PLOS ONE

RESEARCH ARTICLE

A Genetic algorithm aided hyper parameter optimization based ensemble model for respiratory disease prediction with Explainable AI

Balraj Preet Kaur[1], Harpreet Singh[2], Rahul Hans[1], Sanjeev Kumar Sharma[3], Chetna Sharma[4], Md. Mehedi Hassan[5]*

1 Department of Computer Science and Engineering, DAV University, Jalandhar, Punjab, India, **2** Department of Computer Science and Engineering, Thapar Institute of Engineering and Technology, Patiala, India, **3** Department of Computer Science and Applications, DAV University, Jalandhar, Punjab, India, **4** Chitkara University Institute of Engineering and Technology, Chitkara University, Punjab, India, **5** Computer Science and Engineering Discipline, Khulna University, Khulna, Bangladesh

* mehedihassan@ieee.org

OPEN ACCESS

Citation: Kaur BP, Singh H, Hans R, Sharma SK, Sharma C, Hassan M.M (2024) A Genetic algorithm aided hyper parameter optimization based ensemble model for respiratory disease prediction with Explainable AI. PLoS ONE 19(12): e0308015. https://doi.org/10.1371/journal.pone.0308015

Editor: Ren Qi, University of Electronic Science and Technology of China, CHINA

Received: April 3, 2024

Accepted: July 16, 2024

Published: December 2, 2024

Data Availability Statement: Data are available at: www.kaggle.com/marianarfranklin/mexico-covid19-clinical-data/.

Funding: The author(s) received no specific funding for this work.

Competing interests: The authors declare no conflict of interest.

Abstract

In the current era, a lot of research is being done in the domain of disease diagnosis using machine learning. In recent times, one of the deadliest respiratory diseases, COVID-19, which causes serious damage to the lungs has claimed a lot of lives globally. Machine learning-based systems can assist clinicians in the early diagnosis of the disease, which can reduce the deadly effects of the disease. For the successful deployment of these machine learning-based systems, hyperparameter-based optimization and feature selection are important issues. Motivated by the above, in this proposal, we design an improved model to predict the existence of respiratory disease among patients by incorporating hyperparameter optimization and feature selection. To optimize the parameters of the machine learning algorithms, hyperparameter optimization with a genetic algorithm is proposed and to reduce the size of the feature set, feature selection is performed using binary grey wolf optimization algorithm. Moreover, to enhance the efficacy of the predictions made by hyperparameter-optimized machine learning models, an ensemble model is proposed using a stacking classifier. Also, explainable AI was incorporated to define the feature importance by making use of Shapely adaptive explanations (SHAP) values. For the experimentation, the publicly accessible Mexico clinical dataset of COVID-19 was used. The results obtained show that the proposed model has superior prediction accuracy in comparison to its counterparts. Moreover, among all the hyperparameter-optimized algorithms, adaboost algorithm outperformed all the other hyperparameter-optimized algorithms. The various performance assessment metrics, including accuracy, precision, recall, AUC, and F1-score, were used to assess the results.

1. Background and rationale

Respiratory diseases are one of the main causes of mortality worldwide. Recently, one of the major respiratory diseases known as COVID-19, which has claimed a lot of lives globally, is one of the most disastrous pandemics seen by the human race in this century. The global lockdown and social distancing were new notions for the global population, living with these constraints was one of the most challenging tasks that changed the lifestyle of the population all around the world. The symptoms of this disease may vary from person to person depending upon the immunity of the body. The virus causing this deadly disease spreads to the respiratory tract of a person and causes grave damage to the lungs, which further leads to serious breathing problems which further reduces the oxygen level and requires further ventilator support for survival. COVID 19 alone has claimed over 6.5 million lives worldwide and diseases like chronic obstructive pulmonary disease (COPD) claim millions of lives every year worldwide [1]. Fig 1 shows the mortality in US alone in the year 2020 due to respiratory diseases.

While the World was battling with COVID-19, every possible avenue was explored to find the solution to this deadly disease [2]. To reduce the deadly effects of the disease, swift diagnosis of the disease is one of the most important factors to reduce the mortality rate due to this pandemic. Various diagnosis mechanisms, including real-time reverse transcriptase-polymerase chain reaction (RT-PCR), were taken into consideration for the diagnosis of the disease which appeared to be time time-consuming process [3]. Furthermore, different medical imaging systems, such as computed tomography (CT) and X-ray, can aid in swift diagnosis of the disease also new possibilities including artificial intelligence and big data can have been explored to control the spread of the pandemic [4].

In recent times, machine learning-based computer-aided diagnosis systems have come up as one of the most significant domains of research that assist radiologists in the accurate diagnosis of disease using medical images.

The ability of machine learning to adapt and learn from new data has enabled researchers to continuously refine strategies for managing and mitigating the impact of the disease, showcasing the potential of technology in addressing global health challenges [5]. Through the analysis of vast datasets, machine learning algorithms can be employed to predict the disease. Machine learning models have also been instrumental in developing diagnostic tools, such as predictive models for early detection of COVID-19 based on symptoms or imaging data [6].

In this context, this study aspires to utilize machine learning approaches and other clinical variables in the patient's data for the development of a predictive model that can identify

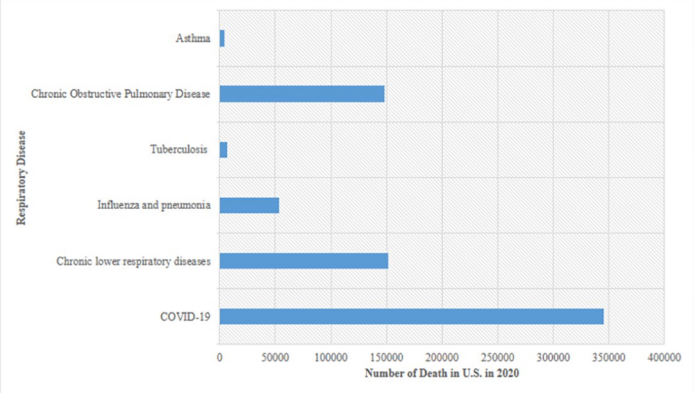

Fig 1. Mortality in U.S. in the year 2020 due to respiratory diseases.

https://doi.org/10.1371/journal.pone.0308015.g001

individuals with the existence of respiratory disease at an early stage and distinguish them from those who are healthy. Therefore, the main objective of this research is to assess and compare the outcome of the proposed model employing various hyperparameter tuning techniques with other state-of-the-art machine learning models.

1.1 Motivation

In recent times, machine learning has come up as one of the most promising domains of research proving its capability in the development of CAD systems for the diagnosis of various diseases [7]. However, for the impeccable deployment of these models, there is a huge room for improvement in various aspects viz. parameter tuning; for selecting the optimal parameters of the model that can lead to better results, and for feature selection; with an aim to reduce the dimensionality of dataset. This research considers the problem of respiratory disease classification and with an aim to improve the performance of existing machine learning algorithms, this research considers tuning the parameters of the algorithms and feature selection.

1.2. Problem identification

In this research, the authors aim to get into the bottom of two different problems required for the successful deployment of machine learning based systems which are parameter tuning and feature selection. Each machine learning algorithm has a different number and types of operations involving the use of different parameters [8]. For the successful deployment of these algorithms, the values of these parameters must be tuned to get the optimal set of values with an aim to achieve better classification accuracy. The problem of parameter tuning is regarded as an optimization problem that tries to optimize the various parameters to get the best set of parameters that aid in getting better accuracy [9]. The second problem considered by the researcher is feature selection, in which the authors try to reduce the dimensionality of the dataset by considering the most pertinent features and removing all the redundant features with an aim to increase the classification accuracy. Feature selection is also regarded as a multi-objective optimization problem that involves two different objectives viz. maximizing classification accuracy and minimizing the number of features [10].

Both the problems are complex optimization problems with different natures and require to be addressed differently to find the best solutions within a bearable time frame. Keeping in mind these goals an integrated system is required that simultaneously addresses all these issues and gives better classification accuracy.

1.3 Challenges and limitation of existing machine learning approaches in disease diagnosis

Machine learning (ML) has shown significant promise in disease diagnosis by automating and enhancing various aspects of the diagnostic process. However, there are several challenges and limitations that currently affect the efficacy and reliability of these approaches.

- Data quantity and quality.—High-quality, labeled medical data is scarce due to privacy constraints and the difficulty of obtaining sufficient cases for rare diseases, leading to data imbalances that bias machine learning models. Additionally, errors and inconsistencies in medical data from manual entry and diagnostic inaccuracies introduce noise, impairing model performance [11].
- Model related challenges.—Machine learning models in healthcare often overfit to training data, leading to poor generalization to new patients and variability across different settings, limiting their robustness. Many models, especially deep learning ones, are "black boxes," making

their decisions difficult to interpret, which hampers clinical trust. Additionally, these models can perpetuate biases from training data, resulting in unfair treatment across diverse patient groups and raising ethical concerns about equity and fairness in medical outcomes [12].

- Implementation Challenges.—Integrating machine learning models into clinical workflows presents challenges, including the need for significant changes in how clinicians operate and manage data, alongside potential resistance from users due to trust issues and concerns about job security. ML models require extensive validation in clinical settings, a process that is costly and time-consuming, compounded by complex regulatory requirements [13].
- Other Challenges.—Training and deploying machine learning models, especially deep learning ones, demands significant computational resources, which can be a constraint for many healthcare facilities, particularly when real-time processing is required. Despite advancements, manual feature engineering remains essential for capturing domain-specific knowledge, a process that is both labor-intensive and dependent on expertise. Selecting relevant features from complex medical data is also crucial but challenging for model performance.

This study integrates advanced machine learning techniques with a framework based on SHapley Additive ExPlanations (SHAP) to address the limitations mentioned earlier, significantly enhancing the accuracy of COVID-19 diagnostic predictions. Genetic Algorithms (GAs) are employed for hyperparameter optimization due to their efficiency and effectiveness in locating optimal solutions [14]. By simulating the principles of natural selection, GAs thoroughly explore the hyperparameter search space, which helps in developing superior machine learning models with a high likelihood of reaching the global minimum and avoiding local minima [15]. For feature selection, the binary grey wolf algorithm is used, drawing inspiration from the behavior of grey wolves during round-up and hunting. The algorithm incorporates four types of grey wolves—alpha, beta, delta, and omega—to emulate the leadership hierarchy [16]. The optimization process includes the three main steps of hunting. searching for prey, encircling prey, and attacking prey, which are applied to enhance the model's performance.

1.4 Research contributions

The contribution of the proposed research is fourfold; which has been summarized in the points below.

- Firstly, the algorithms' hyperparameter tuning is proposed to enhance the efficacy of the machine learning classification algorithms.
- Secondly, an ensemble learning model is developed considering the performance of the various parameter-tuned classification algorithms.
- Thirdly, to select the most relevant feature nature-inspired metaheuristic algorithm is considered for reducing the dimensionality of the dataset to increase the classification accuracy in a bearable time.
- Lastly, to comprehensively analyze the observations' prediction outcomes and interpret the justification behind the model's classification decisions, SHAP analysis is performed.

1.5 Structuring of the paper

The rest of the article is structured as follows. Section 2 presents a concise overview of state-of-the-art research in the domain of disease detection using machine learning. The proposed

model is presented in section 3. Section 4 describes the hyperparameter tuning with the Genetic algorithm. Section 5 briefly describes the dataset considered in this research. Section 6 presents experimental results and discussions. Feature importance using Explainable AI (SHAP Analysis) is discussed in section 7. Section 8 briefly presents the conclusions and future work.

2. Literature survey

This section summarizes the applications of machine learning in the domain of disease diagnosis, more specifically the diagnosis of COVID-19. Alali et al. [17] developed a highly efficient GPR-driven model to forecast the number of COVID-19 cases. The authors employed Bayesian optimization to fine-tune the hyperparameters of the Gaussian process regression in their model. Yank et al. [18] focused on enhancing the hyperparameters of well-known machine learning algorithms. Kumar et al. [19] presented an enhanced machine learning paradigm for the early detection of this illness. Modern Harris hawks optimization (HHO) algorithms based on random forest (HHORF), light gradient boosting (HHOLGB), extreme gradient boosting (HHOXGB), categorical boosting (HHOCAT) and support vector classifier (HHOSVC) were used to maximize the hyperparameters of the machine learning algorithms.

Mohsen et al. [20] used the generalized weighted ensemble with internally tuned hyperparameters (GEMITH) as a nested optimization-based technique that considers the tuning of hyperparameters and determining optimal weights for combining ensembles. Moreover, a heuristic approach was utilized to generate diverse and effective base learners, while Bayesian search was employed to expedite the optimization procedure.

Mohana et al. [21] used deep learning techniques on 350 images from X-ray datasets, the histogram equalization method was used for image preprocessing, and convolution neural network designs like ResNet-50 and VGG-16 were used for image categorization. The results indicated that, VGG-16 results in greater test and train precision. Further, to improve the results, hyper parameter optimization was used to fine-tune the VGG-16's precision.

Soufiane et al. [22] presented the effectiveness of five different machine learning algorithms, namely Random Forest, Ada Boost, XGBoost, SVM and Decision Tree. For training and evaluation in the first experiment, each model used default parameters. In the second trial, the author's employ the Grid Search function to identify the model's ideal setup on a collection of anonymous individuals with or without COVID-19 illness. Aljouie et al. [23] employed four widely used machine learning methods, along with three data balancing approaches and feature selection techniques. Mohammad et al. [24] used a variety of machine learning techniques to predict the mortality rate among COVID-19 patients.

Many researchers have considered the use of feature selection techniques [25] for the diagnosis of the disease, which have been summarized in this section. Mehrdad et al. [26] introduced a new method for diagnosing COVID-19 that combines feature selection with random forest. The proposed method enhances the feature space, simplifies complexity, and provides clinicians with a decision tree-like analysis, facilitating easier explanation. Experimental results demonstrated that the developed prediction model surpassed existing methods and baseline algorithms in terms of performance. Fatih et al. [27] presented a novel approach for detecting COVID-19 automatically, employing a combination of fused dynamic exemplar pyramid feature extraction and hybrid feature selection techniques using deep learning. Extensive testing on various datasets demonstrated the method's ability to achieve a high level of accuracy in detecting COVID-19. Chattopadhyay et al. [28] created various methods for COVID-19 detection, but only a few of them produced acceptable findings. The study makes two contributions,

i.e., extracting deep features from the image dataset before introducing a totally new feature selection method called Clustering-based Golden Ratio Optimizer (CGRO).

