ **then**
        $BestGlobal \leftarrow particle.pos$
        $BestGlobalScore \leftarrow current\ score$

return $BestGlobal$

The objective functions of machine learning models play a crucial role in optimizing the performance and accuracy of these models. These functions serve as the guiding principle for the learning algorithm to minimize or maximize a specific metric, ultimately leading to the best possible model fit.



(a) (b)

Fig. 3. Balancing the COVID-19 Dataset by SMOTE

The choice of objective function is critical in guiding the learning process and shaping the behavior of machine learning models. In this study, a supervised learning is employed, where models are trained using labeled data, the most common objective function is the loss function. The loss function quantifies the discrepancy between the predicted outputs and the actual labels. The goal is to minimize this discrepancy, which is achieved through the PSO algorithm. The following subsections provide a description of the objective functions and hyperparameters that need to be tuned for the six machine learning models [50][51].

1) k-nearest neighbor (knn): for each test sample in knn, the predicted class is determined by identifying the majority class among its k-nearest neighbors within the training set. A class y is denoted by:

$$L_{\mathrm{CE}} = -\frac{1}{N}\sum_{i,j}^{m_j y_{ij}} \log(p_{ij}) \qquad (2)$$

2) *Boosted Random Forest :* Random Forest is an ensemble learning method that combines multiple decision trees to make predictions. Unlike some other algorithms, Random Forest does not have a specific objective function that is directly optimized during training. Instead, it uses a combination of techniques, including bootstrapping and random feature selection, to build a diverse set of decision trees and aggregate their predictions.

3) Artificial Neural Network (ANN): ANN classifier de- pends on the specific type of objective functions being used. There are several common objective functions for different types of ANN classifiers.

Cross-Entropy Loss:

Let yi be the true class label for the i-th sample (binary or one-hot encoded) pi be the predicted class probability for the i-th sample, then the cross-entropy loss is given

by:

$$L_{\mathrm{CE}} = -\sum_i y_i \log(p_i) \qquad (3)$$

Hinge Loss:

Let y_i be the true class label (-1 or 1) for the i-th sample f_i be the predicted output (before applying the sign function) for the i-th sample, then the hinge loss is given by:

$$L_{\mathrm{hinge}} = \frac{1}{N}\sum_i \max(0, 1\ y\ N \cdot f_i) \qquad (4)$$

Mean Squared Error (MSE) Loss:

Let yi be the true class label (0 or 1) for the i-th sample pi be the predicted class probability (between 0 and 1) for the i-th sample, then the mean squared error loss is given by

$$L_{\mathrm{MSE}} = \frac{1}{} \ {}_i(yN — p_i) \qquad (5)$$

4) Multilayer Perceptron Neural Network (MLP): The objective function of an MLP classifier using cross-entropy loss is given by:

$$L_{\mathrm{CE}} = -\frac{1}{N}\sum_{i}\sum_{j} y \log(p_{ij} \qquad (6)$$

where y_i is the true class label (binary or one-hot encoded)

for the i-th sample and p_i be the predicted class probabilities (output) for the i-th sample. The hyperparamters that need to be tune in MLP are: **hidden_layer_sizes** that determines the quantity of neurons present in each hidden layer of the MLP, **activation** which decides the activation function applied in the hidden layers, **solver** that designates the optimization algorithm utilized for training the MLP, **alpha** that commonly referred to as weight decay, which serves the purpose of mitigating overfitting, **learning_rate** establishes the learning rate employed for weight updates during training, **batch_size** determines the quantity of samples utilized in each training batch, and **max_iter** that sets the maximum number of itera- tions (epochs) for training the MLP.

