Service Oriented Computing and Applications (2024) 18:163–182
https://doi.org/10.1007/s11761-023-00382-8

SPECIAL ISSUE PAPER

Metaheuristic-based hyperparameter optimization for multi-disease detection and diagnosis in machine learning

Jagandeep Singh[1] · Jasminder Kaur Sandhu[1] · Yogesh Kumar[2]

Received: 3 March 2023 / Revised: 25 November 2023 / Accepted: 26 December 2023 / Published online: 23 January 2024

Abstract

Metaheuristic algorithms with machine learning techniques have become popular because it works so well for problems like regression, classification, rule mining, and clustering in health care. This paper's primary purpose is to design a multi-disease prediction system using AI-based metaheuristic approaches. Initially, the data is collected in the form of diverse classes, which include Id, gender, date of birth, etc. The data has been preprocessed, normalized, and graphically represented to improve its quality and detect any errors. Later, machine learning models, such as decision tree, extra tree classifier, extreme gradient boosting classifier, light gradient boosting machine classifier, random forest, and artificial neural network, are initially trained without optimizing hyperparameters and then fine-tuned by integrating various hyperparameter optimizers such as grid search CV, random search, hyperband, and genetic search. During experimentation, it is found that optimizing the models using random search optimizer computed the highest accuracy of 100% as compared to the rest of the hyperparameter optimizers. In the context of 'Service Oriented Computing and Applications,' our multi-disease prediction system offers valuable innovation. It aligns with the goal of enhancing healthcare services, patient outcomes, and healthcare efficiency. Our pioneering integration of metaheuristic algorithms and machine learning introduces intelligent healthcare solutions, with the study's focus on hyperparameter optimization and achieving 100% accuracy demonstrates practical significance in SOC and its applications.

Keywords Metaheuristics · Medical data · Hyperparameters · Random search · Machine learning · Artificial neural network

1 Introduction

There are three main ways to handle combinatorial-based problems, one of which way is examining the space of multiple combinations in such a way that either we can find a solution or it will be proven that the solution is inconsistent. Pruning techniques and heuristics have not always been able to restrain such problems because of either higher computational time or deliberately ignoring some combinations [1]. As a result, we are not able to find the optimal solution, and also using these techniques, we would not be able to prove whether the optimality of the combination which is found is actually optimal or not. In fact, these problems are even more complex for computer scientists because solving them requires a huge number of combinations. Hence to solve such challenging and complex combinatorial issues, there are some generic methods which are known as metaheuristics [2]. A metaheuristic is a combination of two words, i.e., 'meta' as well as 'heuristics,' in which meta means beyond and heuristic means search. This advanced heuristic algorithm uses trial and error to find the optimal solution at advanced levels [3]. Moreover, metaheuristic algorithms are used for solving the problem to provide a general solution with genuine accuracy and speed without even having the information about the problem. Metaheuristics are a family of various algorithms designed to solve a high and complex number of combinatorial problems without even having to adapt them deeply to each problem. In a true sense, they also

✉ Yogesh Kumar
yogesh.arora10744@gmail.com

Jagandeep Singh
jagansidhu@gmail.com

Jasminder Kaur Sandhu
dr.jasminder.sandhu@gmail.com

[1] Department of CSE, Chandigarh University, Gharuan, Mohali, India

[2] Department of CSE, School of Technology, Pandit Deendayal Energy University, Gandhinagar, Gujarat, India

do not guarantee optimal solutions but try to provide them in a considerable amount of time [4].

Metaheuristic techniques have been successfully applied to machine and deep learning techniques for speeding up the training and testing process. In fact, in these years, considerable growth has been found in integrating AI-based techniques with metaheuristics to solve combinatorial optimization-based problems [5]. It has also been found that combining metaheuristics with AI leads to an effective, efficient, and robust search which improves the execution of the models in terms of their quality of the solution, robustness, and convergence rate [6]. Machine learning, the subset of artificial intelligence, learns from a huge dataset to make correct predictions. It is broadly classified into three types of learning, i.e., supervised, unsupervised, and semi-supervised. Supervised learning works on the set of labeled data where the goal or aim is the output prediction in the form of a response variable from the explanatory input variables [7]. This type of learning is majorly used in classification, regression, reducing dimensions, predicting time series, and reinforcement learning. Likewise, in the case of unsupervised learning, the models are being trained with the unlabeled set of data which is used to obtain compact descriptions of the data and does not include any response variable as an output [8]. Unsupervised learning had various tasks such as detecting anomalies, clustering, categorizing, modeling time series, etc. Semi-supervised learning is similar to supervised learning, but here all the input data do not have an associated output value. This type of learning is mainly used in those problems in which the models are being trained with a huge set of unlabeled samples, and only a few are labeled manually [9]. On the other hand, the subset of machine learning, i.e., deep learning uses multiple layers to process complex structure data. There are many deep learning techniques such as recurrent neural network, convolutional neural network, restricted Boltzmann machines, deep Boltzmann machines, auto-encoders, and deep belief networks [10]. Deep learning plays a vital role in those research areas which have to process large and big data for recognizing patterns, processing natural languages, classifying images, etc. In deep learning, the algorithms learn the raw data layer by layer so that the data can be transformed from raw feature space to transformed feature space [11]. In addition, deep learning algorithms also work on nonlinear functions and are perfect for classification. In fact, the convolutional neural network in deep learning is the most highly used, particularly for image detection and classification [12].

Although machine and deep learning have been widely used by the research and development industry and have outperformed in their respective areas, they also have specific areas for improvement, such as being time-consuming and space-consuming [13]. Hence, in this paper, to overcome the drawback of AI-based techniques, the main aim of the research is to develop a metaheuristic-based disease prediction system using machine learning models. Besides this, we also want to examine the performance of models with and without applying metaheuristic optimizers based on their F1 score, recall, accuracy, and precision.

Within the context of ‘Service Oriented Computing and Applications (SOCA),’ our multi-disease prediction system offers significant value. Our research directly aligns with this goal by introducing an innovative AI-driven approach that can enhance healthcare services, improve patient outcomes, and streamline healthcare processes. The integration of metaheuristic algorithms with machine learning techniques represents a groundbreaking contribution, creating new possibilities for intelligent and adaptable service-oriented healthcare solutions. Moreover, our study’s focus on optimizing hyperparameters and achieving the great accuracy highlights the practical applicability and effectiveness of our approach in the SOC field and its applications.

1.1 Contribution

The following contributions have been made to the completion of this research:

- Initially, the data is collected from medical dataset in which there are 14 attributes such as id, gender, date of birth, education, employment status, children, marital status, and disease. Besides this, 12 classes of diseases from the same dataset have been taken, such as hypertension, Alzheimer’s disease, multiple sclerosis, endometriosis, prostate cancer, heart disease, HIV/AIDS, gastric, skin cancer, kidney disease, breast cancer, and schizophrenia to train the models.
- The dataset is preprocessed in the second phase to identify NAN or missing values. Encoders are applied to the selected attribute to normalize the values.
- In the third phase, the data is graphically visualized to detect the anomaly as well as to understand it in a better way. The exploratory data analysis of various categories of the dataset has been generated and described, such as the number of males and females, the count of people having the aforementioned kind of diseases, the number of people either in or not in the military service, etc.
- In the fourth phase, various hyperparameter optimizers, like grid search CV, random search, hyperband, and genetic optimizer on integrating with the machine learning models, are applied.
- The last phase evaluates models based on performance parameters like precision, accuracy, recall, and F1 score for both combined and distinct dataset classes.

1.2 Paper organization

Section 1: brief description of the metaheuristic, machine, and deep learning models, the need for metaheuristics in these models along with the aim of this research as well as the contribution made.

Section 2: information about the work done by the researchers who have applied metaheuristic optimizers to improve AI-based techniques as well as for detecting various diseases.

Section 3: defines the research methodology, which covers the description of the dataset, preprocessing of the data, exploratory data analysis, applied hyperparameter optimizers, and machine learning models, along with the performance metrics.

Section 4: applied models are examined and analyzed.

Section 5: conclusion of the whole paper.

2 Background

This section is split into two parts. The first shows how experts have used metaheuristic techniques to improve AI-based learning models, and the second shows how these techniques have been used in the field of health care.

2.1 Metaheuristics for optimizing machine learning algorithms

Faris et al. [14] introduced the use of the multi-verse optimizer (MVO), a nature-inspired algorithm, to train multilayer perceptron neural networks. The novel method was evaluated on nine diverse biomedical datasets sourced from the UCI machine learning repository and was compared with contemporary evolutionary metaheuristic algorithms. Aljarah et al. [15] introduced the grasshopper optimization algorithm, inspired by swarm behavior as a novel work, to optimize support vector machine (SVM) parameters and simultaneously selected the best feature subset. The method's effectiveness was assessed across eighteen benchmark datasets with varying dimensions. Comparison was made against seven established algorithms and the popular grid search technique for SVM parameter tuning. Experimental results revealed their proposed approach performed well in terms of classification. Tao et al. [16] presented a novel approach using genetic algorithm to enhance the predicting accuracy of a hospitalization expense model through parameter optimization as well as feature selection of support vector machine (SVM). Hospitalization expense data had been preprocessed, clustered using k-means, and transformed into a chromosome with kernel function, kernel penalty factor, and feature mask. A fitness function combining classification accuracy and feature count guided GA in optimizing SVM parameters and feature subset selection. The algorithm was compared to GA-PCA and PSO-PCA in single-parameter optimization, demonstrating its efficiency in achieving improved classification results by swiftly identifying suitable feature subsets and SVM parameters. In the same way, Faris et al. [17] discussed about a new hybrid encoding scheme and used well-known stochastic population-based methods for optimizing the number of hidden neurons as well as connection weights in a single hidden-layer feedforward neural network. Through experiments conducted on twenty-three standard classification datasets, their proposed technique was both qualitatively and quantitatively benchmarked. Their findings indicated that the hybrid encoding scheme facilitates efficient optimization of both hidden node count and connection weights by various optimization algorithms.