Kenway et al. [29] suggested a framework which was divided into three stages that are linked together. Initially, features are extracted from CT images using the Convolutional Neural Network (CNN) known as AlexNet. Next, a feature selection method called Guided Whale Optimization (Guided WOA) is employed, which is based on Stochastic Fractal Search (SFS). Pramanik et al. [30] proposed a computer-aided diagnosis (CAD) system for detecting Pneumonia from chest X-rays, employing deep learning and a metaheuristic algorithm. The approach involved extracting deep features from a pre-trained ResNet50 model, which is fine-tuned on a specific Pneumonia dataset. The proposed method is evaluated using well-known UCI datasets, gene expression datasets based on microarray analysis, and a dataset for predicting COVID-19. Yagin et al. [31] discusses a study that utilizes machine learning techniques, specifically the XGBoost algorithm, to classify and assess COVID-19 patients based on genomic biomarkers. The model aims to provide a clear interpretation of individualized and overall risk estimation for COVID-19, aiding physicians in understanding the impact of key genomic features. The study highlights the importance of external validation, integration of clinical risk factors, and the need for multi-center trials to enhance the predictive accuracy of the model. Additionally, the use of Local Interpretable Model-Agnostic Explanations (LIME) and SHapley Additive exPlanations (SHAP) frameworks improves the accuracy of COVID-19 diagnostic prediction and aids in explaining predictions to clinicians. Hamal et al. [32] presents a study on using machine learning models to classify COVID-19-associated lung changes from X-ray images. The research evaluates various models and identifies VGG-19 with data augmentation as the top performer, achieving high precision, recall, and F1 scores for COVID-19, pneumonia, and healthy individuals. The study emphasizes the importance of image pre-processing, tuning, and augmentation in enhancing model performance. Héberger et al. [33] addresses common errors in statistical modeling and focuses on the significance of using performance parameters correctly. It highlights the importance of distinguishing between linear and nonlinear models in modeling processes. The study involves a multicriteria decision-making process to compare various modeling equations and optimization algorithms. It emphasizes the role of variance analysis in detecting outliers and underscores the necessity of data preprocessing. Table 1 presents the comparison of the prominent techniques in the literature.

3. Proposed model

Machine learning has become one of the most significant domains of research these days and has its applications in various domains. For the successful deployments of machine learning models, certain unaddressed issues that can be considered for improvement as mentioned in the problem identification section.

In this light, authors in this research, aspire to use hyperparameters tuning and feature selection for machine learning algorithms to enhance the efficacy of the models. Fig 2 presents the primary steps in developing the proposed model.

Step I- Preprocessing

This step aims to balance the dataset using upsampling techniques, as the COVID-negative cases constitute only 10.5% of the entire dataset, whereas positive samples make up 89.5% (refer to section 5). Following this, certain attributes are subsequently removed from metadata that is not related to the study goal, such as id, ID_Registro, Pecho_Acc, ABR_INT, Fecha_actulization, Ingreso, Fecha_DEF, Pias_origen and naciolandad, etc. Additionally, RESULTADO is taken into account to be a dataset class that contains COVID yes COVID no labels.

Table 1. Comparison of key techniques in their literature.

Author	Dataset Type	Feature Selection	Hyperparameter Tuning	Technique Used	Future work
Dewi et al. [34]	csv	Boruta Feature Selection	Hyperband Optimization	Random Forest, XGBoost, Ensemble Methods	Combining different types of data (e.g., imaging, clinical, and genomic) can improve model robustness.
Soufiane et al. [22]	csv	No	Grid Search	Random Forest, Ada Boost, XGBoost, SVM and Decision Tree	Feature selection techniques and meta heuristic can be used for better performance.
Mehrdad et al. [26]	csv	Fisher score	No	XGBoost, SVM and MLP	Hyperparameter techniques and meta heuristic can be used for better performance.
Kumar et al. [19]	csv	No	Harris Hawks Optimization	Light gradient boosting, gradient boosting classifier, categorical boosting, random forest	More experiments can be conducted using feature selection methods to enhance model performance
Batista et al. [35]	csv	No	No	neural networks, random forests, gradient boosting trees, logistic regression and support vector machines	Feature selection techniques and Hyperparameter tuning can be used for better performance.
Chattopadhay et al. [28]	csv	Clustering-based Golden Ratio Optimizer	Wrapper-based FS algorithm	Support Vector Machines, K-Nearest Neighbor and Extreme Learning Machines.	Increased collaboration between generalizable models can lead to the development of more robust system.
Yasminah et al. [17]	csv	No	Bayesian Optimization	Support vector regression, Boosted trees, Bagged trees, Decision tree, Random Forest, and XGBoost	Feature selection techniques can be used for better performance.
Fatih et al. [27]	images	Local binary pattern	No	k-nearest neighbor	More models can be conducted to test performance.
Kukar et al. [36]	csv	No	No	random forest, neural network, the extreme gradient boosting machine and support vector machines	Feature selection techniques and Hyperparameter tuning can be used for better performance.
Meza et al. [31]	csv	No	No	Random forest, logistic regression, support vector machine, multilayer perceptron (neural network), stochastic gradient descent, XGBoost, and Adaboost	Feature selection techniques and Hyperparameter tuning can be used for better performance.

https://doi.org/10.1371/journal.pone.0308015.t001

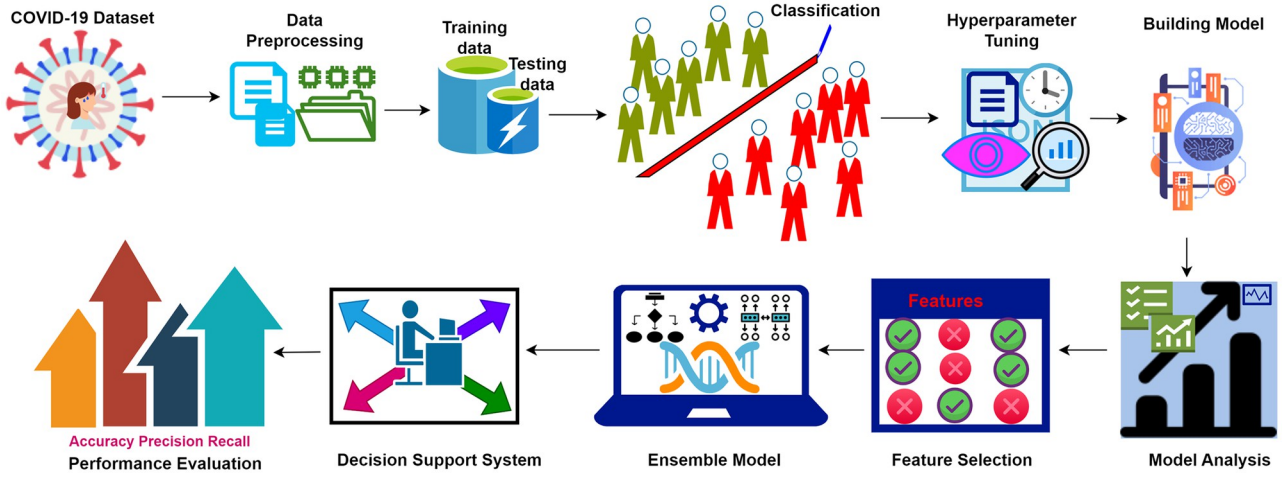

Fig 2. Proposed methodology.

https://doi.org/10.1371/journal.pone.0308015.g002

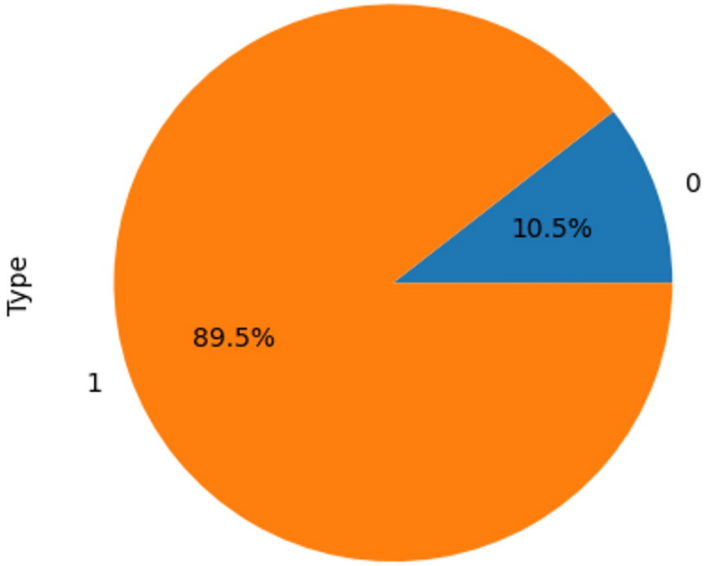

Fig 3. Before upsampling.

https://doi.org/10.1371/journal.pone.0308015.g003

As shown in Fig 3, before upsampling, the samples of the COVID-positive class accounted for 90 percent of the total, but after applying upsampling as shown in Fig 4, both the "yes" and "no" classes now possess an equal number of samples. Subsequent analyses and outcomes are based on this balanced dataset.

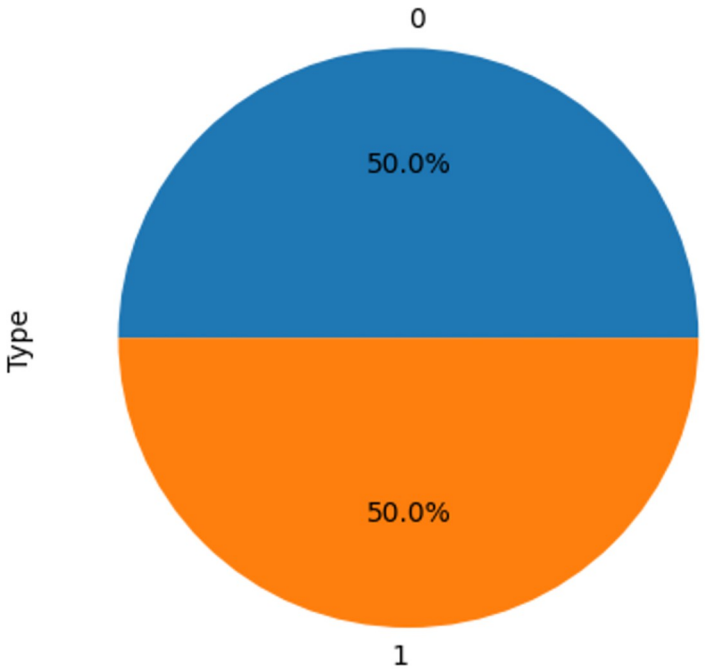

Fig 4. After upsampling.

https://doi.org/10.1371/journal.pone.0308015.g004

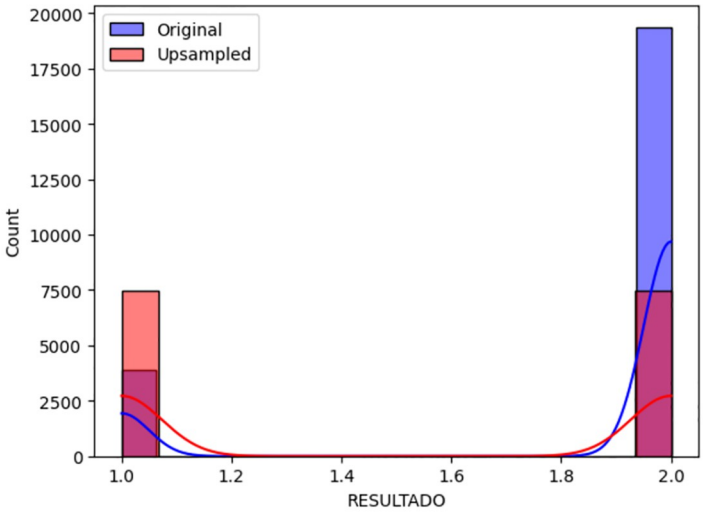

Fig 5. Comparison between original and upsampled dataset.

https://doi.org/10.1371/journal.pone.0308015.g005

Original and after upsampling, a graph illustrating in Fig 5 the count of COVID-positive (Class 1) and COVID-negative (Class 0) cases reveals a balanced dataset. This balance is critical as it ensures equal representation of both classes, thereby enhancing the performance and reliability of the machine learning models. The graph underscores the effectiveness of upsampling in addressing class imbalance, a key factor in improving predictive accuracy and reducing model bias. The Mexico dataset was selected due to its extensive records on respiratory diseases and symptoms, which are highly correlated with COVID-19, providing a robust basis for analysis and comparison.

Step II- Data splitting

For the training and assessment processes, the dataset ratios are 70% and 30%. Two tests were run. In the first, we used the models' preset hyperparameters for training and testing. The confusion matrix was then created after we had computed the success measures. Secondly, classification results were taken with hyperparameter tuning.

Step III- Classification algorithm

The presented system used seven classifiers. Adaboost, Random forest, Extra tree, Decision Tree, Gradient Boosting Classifier, KNN and Light Gradient Boosting Machine.

Step IV- Hyperparameter tuning

Numerous machine learning applications in the actual world heavily rely on hyperparameter optimization. The hyperparameters of these algorithms can be optimized to boost the efficiency of these algorithms. Genetic algorithms, random search, Bayesian Optimization, and grid search are used as optimization methods. The various hyperparameters used by various classifiers are.

LightGBM. num leaves, bagging fraction, feature fraction, learning rate, max depth, subsample, colsample tree, max bin, min child samples.

Adaboost. (subsample, colsample tree, gamma, max depth, min child weight, learning rate, alpha).

Random Forest. (n estimator, criterion, max depth, min sample split).

The efficiency of the categorization can be improved by carefully choosing (tuning) the values of the hyperparameters. When an optimization algorithm is present, the tuning process can be completed, and the full process is referred to as an optimization issue.

Step V- Building and model analysis

The performance was evaluated by considering the confusion matrix with several metrics, including precision, accuracy, area under the curve, error rate, balanced accuracy score, cross-validation score, Kappa index, and F1-score. The 2X2 CM has been used in the suggested model hyperparameter optimization-based ML method to assess the model using the mentioned metrics. The results that were properly categorized are represented by the categorization along the main diagonal. (Higher numbers of the metrics, excluding the error stated above, indicate a more effective model.

Step VI- Feature selection

The proposed research uses one of the latest nature-inspired metaheuristic algorithms, "grey-wolf optimizer" that imitates the natural command structure and foraging strategy of grey wolves [37] for feature selection. The algorithm searches the space of features to find the best features from the original set of features with an aim to maximize the accuracy of prediction and minimize the number of features selected. Fig 6 presents the feature selection process considered in this research. Using feature selection, one can determine the crucial features and eliminate the unnecessary (redundant) ones from the dataset [38]. For various machine learning applications, the feature selection goals include reducing data dimensionality, enhancing prediction performance, and providing good data understanding [39].