4. RESULTS AND DISCUSSION

A. Experiment Settings

Boosted Random Forest, Decision Tree, SVM, MLP, KNN, and ANN were used to validate the suggested methodol- ogy. The PSO algorithm was used to tune the classifier's hyperparameters. The code of the classifiers was written in Python 3 and run on a laptop with Ubuntu 20.04.1 LTS, an Intel® CoreTM i5-8250U CPU @ 1.60 GHz 8, and 8 GB of main memory. The code relied on several package and library dependencies, including Numpy, Pandas, SciPy, Datetime, Scikit Learn, sklearn, Pyswarm, and Matplotlib. In these implementations the dataset was splited into 80% training and 20% testing and validation.

B. Performance evaluation

In order to assess the effectiveness of the machine learning models, the evaluation was conducted using the accuracy metric, as defined in equation 7.

Accuracy: is the ratio of the total correct predictions (TP + TN + FP + FN) by the predictor or classifier to the total data points (TP + TN) of a dataset. Equation 7 is used to calculate the accuracy metric.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (7)$$

C. Results

For figuring out COVID-19 risk, the PSO algorithm was used to optimise the parameters of six different machine learn- ing models: Boosted Random Forest, Decision Tree, SVM, KNN, MLP, and Artificial Neural Networks. The models were trained on a COVID-19 dataset including attributes and clinical factors. Table III summarizes the hyperparameters that were tuned using PSO in this study. To evaluate the performance of the PSO-optimized models, they were compared to baseline models that were trained without optimization. The accuracy was used as the evaluation metric. The results showed that the PSO-optimized models outperformed the baseline models in the basis of the accuracy metric.

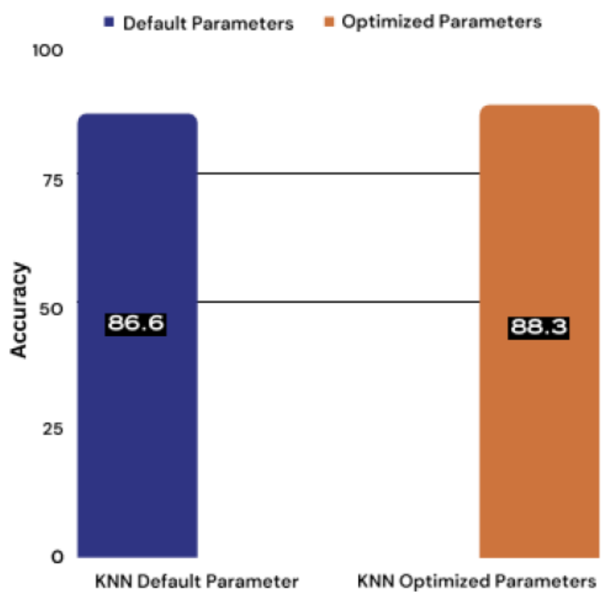
Table iii : Summary of the hyperparameter values of the machine learning models utilised in the experimental work

Model	Hyperparameter	Value
Boosted Random Forest	max depth max featuresn estimator	9 7 67
Decision Tree	max depth min samples split	7 7
SVM	C gamma	79.61 0.155
KNN	K P	32 1

MLP	hidden layers number of neuronstanh learning rate	2 2 1 0.0776
ANN	hidden layer sizes alpha learning rate	(10, 25) 0.00029 0.00226

The accuracy of the optimized models ranged from 88.26% to 92.20%, while the baseline models achieved an accuracy of 86% to 89.52%.

To visualize these results, we created bar charts comparing the accuracy of the PSO-optimized models to the baseline models. The charts showed that for all models, the optimized versions had higher accuracy than the baseline versions. As an illustration, in Figure 6, the comparison chart of Decision Tree models demonstrated that the PSO-optimized model attained an accuracy level of 92.20%, whereas the baseline model reached an accuracy of 89.52%. Overall, these results suggest that the PSO algorithm can ef- fectively optimize the parameters of machine learning models to improve their performance in predicting COVID-19 risk. These findings have important implications for developing accurate and effective public health policies and interventions during the ongoing global efforts to combat the COVID-19 pandemic.