Mirjalili [18] introduced the application of the gray wolf optimizer to train multilayer perceptrons (MLPs) for the first time. Their model was evaluated on eight different datasets, comprising both classification and function approximation tasks. The performance of GWO was compared to several established evolutionary training algorithms, namely evolution strategy, particle swarm optimization, genetic algorithm, and population-based incremental learning. Their experimental results highlighted that the GWO algorithm exhibited competitive outcomes in terms of avoiding local optima, which demonstrated its efficacy in achieving improved solutions. Amirsadri et al. [19] introduced a novel hybrid algorithm for training neural networks that combined a gradient-based approach with a metaheuristic method. Their hybrid algorithm capitalized on both local and global search strategies, overcoming issues related to local optima. The algorithm enhanced the global search capability of the GWO by incorporating Levy flight, a random walk with jumps following the Levy distribution, resulting in more effective exploration of the search space. The improved algorithm was then merged with backpropagation (BP) to leverage the strengths of GWO's enhanced global search and BP's local search in neural network training. The researchers assessed the performance of their algorithm through a comparison with established metaheuristic algorithms across twelve datasets involving classification and function approximation tasks.

2.2 Metaheuristics for feature selection or optimization of machine learning algorithms in the medical domain

Hu and Razmjooy [20] developed a metaheuristic-based system to detect brain tumors in their early stages. The researchers initially segmented the tumor and then extracted features from it. Later, deep belief network integrated with the seagull optimization was used for the classification. The results were also compared with their existing ones in which the particular method obtained the highest Performance in accuracy, which was 88%. Eshtay et al. [21] optimized extreme learning machine input weights as well as hidden neurons using competitive swarm optimizer (CSO). The study considered both classical ELM and its regularized variant, aiming to enhance generalization performance, stabilize classification, and create more compact networks by reducing hidden-layer neurons. The proposed method was evaluated across 15 medical classification tasks. Their results revealed that the approach achieved improved generalization performance, reduced hidden neuron count, and greater stability. Likewise, Shankar et al. [22] integrated the ant lion optimization model with the deep neural network to develop the system for predicting chronic kidney disease. The metaheuristic optimizer was taken to choose the optimal features to enhance the classification process. Their developed system computed accuracy at 96.63%, sensitivity at 98.22%, and specificity at 91.22%. Chitradevi et al. [23] had applied optimization techniques such as LOA, BAT, artificial bee colony, and particle swarm optimization for segmenting brain regions to identify Alzheimer's disease. Their proposed model obtained 95% accuracy, which was best compared to the others. The overall work of their proposed model was to demonstrate the abnormalities inside the brain, which provided a reliable and accurate indication to the clinician about the progression of Alzheimer's disease. Canayaz [24] presented a deep learning-based methodology for early disease diagnosis. The research was conducted on the 1092 X-ray images of lungs having three classes like normal, COVID-19, and pneumonia. Initial data underwent image contrast enhancement preprocessing, which resulted in a new dataset. Deep learning models, like AlexNet, ResNet, VGG19, and GoogleNet, were utilized to extract features from this enhanced dataset. To identify the most effective features, binary gray wolf optimization as well as binary particle swarm optimization was employed. The selected features were then combined and subjected to classification using support vector machines. The approach 99.38% accurately diagnoses the disease at an early stage. Roostaee and Ghaffary [25] had worked on a dataset of 303 people having 14 features in it. The work was conducted using binary cuckoo optimization along with the support vector machine for detecting and diagnosing heart diseases. The features were selected using a metaheuristic technique to obtain the optimal features and were classified using a machine learning model. Nadimi-Shahraki et al. [26] worked on an enhanced version of the whale optimization algorithm called E-WOA. Its performance was assessed and compared to prominent WOA variants for global optimization challenges, demonstrating its superiority. Building on E-WOA's success, a binary variant, BE-WOA, was proposed for effective feature selection, especially in medical datasets. BE-WOA's efficacy was validated on the dataset of medical diseases and distinguished with leading optimization algorithms using fitness, accuracy, sensitivity, precision, and feature count as criteria. Rashid et al. [27] focused on four different metaheuristic algorithm-based optimizers—gray wolf optimizer, harmony search, sine–cosine algorithm, as well as ant lion optimization—to effectively train and structure the long short-term memory technique. These optimizers contributed to addressing the stated concerns raised by the authors in the paper. The proposed approach was applied to classify and analyze real-world as well as medical time series datasets, specifically the Breast Cancer Wisconsin Dataset and the Epileptic Seizure Recognition Dataset. The experimentation was rigorously validated through the application of fivefold cross-validation methodology. Elgamal et al. [28] proposed chaotic Harris Hawks optimization, a metaheuristic optimization technique that improved the HHO algorithm. That improvement required two changes. First, chaotic maps were added to HHO's startup phase to increase search space population diversity. Second, the simulated annealing algorithm refined the best solution, improving HHO's exploitation. Applying CHHO to 14 UCI machine learning repository medical benchmark datasets showed its efficacy. Oyelade et al. [29] introduced a novel metaheuristic algorithm called the Ebola optimization search algorithm, by drawing inspiration from the propagation mechanism of the Ebola virus disease. Their new model was expressed through a system of differential equations. By combining the propagation model with mathematical equations, the EOSA was formulated as a metaheuristic algorithm. The algorithm's effectiveness was evaluated against established optimization methods using a comprehensive range of benchmark functions, both classical and constrained. Moreover, the EOSA was applied to optimize the hyperparameters of a convolutional neural network (CNN) for digital mammography to classify image, achieving an impressive 96.0% accuracy in detecting breast cancer.

3 Methodology

The flow that has been used to conduct the research has been presented in this section. Various libraries have been initially imported, such as Numpy, Pandas, Matplotlib, Keras, seaborn, warnings, and sklearn. The Python language was used to carry out the work in Jupyter, windows 11. The framework for the same is shown in Fig. 1.

3.1 Collection of data

The data has been taken from the medical dataset, which has the standard information related to individuals from various ancestral lines [30]. The data has been taken in the form of .csv format, split into fourteen columns, and 2000 records

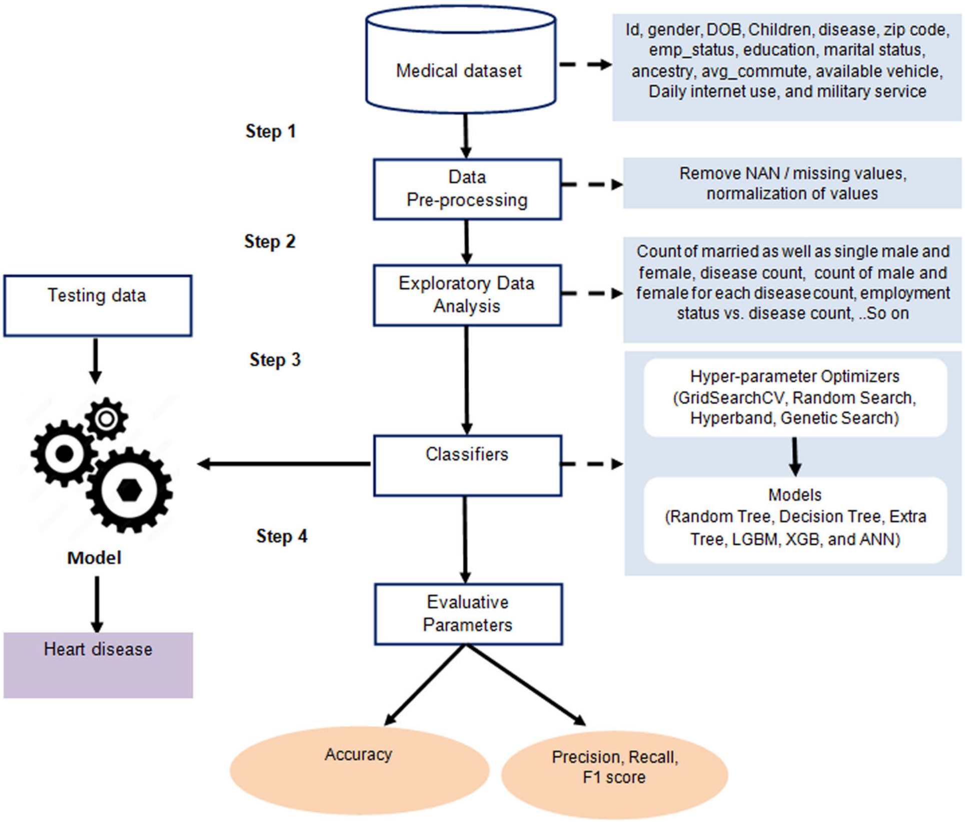

Medical dataset
Id, gender, DOB, Children, disease, zip code, emp_status, education, marital status, ancestry, avg_commute, available vehicle, Daily internet use, and military service
Step 1
Data Pre-processing
Remove NAN / missing values, normalization of values
Step 2
Exploratory Data Analysis
Count of married as well as single male and female, disease count, count of male and female for each disease count, employment status vs. disease count, ..So on
Step 3
Testing data
Hyper-parameter Optimizers (GridSearchCV, Random Search, Hyperband, Genetic Search)
Classifiers
Models (Random Tree, Decision Tree, Extra Tree, LGBM, XGB, and ANN)
Model
Step 4
Evaluative Parameters
Heart disease
Accuracy
Precision, Recall, F1 score