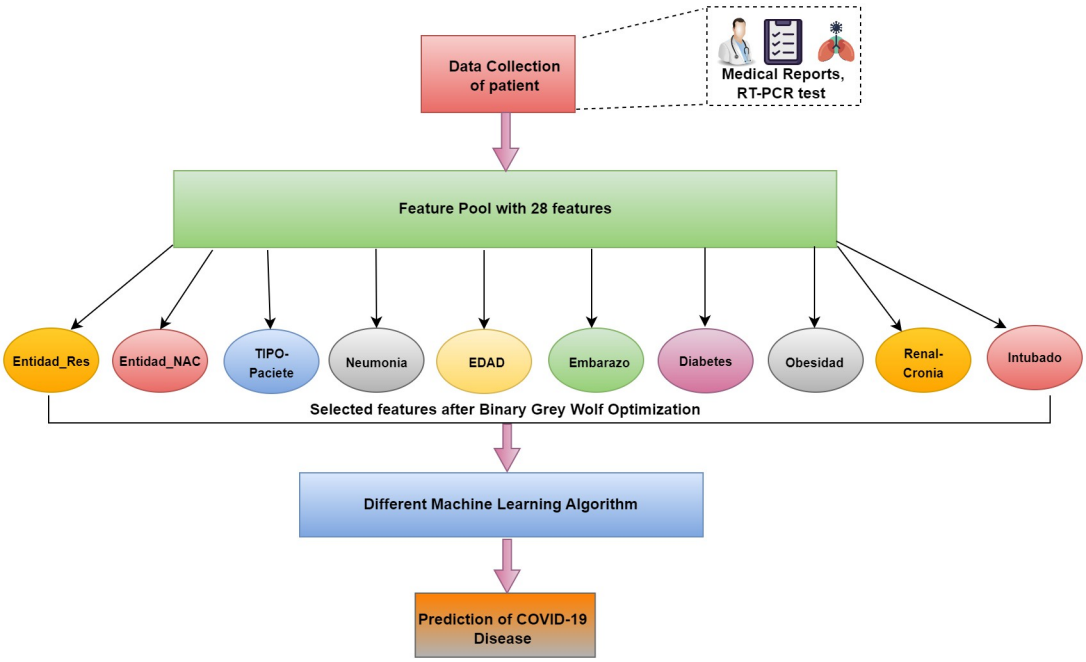

Fig 6. Feature selection process.

https://doi.org/10.1371/journal.pone.0308015.g006

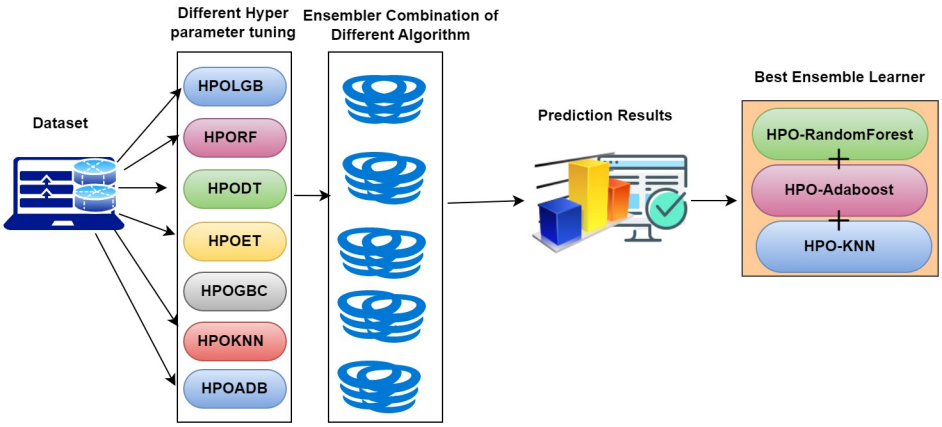

Fig 7. Ensemble model architecture.

https://doi.org/10.1371/journal.pone.0308015.g007

Step VII- Ensemble model

In the proposed model, stacking method [40] is used as ensemble learning. The strategy of this method is employed to enhance the predictive efficacy of machine learning models. This approach entails amalgamating several foundational models to construct a more robust meta-model that capitalizes on the distinct capabilities of each foundational model.

The fundamental concept behind stacking revolves around incorporating one or more meta-level models, which accept predictions from multiple foundational models as inputs and subsequently generate the ultimate prediction. The greatest precision is achieved when Ada-boost, KNN, and Random forest are combined as shown in Fig 7. The mathematical formula [41, 42] is demonstrated as Eq 1

$$y = mode\{cl1, cl2, cl3\} \tag{1}$$

y is the stacking classifier for getting the result by adding three machine learning with the best result.

Step VIII- Performance evaluation

A performance assessment model enables precision and efficiency evaluations. There are numerous methods for rating classifiers. In this study, we utilized the Holdout technique, which involves partitioning the dataset into two separate subsets. a test set and a train set, with each comprising 30% and 70% of the dataset, respectively. The training process involved using the train set to train the data, and afterward, we assessed its predictive capabilities by evaluating it on the hidden test set [43]. To mitigate overfitting in the proposed model, a feature selection process was employed to eliminate noise and remove features that were either redundant or of minimal importance for prediction accuracy. Additionally, an ensemble modeling approach was adopted to further reduce overfitting. Ensemble methods enhance model performance by combining multiple weak learners, which collectively produce more accurate and robust results. By leveraging multiple models to analyze the data, ensemble techniques ensure that the final predictions are more reliable and precise. Additionally, we employed the Cross-validation technique to prevent the over-fitting issue. Then, we determined some assessment metrics, including the F1 score, ROC, memory, accuracy, and precision [44].

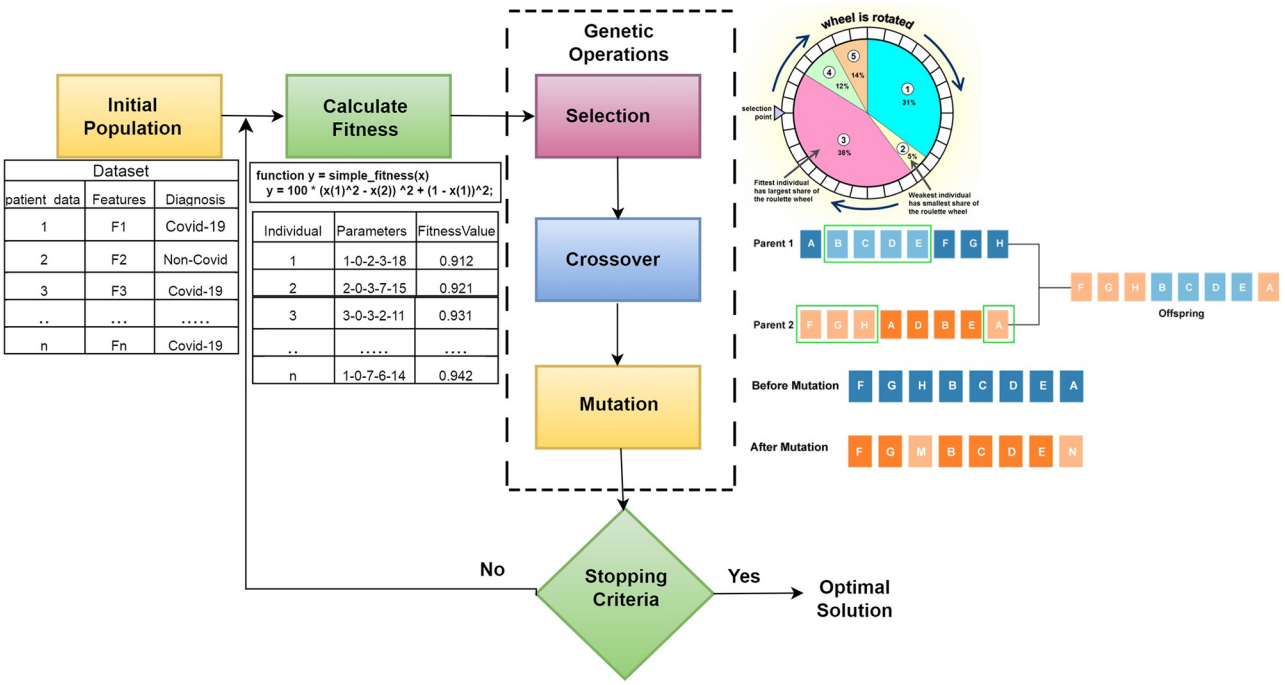

Fig 8. The framework of Genetic algorithm for hyper-parameter optimization.

https://doi.org/10.1371/journal.pone.0308015.g008

4. Hyperparameter tuning using Genetic algorithm

Genetic algorithms can optimize machine learning algorithm's hyperparameters by systematically exploring potential hyperparameter combinations. Fig 8 shows the structure of genetic hyperparameter tuning on a machine learning algorithm [45].

Start by identifying the hyperparameters to fine-tune the machine learning model, such as learning rates or layer sizes. For each hyperparameter, specify the range or values it can take.

Population initialization

Begin with an initial set of hyperparameter combinations, forming a population of candidates. A population comprises individuals or solutions, typically referred to as chromosomes. Each chromosome is composed of a series of genes, where each gene, or multiple genes, depending on the encoding method, represents a single decision variable that will be applied to the objective functions. Various parameter combinations lead to diverse fitness values in vectors. Random mutations in the parameters are introduced within the population, and vectors with higher fitness levels outlive their counterparts [46].

Evaluation phase. Train and assess a machine learning model for each candidate hyperparameter set. Use a performance metric (e.g., accuracy, error) to quantify how well each model performs.

Selection. Choose a subset of candidates (called parents) for the next generation based on their model performance. Candidates who achieve better results are more likely to be selected. Common selection methods include random sampling or ranking candidates.

Crossover (Recombination). Pair up the selected parents and generate new candidate hyperparameter sets (offspring) by merging their hyperparameter values. Crossover can involve blending or swapping hyperparameter values between parent candidates to create offspring.

Table 2. Algorithm for generating hyperparameter.

```
Algorithm. Genetic Algorithm for Hyperparameter Tuning
Result. The fittest hyperparameter in the population
populations [list of n models with different hyperparameters
generation 0;
while generation<max generation do
train_and_evaluate(population);
new_gen ← retains the m fittest individuals;
new_gen ← append random individuals to promote diversity;
mutate(mew_gen);
new_gen ← append offsprings through crossover until k;
population ← new_gen;
generation ← generation+1
end
```

https://doi.org/10.1371/journal.pone.0308015.t002

Mutation. Introduce small, random changes to hyperparameters in some offspring candidates. This introduces diversity into the population. Mutation helps explore the hyperparameter space more extensively.

Population update. Replace some existing candidates with the newly generated offspring candidates. The selection for replacement is often based on the fitness (performance) of the candidates. This step ensures the population size remains consistent.

Termination conditions. Specify when to stop the optimization process. This can be based on a maximum number of iterations, a performance threshold, or a time limit.

Final result. The hyperparameter set that results in the best model performance during the optimization process is considered the optimal configuration [47].

The algorithm for generating hyperparameters using the genetic algorithm is as shown in Table 2.

Here's a breakdown of how this process works.

1. **Generate Initial Population**. Begin by creating an initial set of machine learning (ML) models with randomly selected hyperparameters.
2. **Evaluate Loss Function**. Determine the loss function for each model, such as log-loss, to measure their performance.
3. **Select Top Models**. Identify and select a subset of models with the lowest error rates.
4. **Create Offspring**. Develop a new population of ML models by generating offspring from the top-performing models of the previous generation, making slight adjustments to their hyperparameters. Combine these offspring with models from the previous generation and new models in a specific ratio, for instance, 50/50.
5. **Iterate the Process**. Calculate the loss function for the new population, rank the models, and repeat the process for multiple generations.

Genetic algorithms, while powerful, require careful specification of the loss function, population size, and the ratio of offspring with modified parameters [48].

5. Dataset description

The dataset [49] contains 41 columns which includes clinical data as well as RT-PCR test. The 41 columns have certain attributes that aren't required for the findings, thus they're omitted from the dataset. Some non-relevant fields have been eliminated, including the patient's ID, city name, and patient registration date, as well as nine additional columns. Table 3 shows that

Table 3. Dataset description.

S. No.	Attribute Name	Description
1.	Entidad_um	Region where hospital performed admission
2.	Entidad_Res	Residence of the patient at which region
3.	Delay	Lag in the process of lab report
4.	Entidad_Registro	The actual region from where case assigned
5.	Origen	surveillance of patient (1 = yes, 2 = no)
6.	Sector	identify the institute of national health system
7.	Sexo	gender of patient 1 = female, 2 = male and 99 for undisclosed
8.	Entidad_nac	patient birth state or region
9.	Tipo_paciente	type of care patient received (1 = outpatient, 2 = impatient)
10.	Neumonia	Identifies whether the patient was diagnosed with pneumonia
11.	Edad	Age of the patient
12.	Nacionalidad	check whether patient is Mexican(1) or foreign(2)
13.	Embrazo	Identifies patient is pregnant or not
14.	Habla_lengua_Indig	Patient speaks an indigenous language
15.	Diabetes	Identifies whether the patient was diagnosed with diabetes
16.	EPOC	Classify whether the patient detect with pulmonary disease
17.	Asma	Classify whether the patient diagnosed with asthma or not
18.	Immusupr	Identifies if the patient is immune suppressed
19.	Hipertension	Classify whether the patient diagnosed with hypertension
20.	Otra_Com	Identifies if the patient presents another disease
21.	Cardiovascular	Classify whether the patient diagnosed with cardiovascular disease or not
22.	Obesidad	Classify whether the patient diagnosed with obesity or not
23.	Renal_Cronica	Identifies if chronic renal insufficiency
24.	Tabaquismo	Identifies if tobacco addiction
25.	Otro_Caso	Classify whether the patient diagnosed with any other case diagnosed with SARS COV-2
26.	Migrante	Identifies if the patient is migrant
27.	UCI	Identifies if the patient was admitted to ICU
28.	Intubado	patient need intubation or not(1 = yes, 2 = no,97 = not applicable)
29.	Resultado	The RT-PCR test (1 = positive, 2 = negative)

https://doi.org/10.1371/journal.pone.0308015.t003

the 29 most relevant columns. Out of all the samples in the dataset, subjects having initial respiratory problems were considered for the study which make about 14964 patients' samples [49]. The column's values are the description of attributes. The value 1 means yes and the value 0 means no. Table 3 shows attributes' descriptions as there are 29 main attributes consisting of personal attributes and clinical features [49].

6. Experimental results and discussions

In this section, experimental results obtained after implementation Binary Grey Wolf Optimization for feature selection on dataset. It significantly impacts machine learning model performance by enhancing predictive accuracy, reducing computational complexity, improving interpretability, and ensuring robust generalization. Its effective exploration and exploitation strategies enable the selection of optimal feature subsets, contributing to the development of more efficient, reliable, and scalable machine learning models [50]. Further experimental results obtained after implementation and execution of the proposed model are compared with other state of the art machine learning algorithms, which are decision tree, adaboost,

Table 4. Results of machine learning algorithm.

Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT(Sec)
Decision Tree Classifier	**0.9593**	**0.9722**	**0.9926**	**0.9484**	**0.9700**	**0.9385**	**0.9396**	**0.0810**
Extra Trees Classifier	0.9564	0.9906	0.9935	0.9425	0.9673	0.9328	0.9342	1.1020
Random Forest Classifier	0.9544	0.9872	0.9962	0.9367	0.9655	0.9288	0.9307	1.2150
Gradient Boosting Classifier	0.9489	0.9821	0.9948	0.9283	0.9604	0.9179	0.9203	0.7160
Light Gradient Boosting Machine	0.9484	0.9819	0.9977	0.9250	0.9600	0.9167	0.9196	0.4200
Ada Boost Classifier	0.9475	0.9776	0.9877	0.9200	0.9583	0.9129	0.9165	0.4590
Logistic Regression	0.9469	0.9739	0.9861	0.9198	0.9582	0.9127	0.9163	1.0460
Naive Bayes	0.9454	0.9709	0.9776	0.9198	0.9582	0.9127	0.9163	0.0780
Ridge Classifier	0.9434	0.0000	0.9678	0.9198	0.9582	0.9127	0.9163	0.0650
Extreme Gradient Boosting	0.9384	0.9788	0.9647	0.9198	0.9582	0.9127	0.9163	1.0720
Linear Discriminant Analysis	0.9364	0.9734	0.9577	0.9198	0.9582	0.9127	0.9163	0.1010
SVM—Linear Kernel	0.9176	0.0000	0.9319	0.9093	0.9149	0.8352	0.8437	0.1390
K Neighbors Classifier	0.8300	0.9100	0.8862	0.7967	0.8390	0.6599	0.6644	0.1750
Quadratic Discriminant Analysis	0.5000	0.0000	0.8570	0.5000	0.6667	0.0000	0.0000	0.0800

https://doi.org/10.1371/journal.pone.0308015.t004

random forest, gradient boosting, light gradient boosting, extra tree, logistic regression, ridge classifier, linear discriminant analysis, naïve bayes, K-nearest neighbor and support vector machine, based on different evaluation metrics like accuracy, precision, recall, f1-score, Kappa_stat, MCC, and time required [51].

To validate the results 3-fold cross validation technique is considered. Table 4 summarizes the results obtained by executing different parameters as mentioned above. The results indicate the outperformance of the decision tree classifier classification algorithm in terms of different evaluation parameters.

Fig 9 represents the confusion matrix obtained and the corresponding ROC curve for the top three best algorithms which are the Decision tree, Extra tree and Random Forest. The

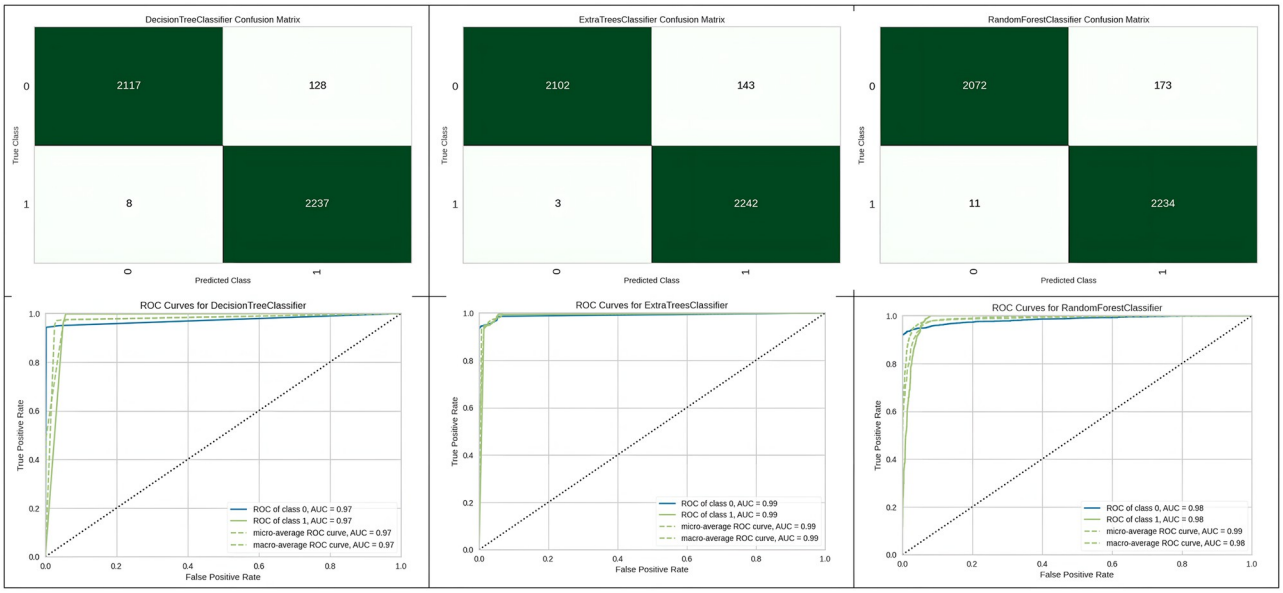

Fig 9. Best three classifier confusion matrix and ROC curve.

https://doi.org/10.1371/journal.pone.0308015.g009

Table 5. Random forest with hyperparameter optimization.

S. no	Hyperparameter optimization Algo	Accuracy	Optimized parameter	Computation Time
1	Default Hyperparameters	0.952053729	max_depth = 2, random_state = 0	10.5s
2	Grid Search	0.9490457097	{'criterion'. 'gini', 'max_depth'. 15, 'n_estimators'. 20}	6.06s
3	Random Search	0.9511253	{'max_depth'. 19,, 'min_samples_leaf'. 8, 'min_samples_split'. 10,criterion'. 'gini', 'max_features'. 16, 'n_estimators'. 56'}	7.89s
4	**Genetic Algorithm**	**0.9585935**	**min_samples_split = 4,max_depth = 92, min_sample s_leaf = 6, max_features = 10, n_estimators = 92**	**15.8s**
5	Bayesian Optimization	0.948092	{'min_samples_leaf'. 4.0, 'max_depth'. 39.0, 'min_samples_split'. 3.0, 'criterion'. 1, 'n_estimators'. 97.0,'max_features'. 5.0}	30.5s

https://doi.org/10.1371/journal.pone.0308015.t005

decision tree model achieved a sensitivity of 96%, a specificity of 92%, and a positive likelihood ratio of 18.19. In comparison, the Extra Trees model demonstrated a sensitivity of 95.5%, while the Random Forest model yielded a sensitivity of 94%, both of which are slightly lower than the sensitivity achieved by the decision tree.

Pondering further, seven different algorithms were considered for hyperparameter optimization using grid search, random search, and Bayesian Optimization and Genetic algorithm techniques. Table 5 represents the results obtained after performing the hyperparameter optimization. The results indicate that when hyperparameter optimization of random forest is performed using a genetic algorithm, the results indicate the outperformance of the algorithm as compared to other algorithms considered for hyperparameter optimization.

Fig 10 represents the confusion matrix obtained, corresponding ROC curve and classification report of the random forest with genetic algorithm hyperparameter optimization.

Table 6 represents the results obtained after performing the hyperparameter optimization on the gradient boosting classifier. The results indicate that when hyperparameter optimization of gradient boosting classifier is performed using a genetic algorithm, the results indicate the outperformance of the algorithm as compared to other algorithms considered for hyperparameter optimization.

Fig 11 represents the classification report, confusion matrix obtained and corresponding ROC curve of the gradient boosting classifier with genetic algorithm hyper parameter optimization. Furthermore, the outperformance of the Adaboost classifier with the genetic algorithm is represented in Table 7. The computation time of random search is often higher than other methods due to its time complexity.

Fig 12 represents the true positive and false positive values in the confusion matrix and ROC curve of the Adaboost classifier with micro average and AUC. Table 8 represents the

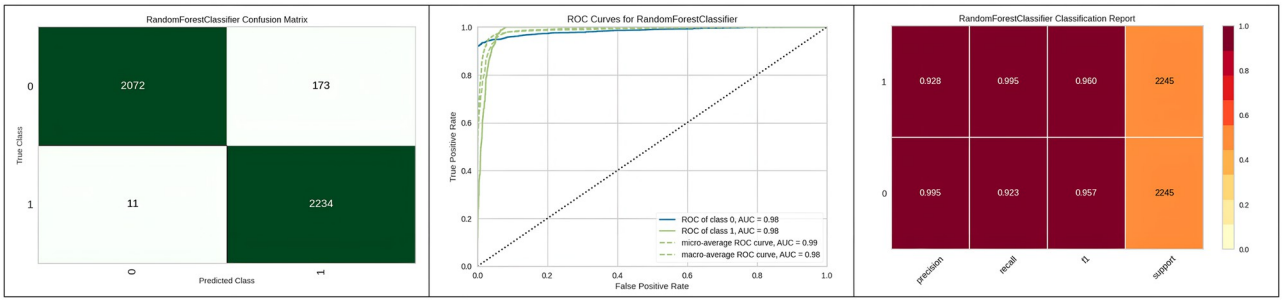

Fig 10. Best optimizer results for random forest.

https://doi.org/10.1371/journal.pone.0308015.g010

Table 6. Gradient boosting classifier results.

S. no	Hyperparameter optimization Algo	Accuracy	Optimized parameter	Computation Time
1	Default Hyperparameters	0.948578654	{'max_depth'.3,'random_state'. none, 'learning_rate'. 0.1, 'subsample'. 1.0, 'n_estimators'.100,}	110.5s
2	Grid Search	0.94905854	{'subsample'. 0.7,'learning_rate'. 0.1, 'n_estimators'.250,'random_state'. 1, 'max_depth'.3}	168.06s
3	Random Search	0.948323611	{'subsample'. 0.5, 'random_state'. 1,, 'max_depth'. 2, 'n_estimators'. 1000, 'learning_rate'. 0.1}	87.89s
4	**Genetic Algorithm**	**0.9521277**	**{'subsample'. 0.5, 'random_state'. 1, 'max_depth'. 2, 'n_estimators'. 745, 'learning_rate'. 0.01}**	**185.8s**
5	Bayesian Optimization	0.9501281	{'subsample'. 0.75, 'random_state'. 1, 'learning_rate'. 0.01,'n_estimators'. 1795, 'max_depth'. 1}	30.5s

https://doi.org/10.1371/journal.pone.0308015.t006

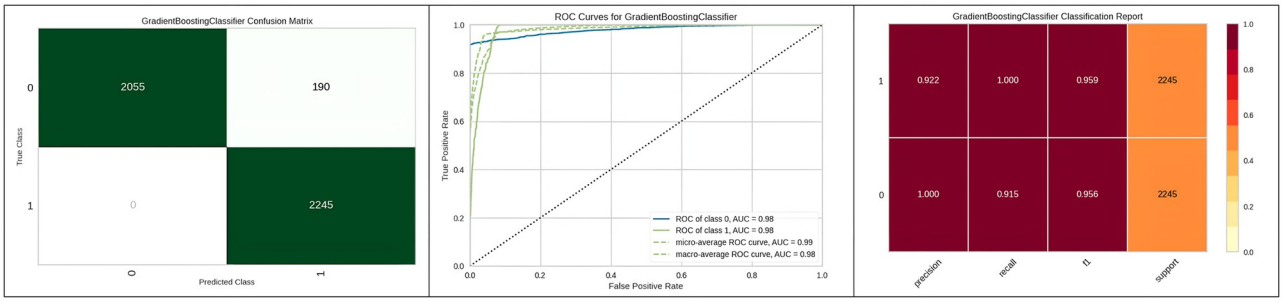

Fig 11. Best optimizer result for gradient boosting classifier.

https://doi.org/10.1371/journal.pone.0308015.g011

Table 7. Adaboost classifier results.

S.no	Hyperparameter optimization Algo	Accuracy	Optimized parameter	Computation Time
1	Default Hyperparameters	0.94732345	learning_rate = 1.0, algorithm = 'SAMME.R', n_estimators = 50	160.5s
2	Grid Search	0.94724405	{'algorithm'. 'SAMME.R', 'n_estimators'. 4, 'learning_rate'. 1.01}	178.06s
3	Random Search	0.9462440	{'learning_rate'. 0.99, 'algorithm'. 'SAMME.R', 'n_estimators'. 7}	197.89s
4	**Genetic Algorithm**	**0.95696142**	**{'learning_rate'. 1.02, 'algorithm'. 'SAMME.R', 'n_estimators'. 8}**	**135.8s**
5	Bayesian Optimization	0.94912440	{'n_estimator'. 20, 'learning rate'. 1.02, algorithm. 'SAMME'}	260.5s

https://doi.org/10.1371/journal.pone.0308015.t007

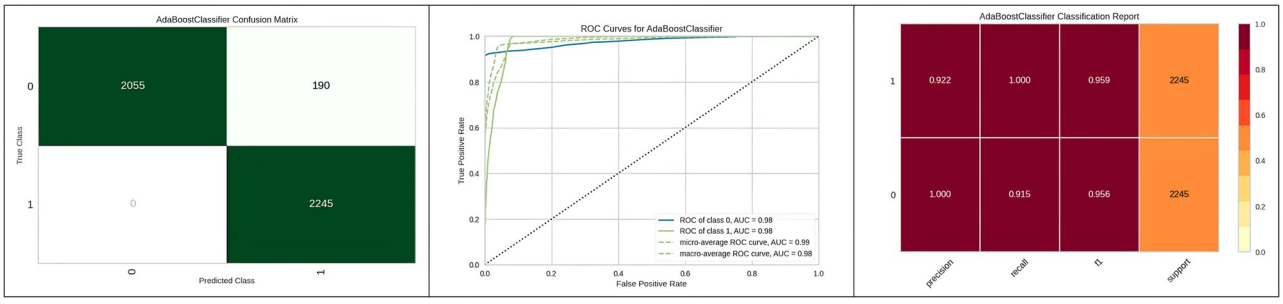

Fig 12. Best optimizer result for Adaboost classifier.

https://doi.org/10.1371/journal.pone.0308015.g012

Table 8. Results of Extra tree.