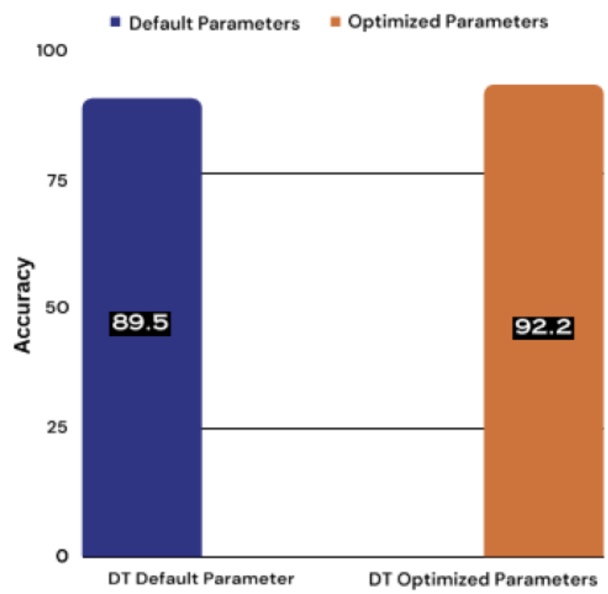
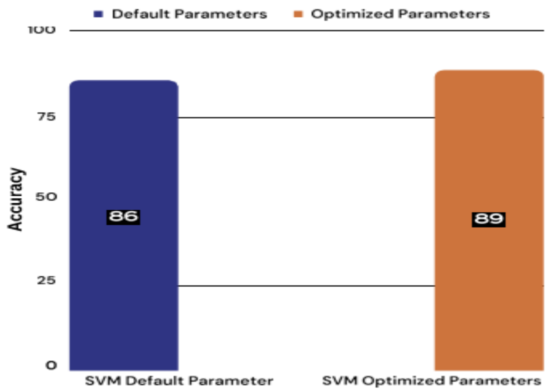


Fig. 4. Accuracy of optimized KNN and its baseline counterpart

Fig. 5. Accuracy of optimized Decision Tree and its baseline counterpart



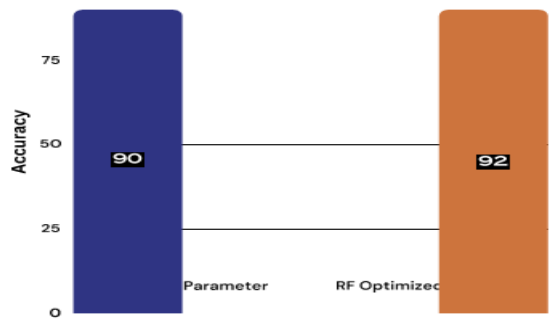
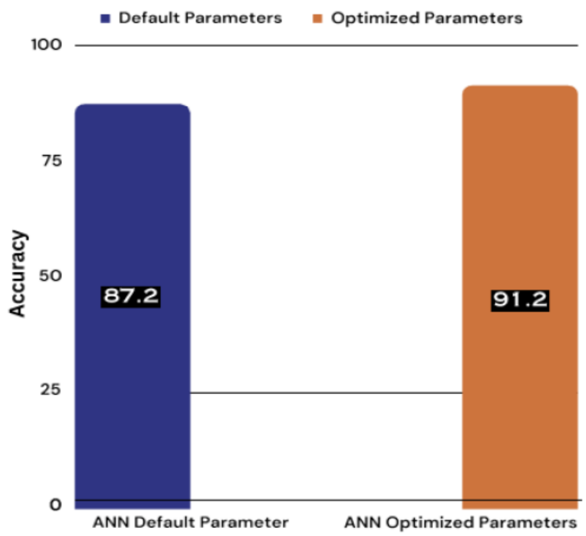