Fig. 1 Proposed design for multi-disease detection system

Table 1 Attributes of the medical dataset

Attribute	Values				
Id	Amelia Nixon	Clara Hicks	Mason Brown	Micheal Rice	Eleanor Ritter
Gender	Female	Female	Male	Male	Female
Dob	1944–03-09	1966–07-02	1981–05-31	1945–02-13	1939–09-03
Zipcode	89,136	94,105	89,127	44,101	89,136
Emp_status	Retired	Employed	Employed	Retired	Retired
Education	Bachelors	Phd/md	Masters	Bachelors	Masters
Martial_status	Married	Married	married	Married	Married
Children	1	4	2	2	3
Ancestry	Portugal	Sweden	Germany	Denmark	Austria
Avg_commute	13.38	15.16	23.60	19.61	36.55
Daily_internet_use	2.53	6.77	3.63	5.00	7.75
Available vehicle	2	2	1	3	1
Military service	No	No	No	No	No
Disease	Hypertension	Endometriosis	Prostate cancer	Multiple sclerosis	Skin cancer

such as ID, gender where 51% are female, and 49% are male, date of birth, Zipcode, etc. In addition, the medical data also contains 12 classes of diseases, such as hypertension, Alzheimer's disease, multiple sclerosis, endometriosis, prostate cancer, heart disease, HIV/AIDS, gastric, skin cancer, kidney disease, breast cancer, and schizophrenia. The attributes of the dataset which have been taken are shown in Table 1.

Table 2 Checking of null values in medical dataset

Attribute	Value
Id	False
Gender	False
Dob	False
Employment_status	False
Education	False
Martial_status	False
Children	False
Ancestry	False
Avg_commute	False
Daily_internet_use	False
Available_vehicles	False
Military_service	False
Disease	False

Table 3 Applying encoders to the dataset

```
Data[‘gender’] = label_encoder.fit_transform(data[‘gender’])
  Data[;employment_statur’] =
  label_encoder.fit_transform(data[‘employment_statur’])
  Data[‘education’] =
  label_encoder.fit_transform(data[‘education’])
  Data[‘marital_status’] =
  label_encoder.fit_transform(data[‘marital_status’])
  Data[‘ancestry’] =
  label_encoder.fit_transform(data[‘ancestry’])
  Data[‘disease’] =
  label_encoder.fit_transform(data[‘disease’])
  Data[‘military_service’] =
  label_encoder.fit_transform(data[‘military_service’])
```

3.2 Data preprocessing

The most important and first phase of developing the model is preprocessing the data in which the null values are being checked using SimpleImputer Library. During execution, it has been found that all the attributes generated false values as shown in Table 2, which means there are no null or missing values against any attribute.

In addition to this, encoding of dataset has been done to normalize it so that the features can be easily selected from the entire dataset for both training and testing dataset. The code to apply encoders is shown in Table 3 for both training and testing dataset.

3.3 Exploratory data analysis

The preprocessed data is later visualized in terms of bar graphs for a better understanding of the data in terms of various attributes such as the number of males and females, count of diseases, number of males and females affected by these diseases, usage of the internet on a daily basis, number of people who are in the military and are not in the military, employment status, ancestral lines, etc.

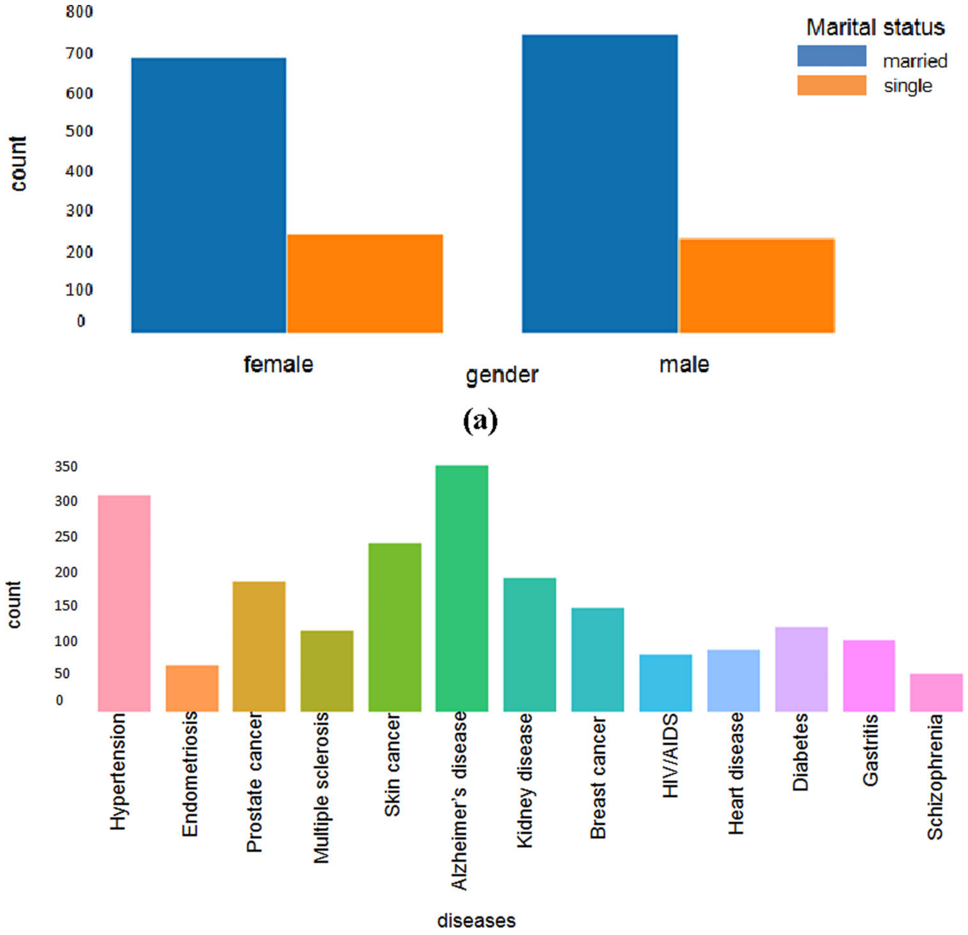
From Fig. 2a, b, gender attribute has been taken to present the count of male and female as well as their marital status. As we can see, there are 950 females, out of which the count of married females is 700, and single 250. On the contrary, out of 1000 males, 750 are married, and 250 are single. From the disease attribute, the data has been analyzed based on a count of diseases. As earlier mentioned, 13 diseases have been taken out of the most occurred one is Alzheimer’s disease with a count of approx 350. The minor count has been shown by Schiropezia of 50. On assaying the count of other diseases, it has been found that hypertension (350), endometriosis (55), prostate cancer (155), multiple sclerosis (100), skin cancer (225), kidney disease (155), breast cancer (150), HIV/AIDS (55), heart disease (55), diabetes (100), and gastric (100). It has been found how many males and females are affected by these diseases based on the disease count.

In Fig. 3c, from the female side, out of 950, the highest disease occurred is Alzheimer’s disease, with a count of 155, followed by the second highest, hypertension, with approximately 152. The least disease count is shown by schizophrenia, with a count of 25, and prostate cancer has a count of 0. Similarly, from the male end, out of 1000, the highest disease count has been shown by prostate cancer and Alzheimer’s disease, with a count of 175. On the contrary, the least count of disease is shown by schizophrenia, with a count of 25, followed by zero counts of endometriosis and breast cancer.

In Fig. 3d, the employment status has been broadly classified into four categories: retired, employed, unemployed, and student. On comparing all these categories with the diseases, it has been found that from retried section, the highest count of disease that has occurred is Alzheimer’s disease (180), followed by hypertension (140). The least disease found for this category is HIV/AIDS, with a count of 5–10. In the employed group, hypertension has the highest count at 115, followed by Alzheimer’s disease at 105.

In contrast, the least count of 15 and 20 has been shown by endometriosis and schizophrenia, respectively. In the unemployed category, the highest disease count has been shown by Alzheimer’s and hypertension, while the lowest disease that has occurred in these people is schizophrenia. At the end for the student category, the count of diseases is very low, and very few diseases have happened to them, such as hypertension, prostate cancer, HIV/AIDS, diabetes, gastritis, and schizophrenia.