S.no	Hyperparameter optimization Algo	Accuracy	Optimized parameter	Computation Time
1	Default Hyperparameters	0.95631	{n_estimators = 100, *, max_depth = None, min_samples_split = 2, criterion = 'gini'}	160.5s
2	Grid Search	0.94428	{'max_features'. 3, 'n_estimators'. 100, 'min_samples_leaf'. 5}	178.06s
3	Random Search	0.948318	{'n_estimators'.200,'min_samples_leaf'. 5, 'max_features'. 4}	197.89s
4	**Genetic Algorithm**	**0.957013**	**{'n_estimators'.500, 'max_features'. 4, 'min_samples_leaf'. 5}**	**135.8s**
5	Bayesian Optimization	0.942421812	min_samples_leaf = 10, n_estimators = 300, max_features = 3	260.5s

https://doi.org/10.1371/journal.pone.0308015.t008

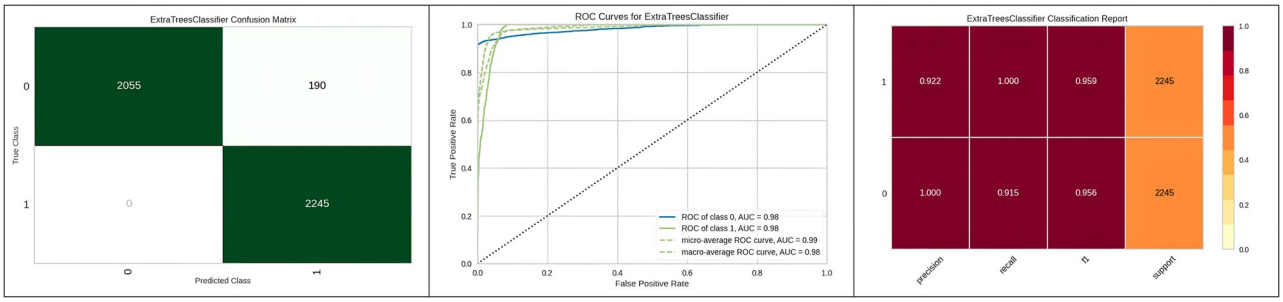

Fig 13. Best optimizer result for Extra tree.

https://doi.org/10.1371/journal.pone.0308015.g013

optimized parameters of Extra Trees using various optimization algorithms. Additionally, Table 8 illustrates the improved results of Extra Trees with the genetic algorithm optimization compared to other hyperparameter optimization algorithms.

Fig 13 represents the confusion matrix, ROC curve, and classification report of Extra Trees with genetic algorithm hyperparameter optimization, showcasing its higher performance compared to other optimization algorithms. Moving ahead, Table 9 represents the results obtained after performing the hyperparameter optimization on the Light gradient boosting machine. The results indicate that when hyperparameter optimization of the Light gradient boosting machine is performed using a grid search algorithm, the results indicate the outperformance of the algorithm as compared to other algorithms considered for hyperparameter optimization.

Table 9. Results of Lightbgm.

S. no	Hyperparameter optimization Algo	Accuracy	Optimized parameter	Computation Time
1	Default Hyperparameters	0.94845	(num_leaves = 31, max_depth = 1, learning_rate = 0.1, n_ colsample_bytree = 1.0, reg_lambda = 0.0, estimators = 100)	160.5s
2	**Grid Search**	**0.956261**	**(boosting_type = 'dart', colsample_bytree = 0.6, learning_rate = 1, max_depth = 5, n_estimators = 20, num_leaves = 5, reg_lambda = 1)**	**158.06s**
3	Random Search	0.949122	{'reg_lambda'. 0.01, 'num_leaves'. 25, 'n_estimators'. 35, 'boosting_type'. 'gbdt','max_depth'. 15, 'colsample_bytree'. 1, 'learning_rate'.0.1}	197.89s
4	Genetic Algorithm	0.948192	(n_estimators = 35, boosting_type = dart,colsample_bytree = 1, max_depth = 5, num_leaves = 50, reg_lambda = 0.1, learning_rate = 0.1)	195.8s
5	Bayesian Optimization	0.947296	(learning_rate = 1, max_depth = 15, n_estimators = 35, num_leaves = 5, reg_lambda = 1, colsample_bytree = 0.6)	260.5s

https://doi.org/10.1371/journal.pone.0308015.t009

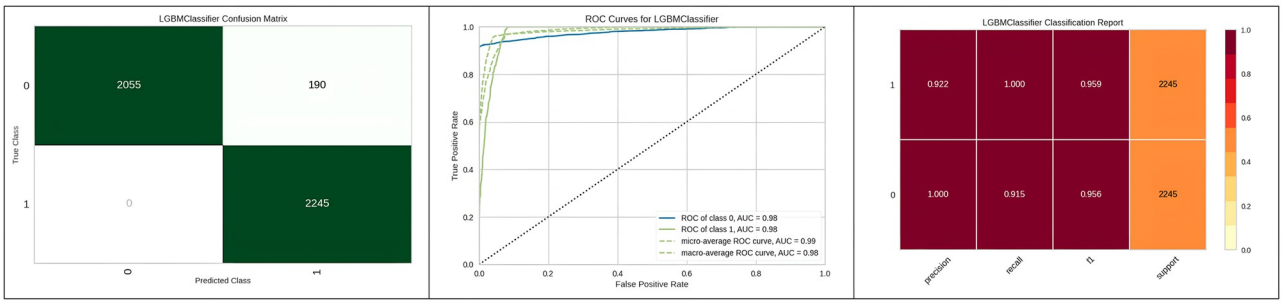

Fig 14. Best optimizer results of Lightbgm.

https://doi.org/10.1371/journal.pone.0308015.g014

Table 10. Results of Decision tree.

S.no	Hyperparameter optimization Algo	Accuracy	Optimized parameter	Computation Time
1	Default Hyperparameters	0.9515469	{random_state = 5,max_depth = 12,criterion = 'ginni'}	80.5s
2	Grid Search	0.94546913	{random_state = 8,max_depth = 19,criterion = 'entropy'}	156.06s
3	Random Search	0.951765	{random_state = 25,max_depth = 22,criterion = 'ginni'}	107.89s
4	**Genetic Algorithm**	**0.9566811**	**{random_state = 42,max_depth = 7,criterion = 'entropy'}**	**135.8s**
5	Bayesian Optimization	0.94684513	{random_state = 5,max_depth = 12,criterion = 'entropy'}	230.5s

https://doi.org/10.1371/journal.pone.0308015.t010

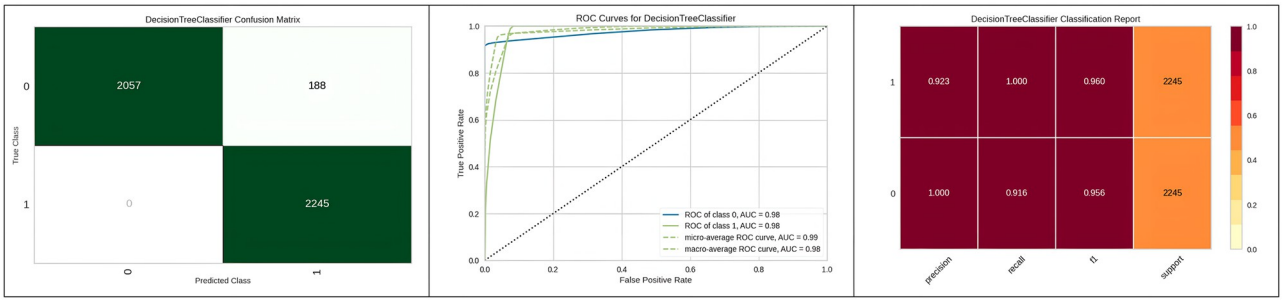

Fig 15. Best optimizer results of Decision tree.

https://doi.org/10.1371/journal.pone.0308015.g015

Fig 14 represents the better results of the Light gradient boosting machine with grid search hyperparameter optimization as a classification report in the form of precision, recall and support report with a graphical representation of ROC and confusion matrix.

Table 10 represents the optimized parameters of the Decision tree using various optimization algorithms. Additionally, Table 10 illustrates the improved results of the Decision tree with the genetic algorithm optimization compared to other hyperparameter optimization algorithms.

Fig 15 represents the confusion matrix of the model, ROC curve and classification report of SVM with genetic algorithm hyperparameter optimization.

Table 11 represents the outcomes following the hyperparameter optimization process. The results demonstrate that the genetic algorithm for hyperparameter optimization in the KNN gives better results than other algorithms considered for the same purpose.

Fig 16 represents the ROC curve and classification value report of KNN with the best hyperparameter.

Table 11. Results of KNN.

S.no	Hyperparameter optimization Algo	Accuracy	Optimized parameter	Computation Time
1	Default Hyperparameters	0.830099	{'n_neighbors'. 5}	60.5s
2	Grid Search	0.82604116	{'n_neighbors'. 20}	176.06s
3	Random Search	0.830941	{'n_neighbors'. 11}	147.89s
4	**Genetic Algorithm**	**0.8354435**	**KNeighborsClassifier__n_neighbors = 17**	**167.8s**
5	Bayesian Optimization	0.8265757	{'n_neighbors'. 13.0}	350.5s

https://doi.org/10.1371/journal.pone.0308015.t011

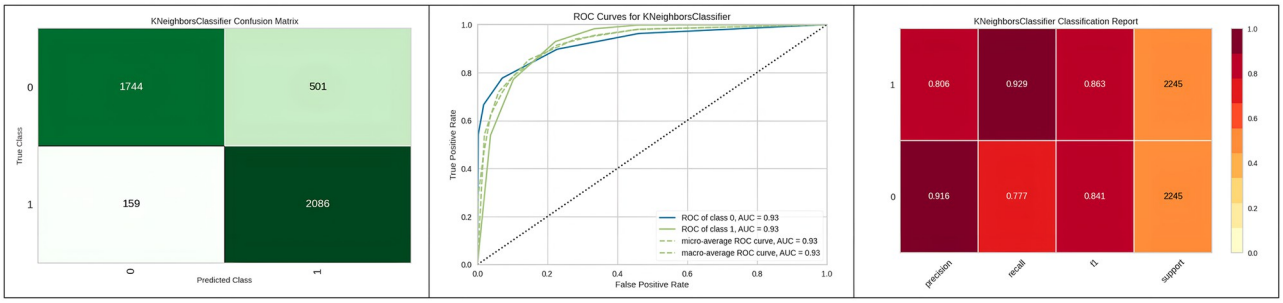

Fig 16. Best optimizer results of KNN.

https://doi.org/10.1371/journal.pone.0308015.g016

Table 12. Results of gradient boosting classifier.

S. no	Hyperparameter optimization Algo	Accuracy	Optimized parameter	Computation Time
1	Default Hyperparameters	0.95072440	{'max_depth'. 3,'n_estimators'.100,'random_state'. none, 'learning_rate'. 0.1, 'subsample'. 1.0}	210.5s
2	**Grid Search**	**0.9564311**	**{'learning_rate'. 0.1, 'n_estimators'. 2000, 'random_state'. 1, 'subsample'. 0.5, 'max_depth'. 1}**	**187.06s**
3	Random Search	0.95191281	{'max_depth'. 4, 'subsample'. 0.5, 'random_state'. 1, 'n_estimators'. 1000, 'learning_rate'. 0.01}	57.89s
4	Genetic Algorithm	0.9492614	{'max_depth'. 1, 'subsample'. 0.75, 'random_state'. 1, 'n_estimators'. 1897, 'learning_rate'. 0.01}	298.8s
5	Bayesian Optimization	0.95081281	{'max_depth'. 1, 'subsample'. 0.75, 'random_state'. 1, 'n_estimators'. 1897, 'learning_rate'. 0.01}	380.5s

https://doi.org/10.1371/journal.pone.0308015.t012

Pondering further, the dataset was reduced using the Binary Grey wolf feature selection technique to find the most relevant features. The dataset with selected features was then used for further testing. Table 12 represents the results of the gradient boosting classifier using a feature selection dataset. The results in Table 12 clearly show the outperformance of the gradient boosting classifier with the Genetic algorithm of hyperparameter optimization.

Fig 17 presents the confusion matrix, ROC curve and classification report obtained after performing hyperparameter optimization using the algorithm on the Gradient boosting classifier.

Table 13 represents the results of Adaboost with a genetic algorithm, demonstrating high accuracy values. Additionally, Fig 18 represents the confusion matrix, classification report and ROC curve to represent the accuracy of the best optimizer's results on Adaboost. The feature selection technique reduces the execution time of the Extra tree with random search hyperparameter optimization with the Genetic Algorithm.

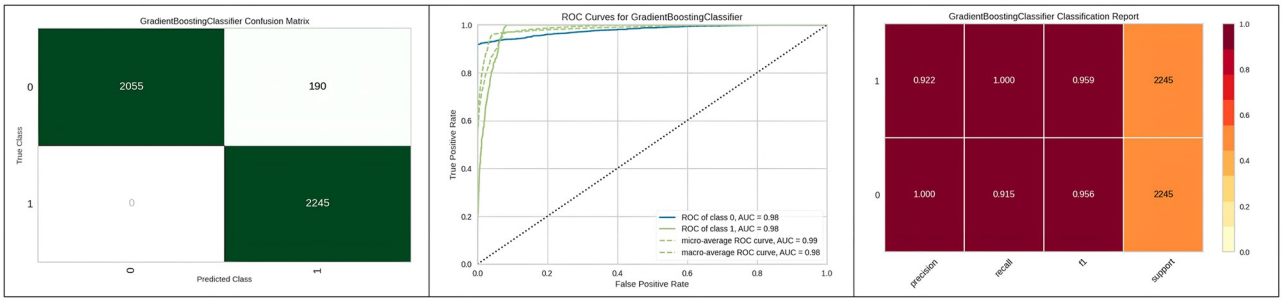

Fig 17. Best optimizer results of gradient boosting classifier.

https://doi.org/10.1371/journal.pone.0308015.g017

Table 13. Results of Adaboost.

S.no	Hyperparameter optimization Algo	Accuracy	Optimized parameter	Computation Time
1	Default Hyperparameters	0.9505268	learning_rate = 1.0, n_estimators = 50, algorithm = 'SAMME.R',	210.5s
2	Grid Search	0.951927292	{'learning_rate'. 1.0, 'algorithm'. 'SAMME.R', 'n_estimators'. 20}	198.06s
3	Random Search	0.9498329	{'learning_rate'. 0.97, 'n_estimators'. 20, 'algorithm'. 'SAMME.R'}	126.89s
4	**Genetic Algorithm**	**0.9565291**	**{'learning_rate'. 1.02, 'algorithm"n_estimators'. 10,. 'SAMME.R'}**	**194.8s**
5	Bayesian Optimization	0.94962440	{'learning_rate'. 0.82, 'n_estimators'. 8, 'algorithm'. 'SAMME'}	230.5s

https://doi.org/10.1371/journal.pone.0308015.t013

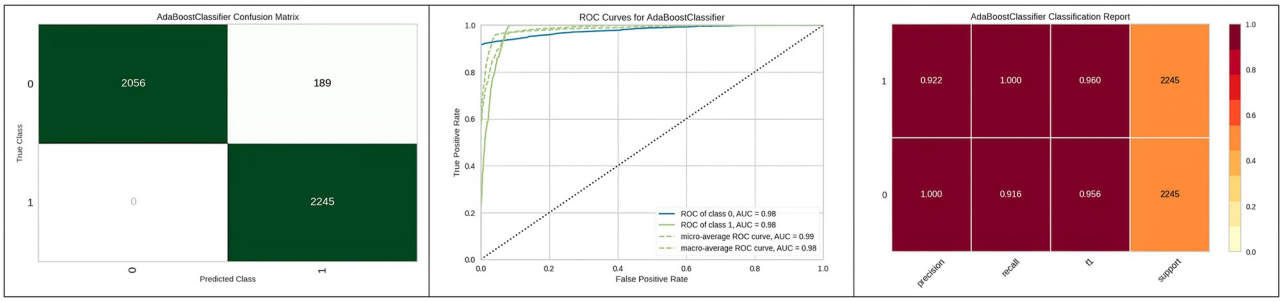

Fig 18. Best optimizer results of Adaboost.

https://doi.org/10.1371/journal.pone.0308015.g018

Table 14 represents the results of hyperparameter optimization on the Extra tree. Fig 19 represents the ROC curve and classification report representing the results of the best optimizer.