Fig. 6. Accuracy of optimized SVM and its baseline counterpart

Fig. 7. Accuracy of optimized Random Forest and its baseline counterpar



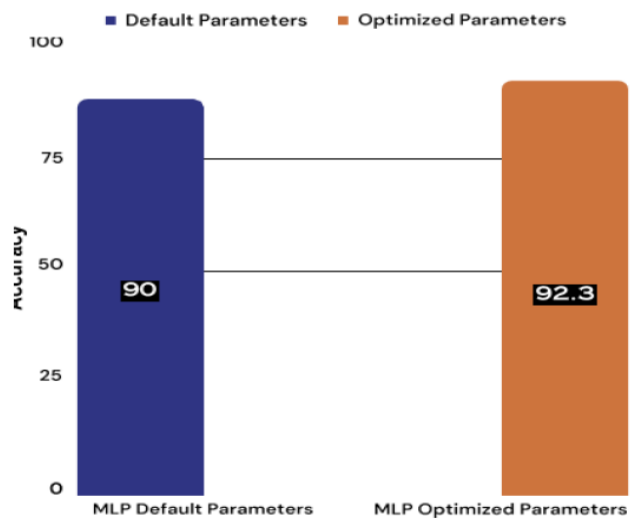


Fig. 8. Accuracy of optimized ANN and its baseline counterpart

Fig. 9. Accuracy of optimized MLP and its baseline counterpart

5. CONCLUSION AND FUTURE WORK

In conclusion, the PSO algorithm was used to optimize the hyperparameters of six machine learning models, including Random Forest, Decision Tree, SVM, KNN, MLP and Artificial Neural Networks, to achieve the best possible accuracy in analyzing and predicting COVID-19 risk. The results showed that the optimized models outperformed the PSO algorithm can effectively improve the performance of machine learning models in predicting COVID-19 risk, which could potentially be useful in developing effective public health policies and interventions. Overall, the findings of this study highlight the potential of using optimized machine learning models to accurately predict COVID-19 risk and provide valuable insights into the ongoing global efforts to combat this pandemic. Future work in this area could focus on several aspects to further improve the performance of the PSO-optimized machine learning models for predicting COVID-19 risk. Therefore, our future work will concentrate on incorpo- rating additional relevant variables, such as healthcare system capacity, to improve the accuracy of the models.This would require extensive data collection and preprocessing efforts, as well as careful selection of relevant variables. Furthermore, we will validate the models using independent datasets from different geographic regions and populations to ensure their generalizability and robustness.

REFERENCES

[1] B. Mahesh, “Machine learning algorithms-a review,” International Jour- nal of Science and Research (IJSR).[Internet], vol. 9, no. 1, pp. 381–386, 2020.

[2] Z. A. A. Alyasseri, M. A. Al-Betar, I. A. Doush, M. A. Awadallah, A. K. Abasi, S. N. Makhadmeh, O. A. Alomari, K. H. Abdulkareem, A. Adam,

R. Damasevicius et al., “Review on covid-19 diagnosis models based on machine learning and deep learning approaches,” Expert systems, vol. 39, no. 3, p. e12759, 2022.

[3] G. Jain, D. Mittal, D. Thakur, and M. K. Mittal, “A deep learning approach to detect covid-19 coronavirus with x-ray images,” Biocyber- netics and biomedical engineering, vol. 40, no. 4, pp. 1391–1405, 2020.

[4] L. Muhammad, E. A. Algehyne, S. S. Usman, A. Ahmad, C. Chakraborty, and I. A. Mohammed, “Supervised machine learning models for prediction of covid-19 infection using epidemiology dataset,” SN computer science, vol. 2, pp. 1–13, 2021.

[5] S.-W. Jian, H.-Y. Cheng, X.-T. Huang, and D.-P. Liu, “Contact tracing with digital assistance in taiwan’s covid-19 outbreak response,” Inter- national Journal of Infectious Diseases, vol. 101, pp. 348–352, 2020.