Figure 4e describes the count of females and males mentioned in the dataset that belong from various countries such as Portugal, Sweden, Germany, Denmark, Austria, Hungary,



(a)

(b)

Fig. 2 EDA of **a** marital status of male and female, **b** count of diseases

Ireland, Ukraine, Russia, Netherlands, Poland, Belgium, Finland, England, Italy, Czech Republic, France, Scotland, Switzerland, and Spain. It has been found that most of the females belong to Switzerland, which counts more than 65, followed by Ireland and Poland, which counts 62. On the contrary, the least number of women has been shown in Scotland, with 35 in number. On the other side, the country which offers the largest count of males is Portugal with 70 in number followed by Sweden, Ireland, and Italy with 60 in number each, while Ukraine and the Czech Republic have shown the least with 40 each in number.

In continuation with this, another Fig. 4f reflects the count of the people based on the diseases such as hypertension, Alzheimer's disease, prostate cancer, breast cancer, etc., who worked either in military service or did not. It has been found that 300 people who do not work in military service mainly deal with Alzheimer's disease, followed by hypertension in 255 people. On the contrary, only 50 non-military service people have schizophrenia which is the lowest among all. Now, on the other hand, when we look at the graph of those people who have been working in military service are suffering from the diseases such as hypertension, prostate cancer, multiple sclerosis, skin cancer, Alzheimer's disease, HIV/AIDS, kidney disease, heart disease, diabetes, gastritis, and schizophrenia and out of all these the highest number of military service people have Alzheimer disease which is 45 in number.

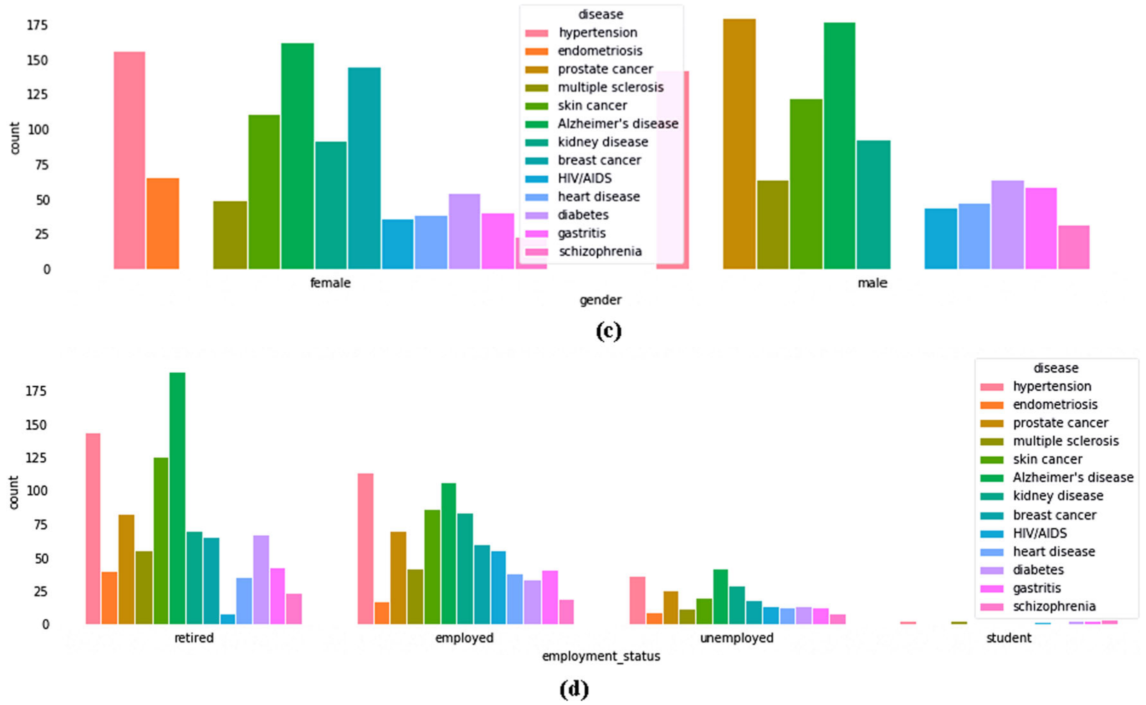
Besides the study of the above data, the graph in Fig. 5a, b is about the average commute, which means the people who travel a long distance for work as well as the people who use the internet daily, has also been presented in which the highest peak is pointed out at more at 0.040 and 0.25, respectively.

3.4 Classifiers

This section gives the brief description about the metaheuristic optimizers as well as the AI models that have been used to develop the multi-disease prediction system.

3.4.1 Applied models

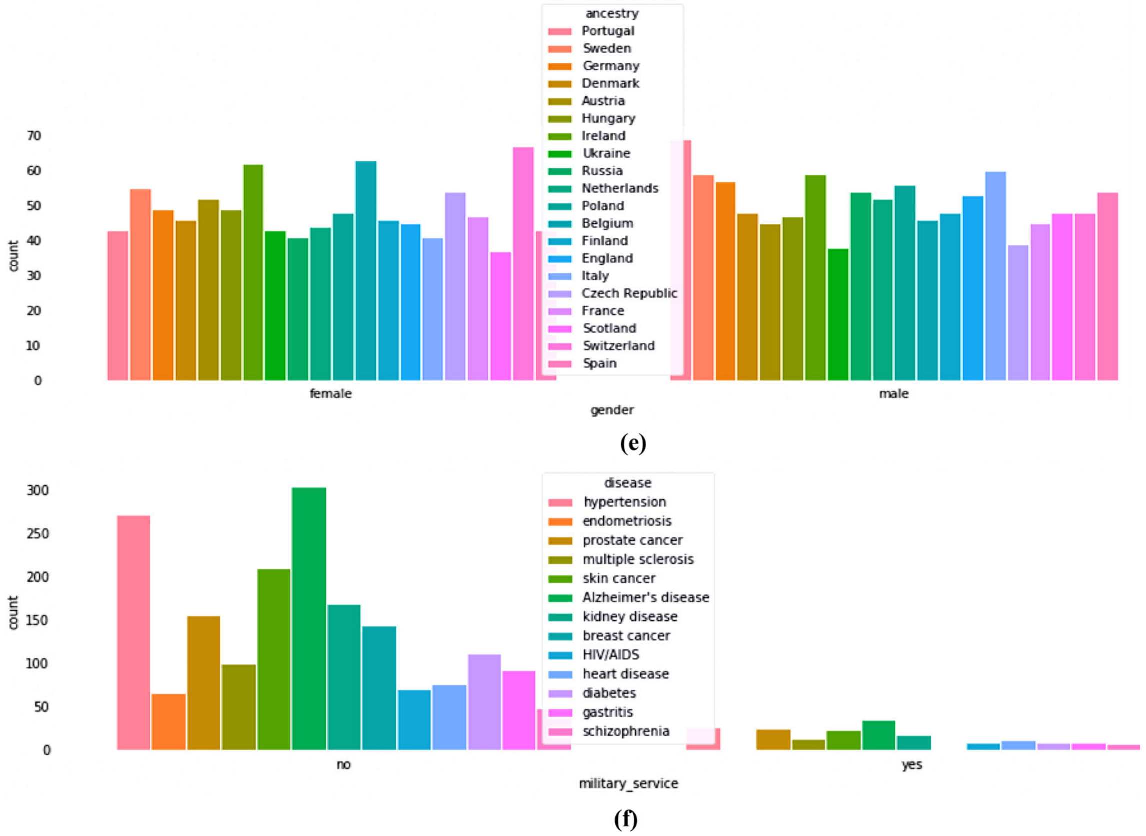
Several machine learning models are used, such as extra tree classifier, random forest, LGBM, decision tree, XGB, and



(c)

(d)

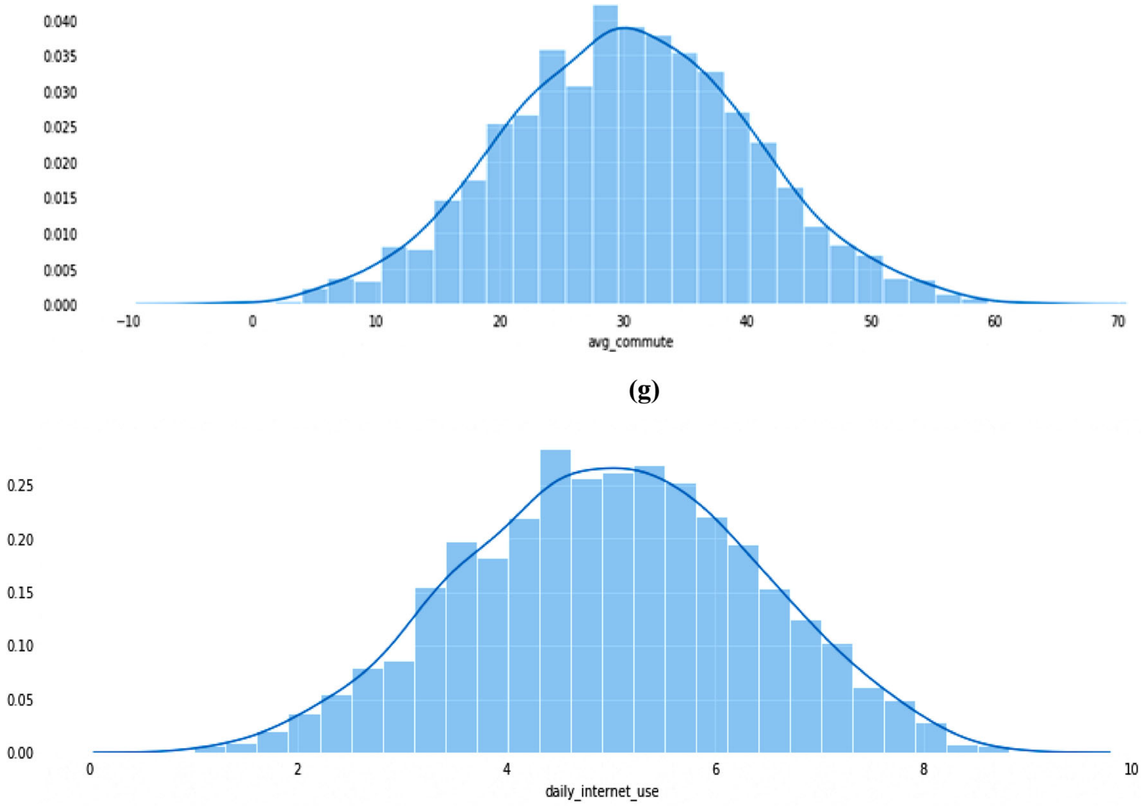
Fig. 3 EDA of **c** gender, **d** employment status vs. diseases