The findings of different hyperparameter optimization on light gradient boosting machine are compared to the results in Table 15, it's clearly show that Lightbgm model has improved slightly in performance with the grid search hyperparameter optimization algorithm.

Fig 20 represents the confusion matrix and ROC curve of the Lightbgm classifier with Grid search hyperparameter optimization.

Table 16 represents the results of Random forest with different hyperparameter tuning algorithms with different parameters of random forest. Furthermore, the results of a Random Forest with Bayesian Optimization, demonstrate high accuracy values.

Fig 21 represents the confusion matrix, ROC curve and classification report of Random forest with the Genetic Algorithm hyperparameter optimization.

Table 14. Results of Extra tree.

S.no	Hyperparameter optimization Algo	Accuracy	Optimized parameter	Computation Time
1	Default Hyperparameters	0.95143	{n_estimators = 100, max_depth = None, min_samples_split = 2, criterion = 'gini',}	160.5s
2	Grid Search	0.9562421	(min_samples_leaf = 10, n_estimators = 200, max_features = 3)	178.06s
3	Random Search	0.9579176	{'min_samples_leaf'. 5, 'max_features'. 4, 'n_estimators'. 100}	197.89s
4	**Genetic Algorithm**	**0.9664197**	**min_samples_leaf = 20, n_estimators = 100, max_features = 4**	**135.8s**
5	Bayesian Optimization	0.952421	min_samples_leaf = 10, n_estimators = 300, max_features = 3	260.5s

https://doi.org/10.1371/journal.pone.0308015.t014

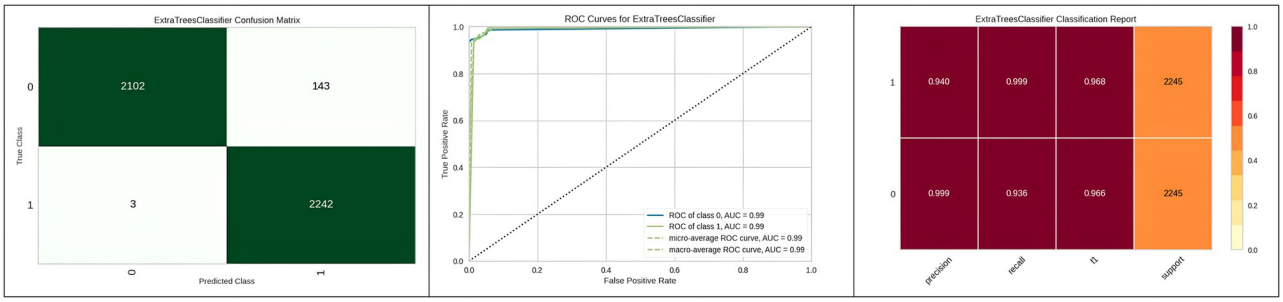

Fig 19. Best optimizer results of Extra tree.

https://doi.org/10.1371/journal.pone.0308015.g019

Table 15. Results of Lightbgm.

S. no	Hyperparameter optimization Algo	Accuracy	Optimized parameter	Computation Time
1	Default Hyperparameters	0.94951234	(learning_rate = 0.1, num_leaves = 31, max_depth = 1, n_estimators = 100, reg_lambda = 0.0, colsample_bytree = 1.0)	160.5s
2	**Grid Search**	**0.9564825**	**(colsample_bytree = 0.6, max_depth = 5, num_leaves = 5, reg_lambda = 1, learning_rate = 1, n_estimators = 20)**	**178.06s**
3	Random Search	0.9501229	{'max_depth'. 10, 'reg_lambda'. 0.1, 'num_leaves'. 50, 'n_estimators'. 20, 'colsample_bytree'. 1, 'boosting_type'. 'dart', 'learning_rate'.0.1}	197.89s
4	Genetic Algorithm	0.9490135	(learning_rate = 0.1,boosting_type = gbdt,colsample_bytree = 0.6, max_depth = 10, num_leaves = 50, reg_lambda = 0.1, n_estimators = 35)	135.8s
5	Bayesian Optimization	0.940296	(colsample_bytree = 0.6,, max_depth = 15, n_estimators = 35, num_leaves = 5, reg_lambda = 1, learning_rate = 1)	260.5s

https://doi.org/10.1371/journal.pone.0308015.t015

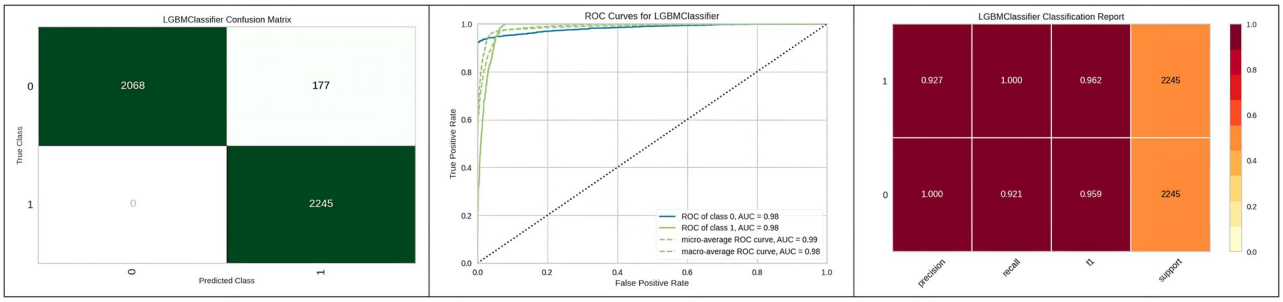

Fig 20. Best optimizer results of Lightbgm.

https://doi.org/10.1371/journal.pone.0308015.g020

Table 16. Results of Random Forest.

S. no	Hyperparameter optimization Algo	Accuracy	Optimized parameter	Computation Time
1	Default Hyperparameters	0.957521518	max_depth = 2, random_state = 0	10.5s
2	Grid Search	0.95905372	{'criterion'. 'entropy', 'max_depth'. 50, 'n_estimators'. 30}	6.06s
3	Random Search	0.9513258	{'min_samples_leaf'. 8, 'criterion'. 'gini', 'max_features'. 35, 'min_samples_split'. 9, 'n_estimators'. 66, 'max_depth'. 42}	7.89s
4	Genetic Algorithm	0.9589267	min_samples_split = 3,'criterion'. 'gini',max_depth = 17, min_sample s_leaf = 10, n_estimators = 120, max_features = 32	15.8s
5	**Bayesian Optimization**	**0.963746**	**{'min_samples_leaf'. 10.0, 'criterion'. 0, 'max_depth'. 43.0, 'min_samples_split'. 3.0, 'n_estimators'. 17.0',max_features'. 10.0}**	**30.5s**

https://doi.org/10.1371/journal.pone.0308015.t016

Table 17. Results of Decision tree.

S.no	Hyperparameter optimization Algo	Accuracy	Optimized parameter	Computation Time
1	Default Hyperparameters	0.9591079	{random_state = 5,max_depth = 12,criterion = 'ginni'}	80.5s
2	Grid Search	0.9445469	{random_state = 5,max_depth = 15,criterion = 'entropy'}	116.06s
3	Random Search	0.9574765	{random_state = 5,max_depth = 58,criterion = 'ginni'}	127.89s
4	**Genetic Algorithm**	**0.96884513**	**{random_state = 42,max_depth = 80,criterion = 'ginni'}**	**135.8s**
5	Bayesian Optimization	0.96084513	{random_state = 5,max_depth = 32,criterion = 'entropy'}	230.5s

https://doi.org/10.1371/journal.pone.0308015.t017

Table 18. Results of KNN.

S.no	Hyperparameter optimization Algo	Accuracy	Optimized parameter	Computation Time
1	Default Hyperparameters	0.86430099	{'n_neighbors'. 5}	60.5s
2	Grid Search	0.85604116	{'n_neighbors'. 15}	176.06s
3	Random Search	0.873094	{'n_neighbors'. 09}	147.89s
4	**Genetic Algorithm**	**0.8877094**	**KNeighborsClassifier__n_neighbors = 17**	**167.8s**
5	Bayesian Optimization	0.8655757	{'n_neighbors'. 8.0}	350.5s

https://doi.org/10.1371/journal.pone.0308015.t018

However, the efficacy of both the Decision tree and KNN models has significantly improved with genetic algorithm by 0.13% in Tables 17 and 18.

Fig 22 represents the confusion matrix and ROC curve of the Decision tree classifier with Genetic Algorithm hyperparameter optimization.

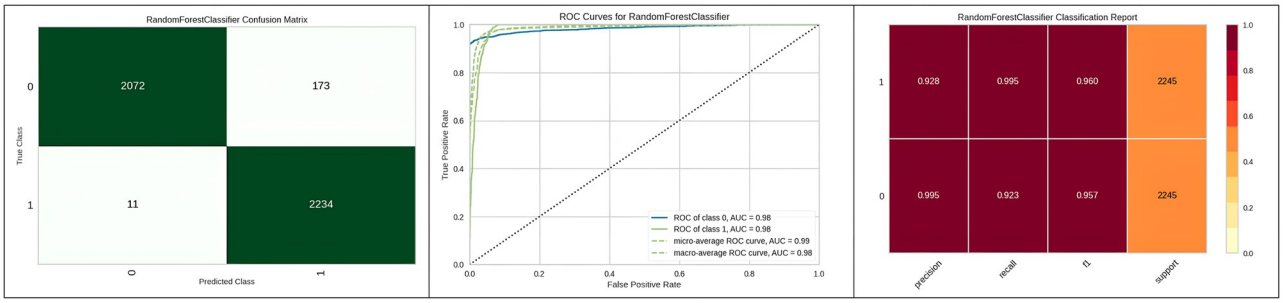

Fig 21. Best optimizer results of Random Forest.

https://doi.org/10.1371/journal.pone.0308015.g021

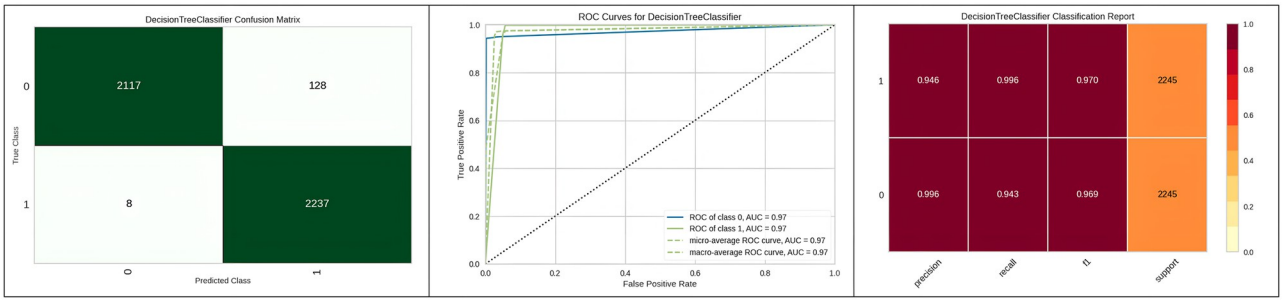

Fig 22. Best optimizer results of Decision tree.

https://doi.org/10.1371/journal.pone.0308015.g022

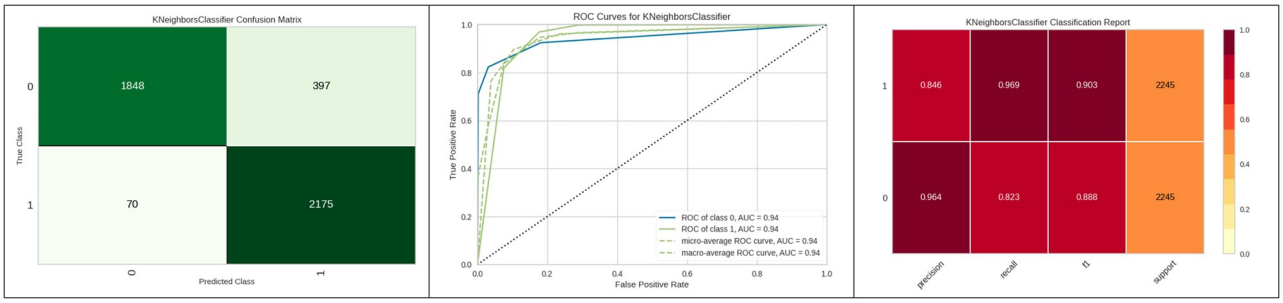

Fig 23. Best optimizer results of KNN.

https://doi.org/10.1371/journal.pone.0308015.g023

Furthermore, Table 18 represents the better performance of the KNN algorithm with genetic algorithm hyperparameter optimization. The results clearly indicate that the algorithm performed with 17 neighbors gives better results.

Fig 23 represents the confusion matrix of the model, ROC curve and classification report of KNN with genetic algorithm hyperparameter optimization.

Moving forward, using the top hyperparameters and a feature-selected dataset, an ensemble model is created. Evaluation and comparison of the best combination of machine learning algorithms compared with other machine learning algorithms. The ensemble model is created by finding the best combination of models with hyperparameter optimization algorithm parameters on the feature selection dataset. The Random forest, Adaboost and KNN model combination perform best as compared to other models. The accuracy of the ensemble model is 98% which is better than the hyperparameter optimized machine learning algorithm.

Fig 24 implies that the ensemble model is capable of distinguishing between positive and negative COVID-19 diagnoses. The findings show that the methods used to create the ensemble model result in more precise and reliable classification.

The ensemble model findings show that combining HPO-KNN, HPO-Random Forest, and HPO-Adaboost improves model performance when compared to other models in the trial. The results show that the feature selection method improves the model success rate on COVID-19 dataset records. As a result, the model trains have more qualified data, which improves efficiency. Furthermore, the use of HPO-KNN, HPO-Random Forest, and HPO-Adaboost improves model stability. Furthermore, because it incorporates predictions from various classification models, the fundamental blocks of the ensemble classification model produce a robust prediction.

Table 19. Comparison table of the proposed model.