[6] L. Hertel, J. Collado, P. Sadowski, J. Ott, and P. Baldi, “Sherpa: Robust hyperparameter optimization for machine learning,” SoftwareX, vol. 12, p. 100591, 2020.

[7] R. Khalid and N. Javaid, “A survey on hyperparameters optimization algorithms of forecasting models in smart grid,” Sustainable Cities and Society, vol. 61, p. 102275, 2020.

[8] H. Alibrahim and S. A. Ludwig, “Hyperparameter optimization: Com- paring genetic algorithm against grid search and bayesian optimization,” in 2021 IEEE Congress on Evolutionary Computation (CEC). IEEE, 2021, pp. 1551–1559.

[9] L. Li and A. Talwalkar, “Random search and reproducibility for neural architecture search,” in Uncertainty in artificial intelligence. PMLR, 2020, pp. 367–377.

[10] R. Turner, D. Eriksson, M. McCourt, J. Kiili, E. Laaksonen, Z. Xu, and I. Guyon, “Bayesian optimization is superior to random search for machine learning hyperparameter tuning: Analysis of the black- box optimization challenge 2020,” in NeurIPS 2020 Competition and Demonstration Track. PMLR, 2021, pp. 3–26.

[11] O. Y. Bakhteev and V. V. Strijov, “Comprehensive analysis of gradient- based hyperparameter optimization algorithms,” Annals of Operations Research, vol. 289, pp. 51–65, 2020.

[12] N. Bacanin, T. Bezdan, E. Tuba, I. Strumberger, and M. Tuba, “Optimiz- ing convolutional neural network hyperparameters by enhanced swarm intelligence metaheuristics,” Algorithms, vol. 13, no. 3, p. 67, 2020.

[13] E. Byla and W. Pang, “Deepswarm: Optimising convolutional neural networks using swarm intelligence,” in Advances in Computational Intelligence Systems: Contributions Presented at the 19th UK Workshop on Computational Intelligence, September 4-6, 2019, Portsmouth, UK 19. Springer, 2020, pp. 119–130.

[14] B. Ouyang, Y. Song, Y. Li, G. Sant, and M. Bauchy, “Ebod: An ensemble-based outlier detection algorithm for noisy datasets,” Knowledge-Based Systems, vol. 231, p. 107400, 2021.

[15] C. Jin, W. Chen, Y. Cao, Z. Xu, Z. Tan, X. Zhang, L. Deng, C. Zheng, J. Zhou, H. Shi et al., “Development and evaluation of an artificial intelligence system for covid-19 diagnosis,” Nature communications, vol. 11, no. 1, p. 5088, 2020.

[16] H. Estiri, Z. H. Strasser, J. G. Klann, P. Naseri, K. B. Wagholikar, and S. N. Murphy, “Predicting covid-19 mortality with electronic medical records,” NPJ digital medicine, vol. 4, no. 1, p. 15, 2021.

[17] Q. Ni, Z. Y. Sun, L. Qi, W. Chen, Y. Yang, L. Wang, X. Zhang, L. Yang, Y. Fang, Z. Xing et al., "A deep learning approach to characterize 2019 coronavirus disease (covid-19) pneumonia in chest ct images," European radiology, vol. 30, pp. 6517–6527, 2020.

[18] H. El-Behery, A.-F. Attia, N. El-Fishawy, and H. Torkey, "Efficient machine learning model for predicting drug-target interactions with case study for covid-19," Computational Biology and Chemistry, vol. 93, p. 107536, 2021.

[19] M. Buvana and K. Muthumayil, "Prediction of covid-19 patient using supervised machine learning algorithm," Sains Malaysiana, vol. 50, no. 8, pp. 2479–2497, 2021.

[20] R. M. Wehbe, J. Sheng, S. Dutta, S. Chai, A. Dravid, S. Barutcu, Y. Wu, D. R. Cantrell, N. Xiao, B. D. Allen et al., "Deepcovid-xr: an artificial intelligence algorithm to detect covid-19 on chest radiographs trained and tested on a large us clinical data set," Radiology, vol. 299, no. 1, pp. E167–E176, 2021.