(e)

(f)

Fig. 4 EDA of **e** ancestry, **f** military service



(g)

(h)

Fig. 5 EDA of **g** avg_commute, **h** daily internet use

artificial neural network. In this part, we will talk briefly about each of these models.

Random Forest (RF) It is a machine learning technique given by Leo Breimna and Adele Cutler. This algorithm consists of various decision trees, where each tree has a data sample drawn from the training dataset based on replacement. This theory is also called a bootstrap sample. Further, these decision trees' outputs are combined to generate a single result. The random forest can control both regression and classification problems [31]. There are three hyperparameters in the random forest algorithm: the node's size, the number of trees, and the number of sampled features. An essential characteristic of this algorithm is that it prevents overfitting and can work on the continuous and categorical dataset in case of regression and classification, respectively. In addition, it has also been found that if the decision trees are higher in number, then the higher will be the accuracy of the random forest [32].

Decision Tree Classifier (DT) Like the random forest, this algorithm can also be performed on both regression and classification. If we look at the architecture of the decision tree, there are two nodes, i.e., the decision node and the second one is the leaf node. Decision nodes have multiple branches, which are used to make decisions, while the leaf nodes do not have any other branches and hence are used to generate the output [33]. This supervised machine learning algorithm is mainly a representation of data in graphical form to obtain all the possible solutions to any problem based on the given conditions. The decision tree algorithm starts working from the root node and compares the values of the root attribute with the actual dataset. Based on that comparison, the algorithm switches to the next node by following the corresponding branch. The same operation is repeated for the subsequent node, which means that the algorithm compares the attribute value to those of the sub-nodes and continues. The process is repeated until the leaf node of the tree is reached [34].

Extra Tree Classifier (ET) An extra tree is also known as an extremely randomized trees classifier. It is also an ensemble machine learning-based technique that uses multiple un-pruned decision trees and trains them to get the aggregated results and predict the output based on either a majority vote or taking the average of the outputs generated by the decision trees. The extra tree classifier tree is being constructed using the sample of original training data. For each test node, the random selection of k features is provided to every tree from the set of features so that each decision tree selects the best feature [35]. The extra tree classifier works like the random forest algorithm in which multiple decision trees are created. Still, unlike the replacement theory in a random forest, the sampling of each tree is random. This random sampling helps to create the dataset of each decision tree with unique samples. In fact, instead of calculating the entropy or Gini index value for splitting the data, this algorithm randomly selects the split value, which makes this tree uncorrelated and diversified. Moreover, the extra tree classifier works using Sckit-learn and helps build classification and regression models [36].

LGBM Classifier LGBM stands for light gradient boosting machine classifier, which also works on decision trees to improve its efficiency and reduce the usage of more memory. LGBM classifier uses two techniques, i.e., exclusive feature bundling and gradient-based one-side sampling. These techniques help the algorithm overcome the drawback of the histogram-based algorithm, which is mainly used in all gradient boosting decision tree frameworks [37]. LGBM splits the tree in a leaf-wise manner opposite to that of boosting algorithms which grow in the tree levels. The algorithm selects the leaf with maximum delta loss to grow, and as soon as the leaf is selected, it has been found that the level-wise algorithm shows a greater loss than the leaf-wise algorithm, which enhances the accuracy. But on the contrary, the leaf-wise kind of growth also increases the model's complexity and causes overfitting when dealing with a limited dataset. Overall, LGBM is distributed, fast, and performs well for classification, ranking, or any other machine learning task [38].

XGB Classifier XGB stands for extreme gradient boosting algorithm, which relies on a decision tree and uses boosting to improve performance. It is one of the most effective types of machine learning algorithm, producing the best results compared to other algorithms like the random forest, logistic regression, etc. In this research work, we have used this algorithm as it works nicely with the Sckit-learn machine learning framework is good at solving regression and classification-based problems effectively [39].

Artificial Neural Network (ANN) The word 'artificial neural network' derives from the biological nervous system, which develops the human brain structure. Like the neurons in the human brain are connected, in the same way, the artificial neural network also has specific neurons, also called nodes, that are connected to each other in various types of layers [40]. The role of an artificial neural network is to make the computer understand things and make decisions as humans do in a real-life manner. ANN consists of various elements or layers, such as an input layer, a hidden layer, and an output layer. The input in the form of weights is received in the input layer which is further sent to the hidden layer where bias is added, and an activation function is used to process the equation. In the last, the output is generated in the output layer as a predicted value [41]. In Fig. 6, a mathematical diagram of ANN is shown where the arrow is the connection between the two nodes or neurons and represents the path through which the information is flown. Each connection has a weight which is an integer number used for controlling the signal between the two neurons. If the output layer of ANN can compute the desired output, then weights are not being adjusted, while if the output is undesired, then the weights are being changed till we get better results from the ANN model.

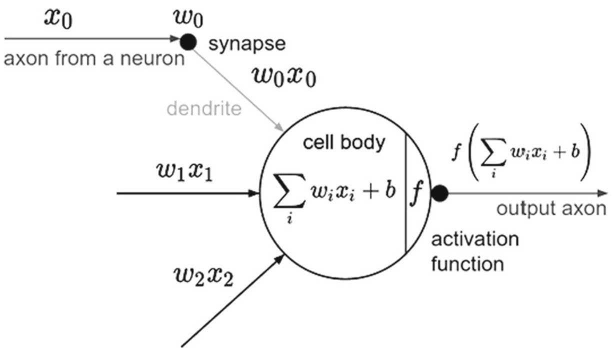


Fig. 6 Mathematical architecture of ANN model

In the context of machine learning models, ***hyperparameters*** are essential tuning parameters that influence the behavior of the model during training. Here, the hyperparameters for various algorithms, such as random forest (RF), decision tree (DT), extra trees (ET), LightGBM (LGBM), XGBoost (XGB), and artificial neural network (ANN), are provided which have been taken while training them with the dataset with the value of cross-validation as 3. For the **Random Forest algorithm,** the 'n_jobs' parameter is set to -1 which enables the usage of all available processors during training. The 'Random_state' parameter is set to 1 for reproducibility to ensure the consistent results across different runs. In case of **decision tree**, the 'Class_weight' is not specified, hence implying equal weight for all classes. The 'Criterion' is set to 'Gini,' which signifies the splitting criterion for decision trees in the forest.

Moving to **LightGBM**, the 'Colsample_bytree' is set to 0.45 that controls the fraction of features to consider when building each tree. The 'Learning_rate' is set to 0.057, and the 'Max_depth' is 14 which determine the maximum depth of the individual trees. Additionally, 'Num_leaves' is set to 5 that influence the maximum number of leaves in one tree. For **XGBoost,** 'Colsample_bytree' is specified as 0.8 to represent the fraction of features to be randomly sampled for each tree. The 'Max_depth' is set to 2 that determine the maximum depth of a tree. The 'Gamma' parameter, which controls the minimum loss reduction required to make a further partition on a leaf node, is set to 0. Later, the objective function is 'Binary:logistic,' that indicates a binary classification problem. Lastly, for the **Artificial Neural Network (ANN)**, 'Neurons' represents the number of neurons in the hidden layer which is set to 132. 'Batch_size' is set to 32, which indicates the number of samples used in each iteration during training. 'Epochs' is set to 20, which define the number of times the learning algorithm will work through the entire training dataset. The 'Activation' function is specified as 'Relu,' and the 'Seed' parameter is set to 27 for reproducibility.

These hyperparameter configurations play a crucial role in shaping the performance and generalization ability of the respective machine learning models. Proper tuning of these parameters is vital to achieving optimal results in model training and deployment.

3.4.2 Hyperparameter optimizers

In this section, the hyperparameters of the models have been fine-tuned using various hyperparameter optimizers such as grid search CV, random search, hyperband, and genetic search to obtain optimal results. Grid search CV is also known as grid search cross-validation. It searches the configurations of all the given hyperparameters and later fine-tunes them. This tool helps in finding the optimal values of a model from the given set of parameters present in a grid. Grid search CV uses the parameters such as an estimator, parameter grid, scoring, cross-validation, verbose, and number of jobs during its execution [42]. On integrating with the entire applied machine learning models with grid search CV, the hyperparameters obtained are presented in Table 4.