Model	Accuracy	Recall	Precision	F-measure
Decision Tree Classifier	0.9593	0.9926	0.9484	0.9700
Extra Trees Classifier	0.9564	0.9935	0.9425	0.9673
Random Forest Classifier	0.9544	0.9962	0.9367	0.9655
Gradient Boosting Classifier	0.9489	0.9948	0.9283	0.9604
Light Gradient Boosting Machine	0.9484	0.9977	0.9250	0.9600
Ada Boost Classifier	0.9475	0.9877	0.9200	0.9583
K Neighbors Classifier	0.8300	0.8862	0.7967	0.8390
Ensemble model	**0.9804**	**0.9994**	**0.9865**	**0.9898**

https://doi.org/10.1371/journal.pone.0308015.t019

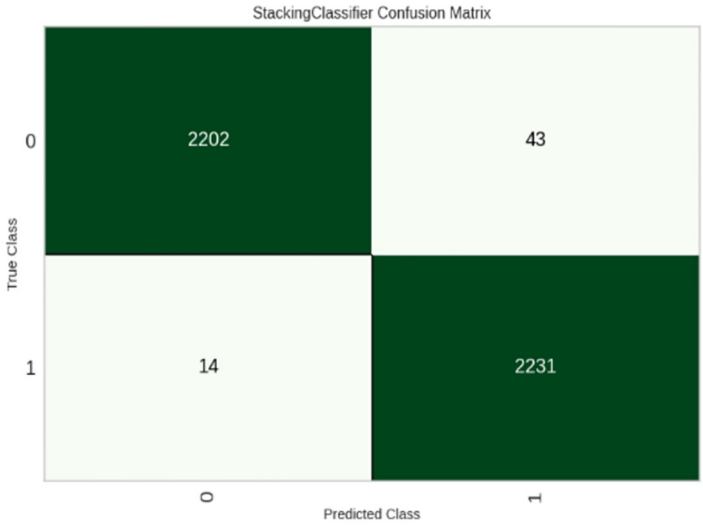

Fig 24. Confusion matrix of ensemble model.

https://doi.org/10.1371/journal.pone.0308015.g024

Table 19 represents the comparison of the Ensemble model with other machine learning algorithms. The results are compared with different evaluation metrics like accuracy, recall, precision and f-measure.

Fig 25 shows the statistical analysis of the proposed model compared with other machine learning models. The graph displays the sensitivity, specificity, positive likelihood, and negative likelihood of each model.

Fig 26 represents a graphical depiction of the ROC demonstrating that the ensemble methods takes the peak in terms of ACC, while the GBM stays at the bottom. Table 20 represents the comparison of related studies in predicting COVID-19 and the proposed model which clearly indicate maximum accuracy with an ensemble of three machine learning algorithms.

7. Feature importance using Explainable AI (SHAP analysis)

AI solutions were black box in nature, necessitating model explanation. If machine learning experts develop tools to comprehend and explain the models they constructed, non-technical people's doubts and suspicions are legitimate. SHAP [5] is one of the tools that was introduced a few years ago. It can deconstruct any machine learning model or deep neural net to make

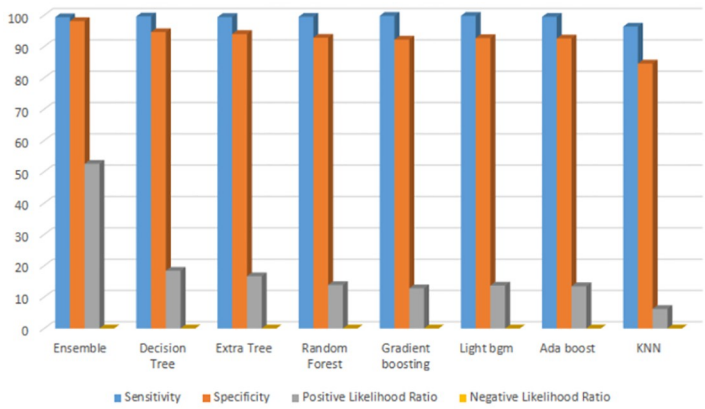

Fig 25. Statistical analysis of proposed model with other model.

https://doi.org/10.1371/journal.pone.0308015.g025

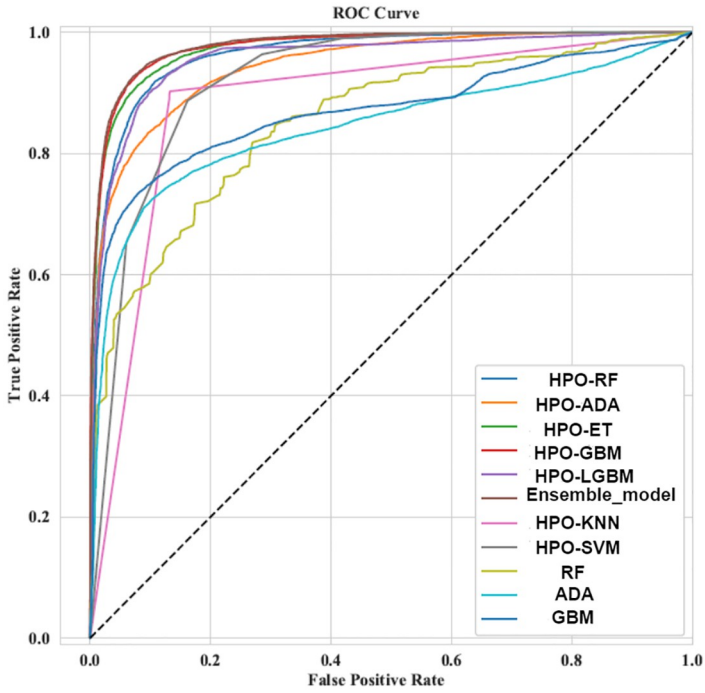

Fig 26. ROC comparison of machine learning algorithm.

https://doi.org/10.1371/journal.pone.0308015.g026

them intelligible to everyone. SHAP analysis explains what (and how) different factors impact your model's decisions. The significance of incoming characteristics in forecasting a target variable is represented by feature importance [52]. Most significantly, the listing of feature significance improves the predictive modeling project's efficacy and efficiency. In this research, we used the SHAP summary image [53].

Explainable AI (XAI) models and methods include Decision Trees, Logistic Regression, and Rule-Based Models for intrinsic interpretability, and post-hoc techniques like LIME, SHAP, and Grad-CAM for explaining complex models. Other approaches like Counterfactual Explanations and Partial Dependence Plots provide global and local insights, while advanced

Table 20. Comparison of related studies in predicting COVID-19 diagnosis.

Reference	Dataset Source	Model Used	Maximum Accuracy
Muhammad et al. [60]	263,007 patients from Mexico	Five models	95%
Han et al. [61]	375 patients from Wuhan	Broad Learning System	95%
Bruce Bode et al. [62]	398 patients from Texas	Many models	91%
Krishnaraj et. al. [63]	263,007 patients from Mexico	Five Ensemble Algorithm	96%
Aldonso et al. [64]	179,098 COVID-19	Many models	90%
Proposed Model	**14964 patients from Mexico**	**Three Algorithm Ensemble**	**98.04%**

https://doi.org/10.1371/journal.pone.0308015.t020

models such as Explainable Boosting Machines and Bayesian Rule [54] Lists combine transparency with predictive power. These XAI techniques enhance trust and accountability in AI by making their decisions understandable and transparent. As we can see, using the SHAP summary represented in Fig 26 has two advantages. feature ordering and the impact of each feature [55].

The feature ranking in decreasing sequence is determined by the location on the y-axis. (Higher importance to lower importance). X-axis SHAP values [56] decide the impact of each feature; positive SHAP values demonstrate a direct link with the target variable, and the opposite is also true. Additionally, the red shading corresponds to higher feature values, contrasting with the blue shading that stands for lower feature values. The irregular and intersecting lines suggest a sense of dispersion [57]. The importance of features for any categorization or forecast can be easily assessed by sorting the features in descending order, with the most important feature occupying the peak point. For example, as shown in Fig 23, visualizes in the form of a bar plot of the best 10 characteristics, with "INTUBADO "at the top. The following dominant characteristics are "INTUBADO", "EDAD", "ENTIDAD_RES", "ENTIDAD_NAC", "EMBARAZO", and so on. In comparison to the other characteristics depicted in the diagram, "DIABETES" stays hidden. Furthermore, as shown in Fig 23, increased "INTUBADO", "EDAD" and "ENTIDAD_RES" have a negative SHAP value, indicating a negative association.

Fig 27 represents the SHAP value of different features. It is obvious that greater values of this characteristic imply a lower probability of survival, i.e., in the case of COVID-19 here, and conversely as well. It's important to highlight that the overview image provides a top-down view of the data [47]. The reliance plot for SHAP values and the feature interaction plot for

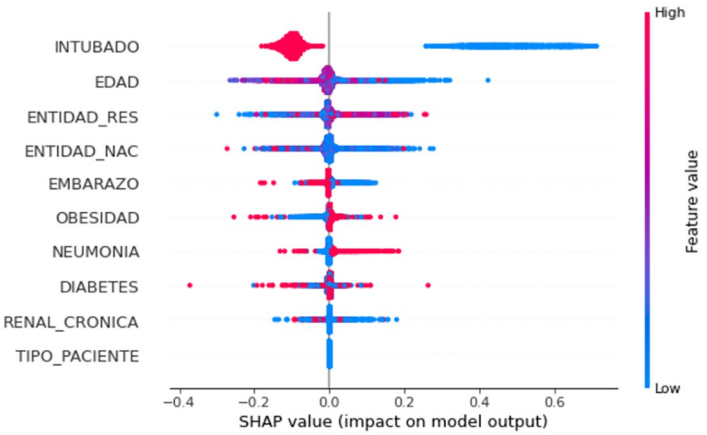

Fig 27. SHAP analysis.

https://doi.org/10.1371/journal.pone.0308015.g027

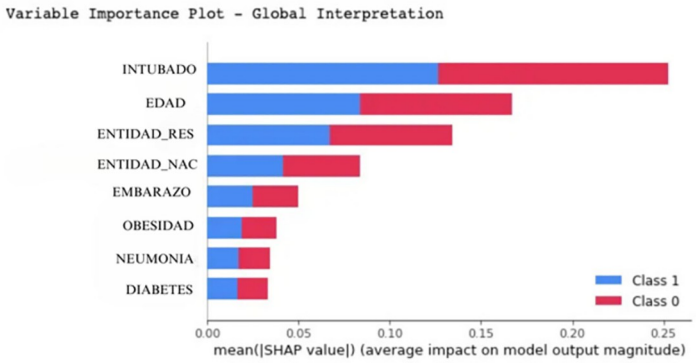

Fig 28. SHAP analysis mean value.

https://doi.org/10.1371/journal.pone.0308015.g028

SHAP values serve as tools to examine a particular feature and instance [48]. This assessment could determine how a sole feature influences the enhancement of model effectiveness, a matter not covered in this research, but reserved for future investigations.

The Fig 28 illustrate the impact of features on the model's predictions, distinguishing between Class 0 (COVID-19 positive) and Class 1 (COVID-19 negative). The graph highlights the significance of each variable with magnitude values, showcasing their importance in predicting COVID-19 outcomes [58]. This visualization effectively underscores the differential contribution of features in classifying COVID-19 status, facilitating a deeper understanding of the model's decision-making process [23, 59].

8. Conclusions and future works

This study aspires to propose an integrated machine learning model for respiratory disease prediction taking into consideration one of the most fatal diseases recently seen by the human race that is COVID-19. Seven contemporary machine learning classifiers have been coupled with different hyperparameter optimization techniques and a feature selection approach with an aim to enhance the prediction capability. The system's efficacy was rigorously evaluated through diverse performance metrics such as ACC, F1-score, MCC, and Kappa index, offering valuable insights from both patient and clinician viewpoints. The incorporation of SHAP values facilitates a comprehensive analysis of prediction outcomes for the observations. This technique of ranking input variables for identifying positive COVID-19 results helps to interpret the justification behind the model's classification decisions. Furthermore, the proposed model can be readily extended to predict other ailments like diabetes, asthma, and hypertension. In summary, this study not only contributes to the realm of respiratory disease prediction but also lays the foundation for broader applications in disease forecasting. The seamless amalgamation of machine learning techniques, clinical datasets, and optimization strategies offers a holistic approach that has the potential to revolutionize healthcare analytics. Although this study is limited by the absence of experiments involving datasets with missing values and the evaluation of model performance on big data. Future work could explore the development of the model into an application integrated with Internet of Things (IoT) technologies. Furthermore, the work can be extended to utilize deep learning models for the extraction of features from the images and using standard machine learning techniques for classification while considering various evolutionary algorithms for feature selection.

Acknowledgments

Thanks to all authors for completing this work properly.

Author Contributions

Conceptualization: Balraj Preet Kaur, Sanjeev Kumar Sharma.

Data curation: Balraj Preet Kaur.

Formal analysis: Balraj Preet Kaur.

Funding acquisition: Chetna Sharma.

Investigation: Chetna Sharma.

Methodology: Harpreet Singh, Chetna Sharma.

Project administration: Md. Mehedi Hassan.

Resources: Harpreet Singh, Rahul Hans.

Software: Harpreet Singh, Rahul Hans.

Supervision: Md. Mehedi Hassan.

Validation: Rahul Hans.

Visualization: Sanjeev Kumar Sharma.

Writing – original draft: Md. Mehedi Hassan.

Writing – review & editing: Balraj Preet Kaur, Chetna Sharma.

References

1. https://covid19.who.int/table/ (accessed December 8, 2023)
2. Mansbridge N. et al., "Feature selection and comparison of machine learning algorithms in classification of grazing and rumination behaviour in sheep," *Sensors (Switzerland)*, vol. 18, no. 10, pp. 1–16, 2018. https://doi.org/10.3390/s18103532 PMID: 30347653
3. Uddin S., Khan A., Hossain M. E., and Moni M. A., "Comparing different supervised machine learning algorithms for disease prediction," *BMC Med. Inform. Decis. Mak.*, vol. 19, no. 1, pp. 1–16, 2019.
4. Ernawan F., Handayani K., Fakhreldin M., and Abbker Y., "Light Gradient Boosting with Hyper Parameter Tuning Optimization for COVID-19 Prediction," *Int. J. Adv. Comput. Sci. Appl.*, vol. 13, no. 8, pp. 514–523, 2022.
5. Muhammad L. J. et al. 2021. "Supervised Machine Learning Models for Prediction of COVID-19 Infection Using Epidemiology Dataset." *SN Computer Science* 2(1). 1–13. https://doi.org/10.1007/s42979-020-00394-7 PMID: 33263111
6. Sharma Ajay, and Pramod Kumar Mishra. 2022. "Performance Analysis of Machine Learning Based Optimized Feature Selection Approaches for Breast Cancer Diagnosis." *International Journal of Information Technology (Singapore)* 14(4). 1949–60. https://doi.org/https://doi.org/10.1007/s41870-021-00671-5
7. Sevinç E., "An empowered AdaBoost algorithm implementation. A COVID-19 dataset study," *Comput. Ind. Eng.*, vol. 165, no. December 2021, p. 107912, 2022.
8. An T. K. and Kim M. H., "A new Diverse AdaBoost classifier," *Proc.—Int. Conf. Artif. Intell. Comput. Intell. AICI* 2010, vol. 1, pp. 359–363, 2010.
9. Sayed S. A. F., Elkorany A. M., and Sayed Mohammad S., "Applying Different Machine Learning Techniques for Prediction of COVID-19 Severity," *IEEE Access*, vol. 9, pp. 135697–135707, 2021. https://doi.org/10.1109/ACCESS.2021.3116067 PMID: 34786321
10. Chowdhury N. K., Kabir M. A., Rahman M. M., and Islam S. M. S., "Machine learning for detecting COVID-19 from cough sounds. An ensemble-based MCDM method," *Comput. Biol. Med.*, vol. 145, no. November 2021, p. 105405, 2022.