[21] L. Wang, Z. Q. Lin, and A. Wong, "Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images," Scientific reports, vol. 10, no. 1, pp. 1–12, 2020.

[22] R. L. Kumar, F. Khan, S. Din, S. S. Band, A. Mosavi, and E. Ibeke, "Recurrent neural network and reinforcement learning model for covid- 19 prediction," Frontiers in public health, vol. 9, 2021.

[23] F. Rustam, A. A. Reshi, A. Mehmood, S. Ullah, B.-W. On, W. Aslam, and G. S. Choi, "Covid-19 future forecasting using supervised machine learning models," IEEE access, vol. 8, pp. 101 489–101 499, 2020.

[24] sudalairajkumar, "The dataset "Novel Corona Virus 2019 Dataset.(2020), year = 2021, howpublished= https://www.kaggle.com/datasets/meirnizri/covid19-dataset, note=Accessed: 2023-06-10."

[25] C.-W. Tsai, C.-H. Hsia, S.-J. Yang, S.-J. Liu, and Z.-Y. Fang, "Optimiz- ing hyperparameters of deep learning in predicting bus passengers based on simulated annealing," Applied soft computing, vol. 88, p. 106068, 2020.

[26] L. Yang and A. Shami, "On hyperparameter optimization of machine learning algorithms: Theory and practice," Neurocomputing, vol. 415, pp. 295–316, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231220311693

[27] N. DeCastro-Garc´ıa, A. L. Munoz Castaneda, D. Escudero Garcia, and M. V. Carriegos, "Effect of the sampling of a dataset in the hyperparameter optimization phase over the efficiency of a machine learning algorithm," Complexity, vol. 2019, 2019.

[28] R. Elshawi, M. Maher, and S. Sakr, "Automated machine learning: State- of-the-art and open challenges," arXiv preprint arXiv:1906.02287, 2019.

[29] D. Wang, D. Tan, and L. Liu, "Particle swarm optimization algorithm: an overview," Soft computing, vol. 22, pp. 387–408, 2018.

[30] A. Al Bataineh and S. Manacek, "Mlp-pso hybrid algorithm for heart disease prediction," Journal of Personalized Medicine, vol. 12, no. 8, p. 1208, 2022.

[31] N. Nayar, S. Ahuja, and S. Jain, "Swarm intelligence and data mining: a review of literature and applications in healthcare," in Proceedings of the Third International Conference on Advanced Informatics for Computing Research, 2019, pp. 1–7.

[32] M. F. Siddiqui, A. Alam, R. Kalmatov, A. Mouna, R. Villela, A. Mi- talipova, Y. N. Mrad, S. A. A. Rahat, B. K. Magarde, W. Muhammad et al., "Leveraging healthcare system with nature-inspired computing techniques: An overview and future perspective," Nature-Inspired Intel- ligent Computing Techniques in Bioinformatics, pp. 19–42, 2022.

[33] S. C. Devi and D. Maheswari, “An embellished particle swarm optimiza- tion technique in vanet for finding optimal route (e-pso),” SN Computer Science, vol. 4, no. 2, p. 109, 2022.

[34] Y. Deng and X. Shi, “An accurate, reproducible and robust model to predict the rutting of asphalt pavement: Neural networks coupled with particle swarm optimization,” IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 11, pp. 22 063–22 072, 2022.

[35] Faiz, M., Daniel, A.K. A multi-criteria cloud selection model based on fuzzy logic technique for QoS. Int J Syst Assur Eng Manag (2022). https://doi.org/10.1007/s13198-022-01723-0

[36] M. Faiz and A. K. Daniel, "FCSM: Fuzzy Cloud Selection Model using QoS Parameters," 2021 First International Conference on Advances in Computing and Future Communication Technologies (ICACFCT), Meerut, India, 2021, pp. 42-47, doi: 10.1109/ICACFCT53978.2021.9837347.