The second hyperparameter optimizer that has been used is random search, in which the combinations in the search space are randomly searched but in the bounded domain. It is somewhat similar to grid search CV, but unlike grid search, here it is necessary to specify the number of models we train. In addition, the set of possible values for the hyperparameters is also not selected here, instead of which statistical distribution is used. Another major difference is that when we apply grid search CV, every possible combination of possible hyperparameters is found to obtain the best model. In contrast, random search does exactly the opposite of it, i.e., it tests only the randomly selected combination of hyperparameters [43]. On integrating with the entire applied machine learning models with random search, the hyperparameters obtained are shown in Table 5.

The third hyperparameter optimizer used is hyperband optimizer which is used to tune iterative algorithms. This novel optimizer is flexible and simple and allocates predefined resources to the models, such as iterations, number of data samples, features, etc. This optimizer generates subsets in small sizes and allocates the aforementioned resources to every hyperparameter combination based on its performance [44]. On integrating the entire applied machine learning models with hyperband search, the hyperparameters obtained are shown in Table 6.

The last hyperparameter optimizer used is genetic search which provides a powerful technique to fine-tune the parameters. The parameters used primarily by genetic search are probability crossover, mutation probability, and population size. This algorithm is based on natural genetic selection. It detects the hyperparameter combinations in every generation, which are then passed to the next one until or unless the combination which is best-performing is detected [45]. On the contrary, this optimizer also has a drawback: it is not suited for parallelization. On integrating the entire applied machine learning models with genetic search, the hyperparameters obtained are shown in Table 7.

3.5 Evaluative parameters

After integrating the machine learning models with the metaheuristic optimizers to fine-tune their hyperparameters for the medical dataset, the performance of these models is evaluated based on various parameters such as precision, accuracy, recall, and F1 score [46–50]. Accuracy is an important parameter used to compute the model's effectiveness in predicting the output class of data. Initially, the models are trained, and their accuracy values are calculated to check how efficiently they have been trained. It is shown by Eq. (1). After accuracy, other values of the models are also computed, such as Precision, in which the relevant class values are obtained out of the retrieved values of the class. On the contrary, recall is the proportion of relevant instances retrieved. These are computed by Eqs. (2, 3). Based on the harmonic values of both precision and recall, the F1 score value is generated using Eq. (4).

$$\text{Accuracy} = \frac{\text{True Positive} + \text{True Negative}}{\text{True Positive} + \text{False Negative} + \text{True Negative} + \text{False Positive}} \tag{1}$$

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} \tag{2}$$

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} \tag{3}$$

$$F1\ \text{score} = 2\frac{\text{Precision} * \text{Recall}}{\text{Recall} + \text{Precision}} \tag{4}$$

Table 4 Fine-tuning of hyperparameters using GridSeacrhCV

Models	Hyperparameters
Random forest	‘n’$_{\text{estimators}}$: [10, 20, 30], criterion: [‘gini’, ‘entropy’], ‘max_depth’: [15, 20, 30, 50]
Decision tree	‘max’$_{\text{depth}}$: [15, 20, 30, 50], ‘n_estimators’: [10, 20, 30], ‘criterion’: [‘gini’, ‘entropy’]
Extra tree	criterion: [‘gini’, ‘entropy’], ‘max’$_{\text{depth}}$: [15, 20, 30, 50], ‘n_estimators’: [10, 20, 30],
LGBM	‘learning_rate’:[0.05], ‘nthread’: [4], ‘silent’: [1], ‘objective’:]binary: logistic’], ‘min_child_weight’: [11], ‘subsample’: [0.8], ‘n-estimators’: [5], ‘missing’: [-999], ‘send’ [1337], ‘colsample_bytree’: [0.7], ‘max_depth’: [6],
XGB	‘nthread’: [4], ‘learning’$_{\text{rate}}$: [0.05], ‘objective’: [binary: logistic], ‘max_depth’: [6], ‘silent’ [1], ‘min_child_weight’: [11] ‘subsample’: [0.8], ‘colsample_bytree’: [0.7], ‘missing’: [-999], ‘n_estimators’: [5], ‘seed’ [1337]
ANN	‘patience’: [2, 5], ‘batch’$_{\text{size}}$: [16, 32], ‘activation’: [‘relu’, ‘tanh’], ‘neurons’: [16, 32], ‘epochs’: [20, 50]

Table 5 Fine-tuning of hyperparameters using random search

Models	Hyperparameters
Random forest	‘max_depth’: sp_randint (5, 50), ‘n’$_{\text{estimators}}$: sp$_{\text{randint(10, 100)}}$, max_features: sp$_{\text{randint(1, 64)}}$, ‘min_samples_leaf’: sp_randint (1, 11), ‘criterion’: [‘entropy’, ‘gini’], ‘min_samples_split’: sp_randint(2, 11),
Decision tree	max_features: sp$_{\text{randint(1, 64)}}$, ‘n’$_{\text{estimators}}$: sp$_{\text{randint(10, 100)}}$, ‘max’$_{\text{depth}}$: sp$_{\text{randint (5, 50)}}$, ‘criterion’: [‘gini, ‘entropy’], ‘min_samples_split’: sp$_{\text{randint(2, 11)}}$, ‘min_samples_leaf’: sp$_{\text{randint (1, 11)}}$
Extra tree	‘n’$_{\text{estimators}}$: sp$_{\text{randint(10, 100)}}$, max_features: sp$_{\text{randint(1, 64)}}$, ‘max’$_{\text{depth}}$: sp$_{\text{randint (5, 50)}}$, ‘min_samples_split’: sp$_{\text{randint(2, 11)}}$, ‘min_samples_leaf’: sp$_{\text{randint (1, 11)}}$ ‘criterion’: [‘gini, ‘entropy’]
LGBM	‘max_depth’: sp_randint (5, 50), ‘n’$_{\text{estimators}}$: sp$_{\text{randint(10, 100)}}$, min_samples_split: sp$_{\text{randint(2, 11)}}$, max_features: sp$_{\text{randint(1, 64),}}$ min_samples_split: sp$_{\text{randint(1, 11)}}$, ‘criterion’: [‘gini, ‘entropy’]
XGB	‘n’$_{\text{estimators}}$: sp$_{\text{randint(10, 100)}}$, max_features: sp$_{\text{randint(1, 64)}}$, ‘criterion’: [‘gini, ‘entropy’] min_samples_split: sp$_{\text{randint(2, 11)}}$, ‘max’$_{\text{depth}}$: sp$_{\text{randint (5, 50)}}$, min_samples_leaf: sp$_{\text{randint (1, 11)}}$
ANN	‘batch’$_{\text{size}}$: [16, 32], ‘epochs’: [20, 50], ‘neurons’: [16, 32] ‘activation’: [‘relu’, ‘tanh’], ‘patience’: [2, 5]

Table 6 Fine-tuning of hyperparameters using hyperband

Models	Hyperparameters
Random forest	‘min_samples_split’: list(range(2, 12)), ‘max_depth’: [2, 3, 4]
Extra tree	‘min_samples_split’: list(range(2, 12)), ‘max_depth’: [2, 3, 4]
Decision tree	‘max_depth’: [2, 3, 4], ‘min_samples_split’: list(range(2, 12))
LGBM	‘max_depth’: [2, 3, 4], ‘min_samples_split’: list(range(2, 12))
XGB	‘min_samples_split’: list(range(2, 12)), ‘max_depth’: [2, 3, 4]
ANN	‘batch’$_{\text{size}}$: [16, 32, 64], ‘activation’: [‘relu’, ‘tanh’, ‘neurons’: sp$_{\text{randint(10, 100)}}$, ‘patience’: sp_randint(3, 20), ‘epochs’: [20, 50]

4 Results

This section defines the results generated by the models in terms of accuracy, precision, recall, and F1 score. The models, such as random forest, extra tree classifier, decision tree classifier, XGB classifier, LGBM classifier, and ANN, have been trained without and under each optimizer like random search, grid search CV, hyperband, and genetic search. Additionally, the models have been reviewed for the classes of various diseases such as hypertension, Alzheimer’s disease, multiple sclerosis, endometriosis, prostate cancer, heart disease, HIV/AIDS, gastric, skin cancer, kidney disease, breast cancer, and schizophrenia by generating their precision, recall, and F1 score values.