11. Zargari Khuzani A., Heidari M., and Shariati S. A., "COVID-Classifier. an automated machine learning model to assist in the diagnosis of COVID-19 infection in chest X-ray images," *Sci. Rep.*, vol. 11, no. 1, pp. 1–6, 2021.

12. Sreedharan R. and Kumar A. P., "Analysis and prediction of smart data using machine learning," *AIP Conf. Proc.*, vol. 2240, no. Ml, pp. 15–21, 2020.

13. Hu P., Pan J. S., and Chu S. C., "Improved Binary Grey Wolf Optimizer and Its application for feature selection," *Knowledge-Based Syst.*, vol. 195, no. xxxx, p. 105746, 2020.

14. Emary E., Zawbaa H. M., and Hassanien A. E., "Binary grey wolf optimization approaches for feature selection," *Neurocomputing*, vol. 172, pp. 371–381, 2016.

15. Ciotti M., Ciccozzi M., Terrinoni A., Jiang W. C., Bin Wang C., and Bernardini S., "The COVID-19 pandemic," *Crit. Rev. Clin. Lab. Sci.*, vol. 0, no. 0, pp. 365–388, 2020. https://doi.org/10.1080/10408363.2020.1783198 PMID: 32645276

16. Velavan T. P. and Meyer C. G., "The COVID-19 epidemic," *Trop. Med. Int. Heal.*, vol. 25, no. 3, pp. 278–280, 2020. https://doi.org/10.1111/tmi.13383 PMID: 32052514

17. Alali Y., Harrou F., and Sun Y., "A proficient approach to forecast COVID-19 spread via optimized dynamic machine learning models," *Sci. Rep.*, vol. 12, no. 1, pp. 1–20, 2022.

18. Yang L. and Shami A., "On hyperparameter optimization of machine learning algorithms. Theory and practice," *Neurocomputing*, vol. 415, pp. 295–316, 2020.

19. Debjit K. et al., "An Improved Machine-Learning Approach for COVID-19 Prediction Using Harris Hawks Optimization and Feature Analysis Using SHAP," *Diagnostics*, vol. 12, no. 5, 2022. https://doi.org/10.3390/diagnostics12051023 PMID: 35626179

20. Shahhosseini M., Hu G., and Pham H., "Optimizing ensemble weights and hyperparameters of machine learning models for regression problems," *Mach. Learn. with Appl.*, vol. 7, no. December 2021, p. 100251, 2022.

21. Mohana Saranya S., Rajalaxmi R. R., Mohanapriya S., Prasida S., and Nithyalaxmi P., "Prediction of Covid-19 Using Hyperparameter Optimized Convolutional Neural Network," *Turkish J. Comput. Math. Educ.*, vol. 12, no. 9, pp. 448–455, 2021.

22. S. Hamida, O. E. L. Gannour, B. Cherradi, H. Ouajji, and A. Raihani, "Optimization of machine learning algorithms hyper-parameters for improving the prediction of patients infected with COVID-19," *2020 IEEE 2nd Int. Conf. Electron. Control. Optim. Comput. Sci. ICECOCS 2020*, no. 1, 2020.

23. Aljouie Abdulrhman Fahad et al. 2021. "Early Prediction of COVID-19 Ventilation Requirement and Mortality from Routinely Collected Baseline Chest Radiographs, Laboratory, and Clinical Data with Machine Learning." *Journal of Multidisciplinary Healthcare* 14. 2017–33. https://doi.org/10.2147/JMDH.S322431 PMID: 34354361

24. Pourhomayoun M. and Shakibi M., "Predicting mortality risk in patients with COVID-19 using machine learning to help medical decision-making," *Smart Heal.*, vol. 20, no. April 2020, p. 100178, 2021.

25. Attallah Omneya. 2022. "An Intelligent ECG-Based Tool for Diagnosing COVID-19 via Ensemble Deep Learning Techniques." *Biosensors* 12(5). https://doi.org/10.3390/bios12050299 PMID: 35624600

26. Rostami Mehrdad, and Oussalah Mourad. 2022. "A Novel Explainable COVID-19 Diagnosis Method by Integration of Feature Selection with Random Forest." *Informatics in Medicine Unlocked* 30(January). 100941. https://doi.org/10.1016/j.imu.2022.100941 PMID: 35399333

27. Ozyurt Fatih, Tuncer Turker, and Subasi Abdulhamit. 2021. "An Automated COVID-19 Detection Based on Fused Dynamic Exemplar Pyramid Feature Extraction and Hybrid Feature Selection Using Deep Learning." *Computers in Biology and Medicine* 132(March). 104356. https://doi.org/10.1016/j.compbiomed.2021.104356 PMID: 33799219

28. Chattopadhyay Soham et al. 2021. "Covid-19 Detection by Optimizing Deep Residual Features with Improved Clustering-Based Golden Ratio Optimizer." *Diagnostics* 11(2). 1–27. https://doi.org/10.3390/diagnostics11020315 PMID: 33671992

29. El-Kenawy El Sayed M. et al. 2020. "Novel Feature Selection and Voting Classifier Algorithms for COVID-19 Classification in CT Images." *IEEE Access* 8. https://doi.org/10.1109/ACCESS.2020.3028012 PMID: 34976558

30. Pramanik Rishav, Sarkar Sourodip, and Sarkar Ram. 2022. "An Adaptive and Altruistic PSO-Based Deep Feature Selection Method for Pneumonia Detection from Chest X-Rays." *Applied Soft Computing* 128. 1–23. https://doi.org/10.1016/j.asoc.2022.109464 PMID: 35966452

31. Yagin Fatma Hilal et al. 2023. "Explainable Artificial Intelligence Model for Identifying COVID-19 Gene Biomarkers." *Computers in Biology and Medicine* 154(November 2022). https://doi.org/10.1016/j.compbiomed.2023.106619 PMID: 36738712

32. Hamal Susmita et al. 2024. "A Comparative Analysis of Machine Learning Algorithms for Detecting COVID-19 Using Lung X-Ray Images." *Decision Analytics Journal* 11(June 2023). 100460. https://doi.org/10.1016/j.dajour.2024.100460

33. Héberger Károly. 2024. "Frequent Errors in Modeling by Machine Learning. A Prototype Case of Predicting the Timely Evolution of COVID-19 Pandemic." *Algorithms* 17(1).

34. Dewi K. C., Mustika W. F., & Murfi H. (2019, March). "Ensemble learning for predicting mortality rates affected by air quality". In Journal of physics. Conference series (vol. 1192, No. 1, p. 012021). IOP Publishing.

35. de Moraes Batista A. F., Miraglia J. L., Rizzi Donato T. H., & Porto Chiavegatto Filho A. D. (2020). "COVID-19 diagnosis prediction in emergency care patients. a machine learning approach". *MedRxiv*, 2020–04.

36. Kukar M., Gunčar G., Vovko T. et al. "COVID-19 diagnosis by routine blood tests using machine learning". *Sci Rep* 11, 10738 (2021). https://doi.org/10.1038/s41598-021-90265-9 PMID: 34031483

37. Kassania S. H., Kassanib P. H., Wesolowskic M. J., Schneidera K. A., and Detersa R., "Automatic Detection of Coronavirus Disease (COVID-19) in X-ray and CT Images. A Machine Learning Based Approach," *Biocybern. Biomed. Eng.*, vol. 41, no. 3, pp. 867–879, 2021.

38. Adimoolam M., Govindharaju K., John A., Mohan S., Ahmadian A., and Ciano T., "A hybrid learning approach for the stage-wise classification and prediction of COVID-19 X-ray images," *Expert Syst.*, vol. 39, no. 4, 2022.

39. Abayomi-Alli O. O., Damaševičius R., Maskeliūnas R., and Misra S., "An Ensemble Learning Model for COVID-19 Detection from Blood Test Samples," *Sensors*, vol. 22, no. 6, 2022. https://doi.org/10.3390/s22062224 PMID: 35336395

40. Sagi O. and Rokach L., "Ensemble learning. A survey," *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 8, no. 4, pp. 1–18, 2018.

41. Ndwandwe D. and Wiysonge C. S., "COVID-19 vaccines," *Curr. Opin. Immunol.*, vol. 71, no. 1, pp. 111–116, 2021. https://doi.org/10.1016/j.coi.2021.07.003 PMID: 34330017

42. McCoy D., Mgbara W., Horvitz N., Getz W. M., and Hubbard A., "Ensemble machine learning of factors influencing COVID-19 across US counties," *Sci. Rep.*, vol. 11, no. 1, pp. 1–14, 2021.

43. AlJame M., Ahmad I., Imtiaz A., and Mohammed A., "Ensemble learning model for diagnosing COVID-19 from routine blood tests," *Informatics Med. Unlocked*, vol. 21, p. 100449, 2020. https://doi.org/10.1016/j.imu.2020.100449 PMID: 33102686

44. R. Shaaque, A. Mehmood, G. S. Choi, R. Shafique, and S. Ullah, "Cardiovascular Disease Prediction System Using Extra Trees Classiier Cardiovascular Disease Prediction System Using Extra Trees Classifier," 2019.

45. Shrivastav L. K. and Jha S. K., "A gradient boosting machine learning approach in modeling the impact of temperature and humidity on the transmission rate of COVID-19 in India," *Appl. Intell.*, vol. 51, no. 5, pp. 2727–2739, 2021. https://doi.org/10.1007/s10489-020-01997-6 PMID: 34764559

46. S. Tripath, "Gradient-Boosting Machine Model," pp. 19–21.

47. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," *Adv. Neural Inf. Process. Syst. 24 25th Annu. Conf. Neural Inf. Process. Syst. 2011, NIPS 2011*, pp. 1–9, 2011.

48. Xia X., Jiang S., Zhou N., Li X., and Wang L., "Genetic algorithm hyper-parameter optimization using taguchi design for groundwater pollution source identification," *Water Sci. Technol. Water Supply*, vol. 19, no. 1, pp. 137–146, 2019.

49. www.kaggle.com/marianarfranklin/mexico-covid19-clinical-data/

50. Thapa, Surendrabikram, Surabhi Adhikari, Awishkar Ghimire, and Anshuman Aditya. 2020. "Feature Selection Based Twin-Support Vector Machine for the Diagnosis of Parkinson's Disease." *IEEE Region 10 Humanitarian Technology Conference*, *R10-HTC* 2020-December(December).

51. Xiong Yibai et al. 2022. "Comparing Different Machine Learning Techniques for Predicting COVID-19 Severity." *Infectious Diseases of Poverty* 11(1). 1–9. https://doi.org/10.1186/s40249-022-00946-4

52. D. Devetyarov, I. Nouretdinov, C. Based, and R. Forest, "Prediction with Confidence Based on a Random Forest Classifier To cite this version. HAL Id. hal-01060649 Prediction with Confidence Based on a Random Forest Classifier," pp. 0–8, 2017.

53. Imam A. T., Alhroob A., and Alzyadat W. J., "SVM Machine Learning Classifier to Automate the Extraction of SRS Elements," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 3, pp. 174–185, 2021.

54. D. A. Pisner and D. M. Schnyer, *Support vector machine*. Elsevier Inc., 2019.

55. Rai N., Kaushik N., Kumar D., Raj C., and Ali A., "Mortality prediction of COVID-19 patients using soft voting classifier," *Int. J. Cogn. Comput. Eng.*, vol. 3, no. June, pp. 172–179, 2022.

56. Florea A. C. and Andonie R., "Weighted Random Search for hyperparameter optimization," *Int. J. Comput. Commun. Control*, vol. 14, no. 2, pp. 154–169, 2019.

57. Haqmi Abas M. A., "Agarwood Oil Quality Classification using Support Vector Classifier and Grid Search Cross Validation Hyperparameter Tuning," *Int. J. Emerg. Trends Eng. Res.*, vol. 8, no. 6, pp. 2551–2556, 2020.

58. Ali Yasser A., Emad Mahrous Awwad Muna Al-Razgan, and Maarouf Ali. 2023. "Hyperparameter Search for Machine Learning Algorithms for Optimizing the Computational Complexity." *Processes* 11 (2).

59. Chieregato Matteo et al. 2022. "A Hybrid Machine Learning/Deep Learning COVID-19 Severity Predictive Model from CT Images and Clinical Data." *Scientific Reports* 12(1). 1–15. https://doi.org/10.1038/s41598-022-07890-1

60. Muhammad L. J., Algehyne E. A., Usman S. S., Ahmad A., Chakraborty C., & Mohammed I. A. (2021). Supervised machine learning models for prediction of COVID-19 infection using epidemiology dataset. SN computer science, 2(1), 1–13. https://doi.org/10.1007/s42979-020-00394-7 PMID: 33263111

61. Han X., Hu Z., Wang S., & Zhang Y. (2022). A survey on deep learning in COVID-19 diagnosis. Journal of imaging, 9(1), 1. https://doi.org/10.3390/jimaging9010001 PMID: 36662099

62. Bode B., Garrett V., Messler J., McFarland R., Crowe J., Booth R., & Klonoff D. C. (2020). Glycemic characteristics and clinical outcomes of COVID-19 patients hospitalized in the United States. Journal of diabetes science and technology, 14(4), 813–821. https://doi.org/10.1177/1932296820924469 PMID: 32389027

63. Chadaga K., Prabhu S., Umakanth S., Bhat K., Sampathila N., & Chadaga R. (2021). COVID-19 mortality prediction among patients using epidemiological parameters: an ensemble machine learning approach. Engineered Science, 16(10), 221–233.

64. Becerra-Sánchez A., Rodarte-Rodríguez A., Escalante-García N. I., Olvera-González J. E., De la Rosa-Vargas J. I., Zepeda-Valles G., et al. (2022). Mortality analysis of patients with COVID-19 in Mexico based on risk factors applying machine learning techniques. Diagnostics, 12(6), 1396. https://doi.org/10.3390/diagnostics12061396 PMID: 35741207