[37] Faiz, M., & Shanker, U. (2016, March). Data synchronization in distributed client-server applications. In 2016 IEEE International Conference on Engineering and Technology (ICETECH) (pp. 611-616). IEEE.

[38] Faiz, M., Daniel, A.K. (2022). Wireless Sensor Network Based Distribution and Prediction of Water Consumption in Residential Houses Using ANN. In: Misra, R., Kesswani, N., Rajarajan, M., Veeravalli, B., Patel, A. (eds) Internet of Things and Connected Technologies. ICIoTCT 2021. Lecture Notes in Networks and Systems, vol 340. Springer, Cham. https://doi.org/10.1007/978-3-030-94507-7_11

[39] Faiz, M., & Daniel, A. K. (2022). Threats and challenges for security measures on the internet of things. Law, State and Telecommunications Review, 14(1), 71-97. [39] X. Luo et al., “Wireless sensor network and ai application for educa- tional technology course,” Journal of Sensors, vol. 2023, 2023.

[40] J. Gao and Y. Huang, “Optimization and recommendation method of distance education resources based on particle swarm optimization algorithm,” in e-Learning, e-Education, and Online Training: 8th EAI International Conference, eLEOT 2022, Harbin, China, July 9–10, 2022, Proceedings, Part II. Springer, 2023, pp. 334–346.

[41] N. Chawla, K. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “Smote: Synthetic minority over-sampling tech- nique,” ArXiv, vol. abs/1106.1813, 2002. [Online]. Available: https://api.semanticscholar.org/CorpusID:1554582

[42] Mall, P. K., et al. "A comprehensive review of deep neural networks for medical image processing: Recent developments and future opportunities. Healthcare Analytics, 4, 100216." (2023).

[43] Narayan, Vipul, et al. "Severity of Lumpy Disease detection based on Deep Learning Technique." 2023 International Conference on Disruptive Technologies (ICDT). IEEE, 2023.

[44] Narayan, Vipul, et al. "A Comprehensive Review of Various Approach for Medical Image Segmentation and Disease Prediction." Wireless Personal Communications 132.3 (2023): 1819-1848.

[45] Mall, Pawan Kumar, et al. "Rank Based Two Stage Semi-Supervised Deep Learning Model for X-Ray Images Classification: AN APPROACH TOWARD TAGGING UNLABELED MEDICAL DATASET." Journal of Scientific & Industrial Research (JSIR) 82.08 (2023): 818-830.

[46] Saxena, Aditya, et al. "Comparative Analysis Of AI Regression And Classification Models For Predicting House Damages İn Nepal: Proposed Architectures And Techniques." Journal of Pharmaceutical Negative Results (2022): 6203-6215.

[47] Kumar, Vaibhav, et al. "A Machine Learning Approach For Predicting Onset And Progression"“Towards Early Detection Of Chronic Diseases “." Journal of Pharmaceutical Negative Results (2022): 6195-6202.

[48] Chaturvedi, Pooja, A. K. Daniel, and Vipul Narayan. "A Novel Heuristic for Maximizing Lifetime of Target Coverage in Wireless Sensor Networks." Advanced Wireless Communication and Sensor Networks. Chapman and Hall/CRC 227-242.

[49] Kumar, Vaibhav, et al. "A Machine Learning Approach For Predicting Onset And Progression"“Towards Early Detection Of Chronic Diseases “." Journal of Pharmaceutical Negative Results (2022): 6195-6202.

[50] Narayan, Vipul, et al. "7 Extracting business methodology: using artificial intelligence-based method." Semantic Intelligent Computing and Applications 16 (2023): 123.

[51] Mall, Pawan Kumar, et al. "A comprehensive review of deep neural networks for medical image processing: Recent developments and future opportunities." Healthcare Analytics (2023): 100216.