Table 7 Fine-tuning of hyperparameters using genetic search

Models	Hyperparameters
Random forest	'n_estimators': range(10,100), 'max_depth': range(5,50), 'min_samples_leaf':range(1,11), 'criterion':['gini','entropy'], 'min_samples_split':range(2,11)
Decision tree	'max_depth': range(5,50), min_samples_leaf: range(1,11) 'criterion':['gini','entropy'], 'min_samples_split':range(2,11)
Extra tree	'min_samples_split':range(2,11), 'min_samples_leaf': range(1,11), 'max_depth': range(5,50), 'criterion':['gini','entropy']
LGBM/ XGB	'criterion':['gini','entropy'], 'max_depth': range(5,50)
ANN	'batch_size': [16,32,64], 'activation': ['relu','tanh'], 'epochs': [20,50], 'neurons':range(10,100), 'patience':range(3,20)

In Table 8, initially, the models are evaluated for the combined dataset based on accuracy under different optimizers. In evaluating the performance of various classifiers, it is evident that the light gradient boosting machine (LGBM) outperforms other models by achieving the highest accuracy at 95.54%. The decision tree and artificial neural network (ANN) also exhibit commendable accuracy levels of 91.25% and 92.21%, respectively. However, the random forest and extra tree classifier computed lower accuracies at 76.83% and 74.77% which indicates potential limitations in their predictive capabilities. Notably, the XGBoost (XGB) model lags significantly behind with an accuracy of 43.56% which thereby suggests a suboptimal performance in comparison to the other models considered in this analysis. In assessing the model performance with the usage of grid search cross-validation for hyperparameter tuning, the decision tree, light gradient boosting machine (LGBM), and artificial neural network (ANN) stand out by achieving perfect accuracies of 100%. This suggests that the selected hyperparameters through the grid search optimization led to highly fitting models for these algorithms. The random forest and extra tree classifier also demonstrate strong performances with accuracies of 92.39% and 96.91%, respectively. On the other hand, the XGBoost (XGB) model lags behind the top performers with an accuracy of 90.67% which indicates a comparatively lower optimization level through the grid search process. The application of hyperband optimization reveals distinctive performance patterns among the considered models. Notably, LGBM, XGB, and ANN exhibit exceptional accuracies, each achieving a perfect score of 100%. These results suggest that hyperband effectively identified hyperparameters that significantly enhance the predictive capabilities of these algorithms. However, the decision tree, extra tree classifier, and random forest models show lower accuracies at 67.21%, 59.66%, and 55.83%, respectively. This variance implies that hyperband may not have been as successful in finding optimal hyperparameters for these tree-based models, and further exploration into the specific configurations and potential limitations is warranted.

Table 8 Evaluation of models based using various optimizers

Optimizers	Models	Accuracy (%)
Normal	Decision tree	91.25
	Random forest	76.83
	Extra tree classifier	74.77
	LGBM	95.54
	XGB	43.56
	ANN	92.21
Grid search CV	Random forest	92.39
	LGBM	100
	ANN	100
	Extra tree classifier	96.91
	Decision tree	100
	XGB	90.67
Random Search	Random forest	100
	ANN	100
	LGBM	100
	Decision tree	100
	XGB	100
	Extra tree classifier	100
Hyperband	XGB	100
	Extra tree classifier	59.66
	Decision tree	67.21
	LGBM	100
	Random forest	55.83
	ANN	100
Genetic	Decision tree	100
	Random forest	95.02
	Extra tree classifier	93.53
	LGBM	100
	XGB	100
	ANN	100

On applying genetic optimization, it demonstrates notable improvements in the execution of various machine learning models. The decision tree, light gradient boosting machine (LGBM), XGBoost (XGB), and artificial neural network (ANN) all achieve perfect accuracies of 100% and indicate that the genetic algorithm effectively identified hyperparameter configurations that enhance their predictive capabilities. The random forest and extra tree classifier also exhibit commendable accuracies at 95.02% and 93.53%, respectively, as well as further highlight the effectiveness of the genetic optimization approach. The consistent high accuracies across diverse models suggest that the genetic algorithm provides a robust and versatile method for optimizing hyperparameters, contributing to enhanced model performance.

On assaying the results thoroughly, the models performed better after applying a random search optimizer as they computed the highest accuracy of 100% as compared to getting trained under other optimizers. On the contrary, when the same models were calculated without any optimizer, none of them showed the best accuracy values as they showed after applying optimizers.

Likewise, in Table 9, the models have also been evaluated based on precision, recall, and F1 score. In this performance analysis of various machine learning models across different optimizers, we observe notable variations in precision, F1 score, and recall metrics. Firstly, considering the **normal training scenario**, random forest exhibited a high precision of 0.90, while XGB classifier lagged significantly with precision as low as 0.17. Decision tree classifier, extra tree classifier, and ANN presented moderate performance across the metrics. When employing **Grid Search CV** for hyperparameter tuning, remarkable improvements were observed for most models. Random forest achieved an impressive precision of 0.95, and decision tree classifier attained perfect scores in precision, F1 score, and recall. LGBM Classifier and XGB classifier also demonstrated significant enhancements in their performance metrics. **Random Search** further elevated the model performances to perfect scores for all metrics, showcasing the effectiveness of this optimization method. Notably, all models, which include RF, decision tree classifier, extra tree classifier, LGBM classifier, extreme gradient boosting classifier, and ANN, achieved maximum precision, F1 score, and recall. **Hyperband optimization** yielded mixed results, with some models experiencing a decline in performance, particularly in precision and F1 score. LGBM classifier and XGB classifier maintained perfect scores, while random forest exhibited a decrease in precision to 0.42. Finally, **genetic algorithms** proved effective in enhancing model performances, with most models achieving perfect precision, F1 score, and recall. Decision tree classifier, extra tree classifier, as well as random forest displayed high precision and recall, emphasizing the robustness of genetic optimization.

Besides this, precision, recall, and F1 score of all the models under the four applied optimizers, such as grid search CV, random search, hyperband, and genetic Search, along with no optimizer for different classes have been also calculated in the form of the bar plot graph to examine the slope between them as shown in Fig. 7.

In Fig. 7a, the models were trained and assessed at first based on precision, recall, and F1 score, and no optimizers were used. When trained with random forest, it has been found that the model computed the highest precision value with 1.00 for the class Alzheimer's disease, HIV/AIDS, schizophrenia, and skin cancer. Likewise, in the case of a recall, the highest value of 1.00 has been obtained for the class hypertension, breast cancer, Alzheimer's disease, and prostate cancer. In the end, while evaluating for F1 score, the highest values have been computed for the class Alzheimer's disease only. On applying the decision tree, it has been discovered that the model obtained the highest precision value of 1.00 for each class except for endometriosis, gastric, and heart diseases.

Similarly, the highest recall along with the F1 score values is obtained for all the classes except endometriosis and gastritis. Further, when another model trained the same classes, i.e., the extra tree classifier, the classes that fall under the highest precision are HIV/AIDS, diabetes, endometriosis, gastritis, heart diseases, kidney disease, and multiple sclerosis with 1.00. Like the decision tree, the highest precision as well as F1 score value of 1.00 for each class except gastritis and heart diseases is obtained by the LGBM classifier. In contrast, the best recall value was computed for all classes except gastritis by the LGBM classifier. But when the same dataset was calculated using the XGB classifier, the highest F1 score, precision, and recall were obtained by Alzheimer's disease and skin cancer. At the same time, as for the maximum of the other classes, the model computed 0 values, meaning no relevant class has been returned from irrelevant ones. When ANN Model was applied, it generated the highest values of 1.00 for all three evaluative parameters.

In Fig. 7b, using grid search CV, the highest precision value of 1.00 has been obtained by random forest for the classes such as HIV/AIDS and schizophrenia and a recall value of 1.00 for classes like breast cancer, prostate cancer, Alzheimer's disease, and hypertension. On the contrary, the highest F1 score value has been obtained for the class hypertension with 0.99, while the least has been obtained for the class schizophrenia with 0.63. Likewise, the decision tree, LGBM, and ANN classifier computed the best values for all the classes of 1.00 in terms of precision, recall, and F1 score. In the case of the extra tree classifier, the highest precision, recall, and F1 score have been computed for the class Alzheimer's disease, HIV/AIDS, endometriosis, and skin cancer with 1.00.

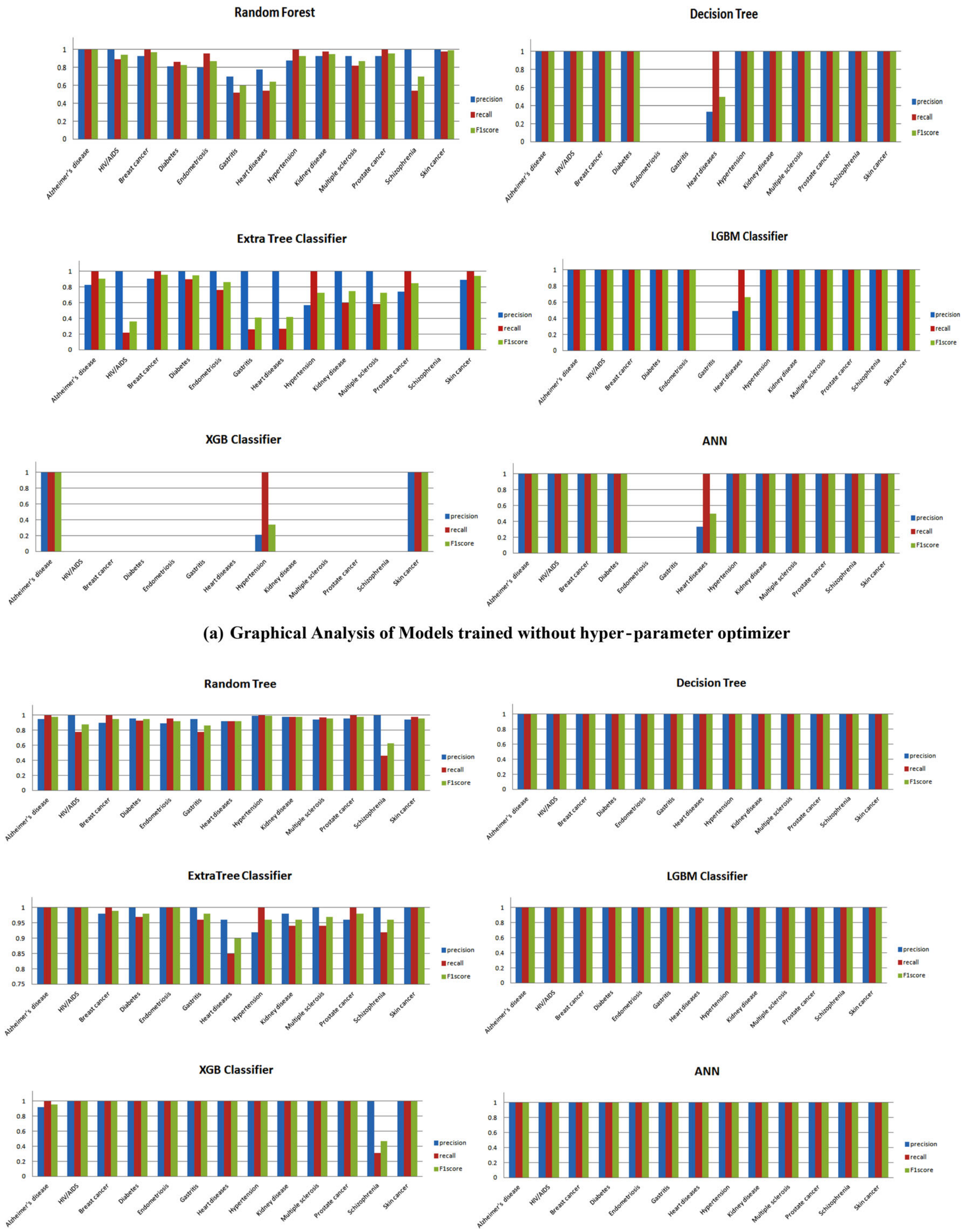
Table 9 Performance of models for various optimizers

Optimizers	Models	Precision	F1 Score	Recall
Normal	Random forest	0.90	0.87	0.85
	Extra tree classifier	0.84	0.68	0.66
	Decision tree classifier	0.79	0.81	0.85
	XGB classifier	0.17	0.18	0.23
	LGBM classifier	0.88	0.90	0.92
	ANN	0.79	0.81	0.85
Grid search CV	LGBM classifier	1.00	1.00	1.00
	Decision tree classifier	1.00	1.00	1.00
	Extra tree classifier	0.98	0.98	0.97
	ANN	1.00	1.00	1.00
	XGB classifier	0.99	0.96	0.95
	Random forest	0.95	0.92	0.91
Random search	Random forest	1.00	1.00	1.00
	XGB	1.00	1.00	1.00
	Extra tree classifier	1.00	1.00	1.00
	LGBM	1.00	1.00	1.00
	Decision tree	1.00	1.00	1.00
	ANN	1.00	1.00	1.00
Hyperband	Random forest	0.42	0.31	0.33
	Decision tree	0.37	0.39	0.46
	Extra tree classifier	0.48	0.35	0.38
	LGBM	1.00	1.00	1.00
	XGB	1.00	1.00	1.00
	ANN	1.00	1.00	1.00
Genetic	Random forest	0.96	0.94	0.93
	XGB	1.00	1.00	1.00
	Extra tree classifier	0.98	0.96	0.95
	LGBM	1.00	1.00	1.00
	Decision tree	1.00	1.00	1.00
	ANN	1.00	1.00	1.00

In Fig. 7c, under random search, all the models, such as RF, XGBoost, DT, LGBM classifier, extra tree classifier, and ANN, have computed the highest precision, recall, and F1 score values of 1.00 for all the classes. It means that a relevant search retrieved all the results.

In Fig. 7d, under hyperband search, the LGBM, XGB, and ANN classifiers all computed 1.00 as the highest score for recall, precision, and F1 score. At the same time, random forest obtained the highest precision for the classes such as breast cancer, Alzheimer's disease, HIV/AIDS, and skin cancer. The best recall has been computed by Alzheimer's disease, hypertension, and skin cancer; however, the highest F1 score has been calculated by Alzheimer's disease and skin cancer. Likewise, decision tree also obtained the highest precision, recall, and F1 score value for Alzheimer's disease, hypertension, kidney disease, and skin cancer. Moving to extra tree classifier, it has shown the average performance but on the contrary, these three techniques have also generated zero precision, recall, and F1 score values for certain classes which means that they are failing to correctly classify the instances of those classes.

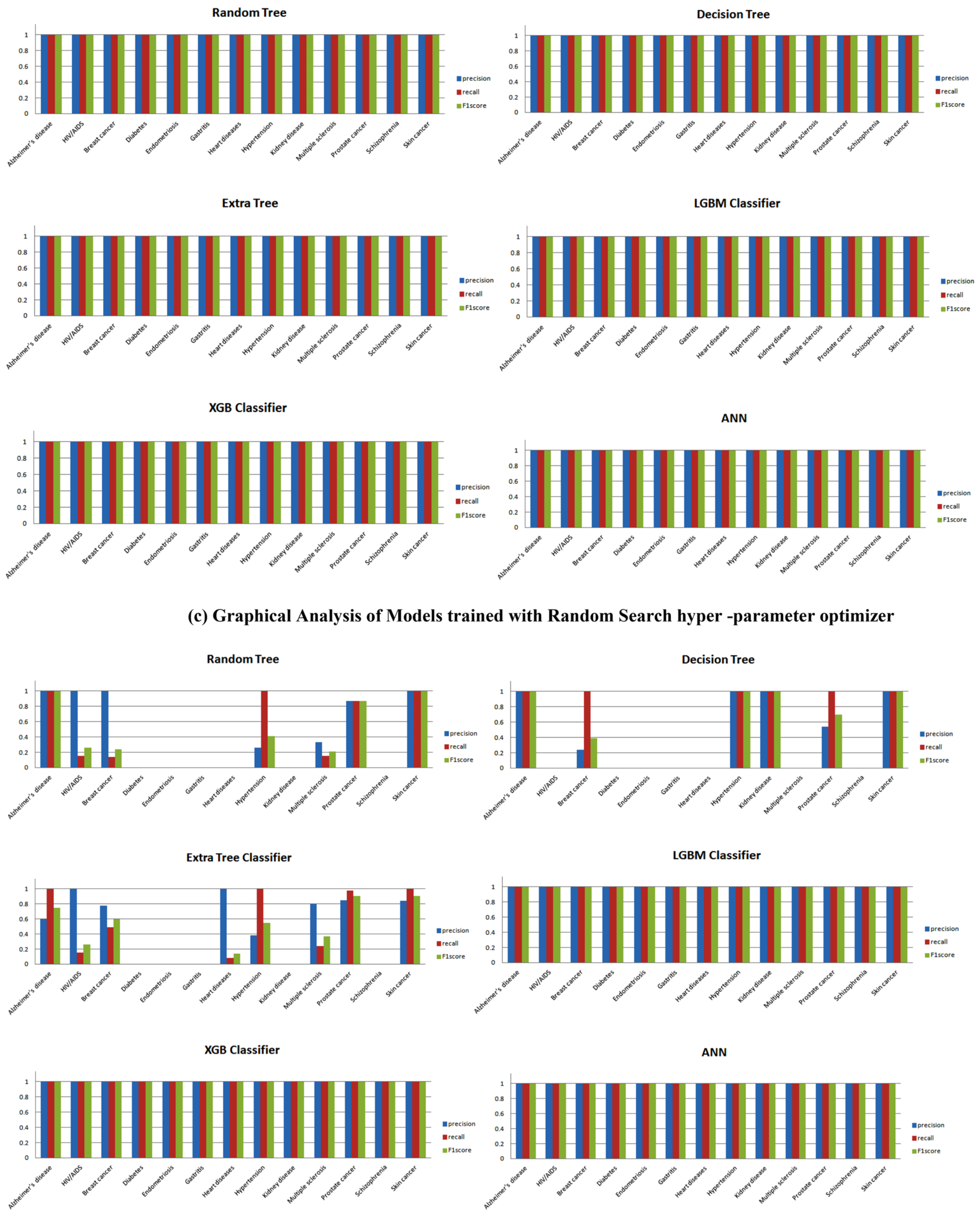
In Fig. 7e, under genetic search, the models, such as the decision tree classifier, LightGBM, XGB classifier, and ANN, have calculated the best recall, precision, and F1 score values of 1.00 while using random forest classifier, the best F1 score, precision, and recall value has been acquired for the classes such as diabetes, endometriosis, hypertension, and kidney disease with 1.00. In the end, using extra tree classifier, the highest precision values have been computed by all the classes except hypertension, breast cancer, Alzheimer's disease, prostate cancer, and skin cancer. In the same way, Alzheimer's disease, endometriosis, high blood pressure,



(a) Graphical Analysis of Models trained without hyper-parameter optimizer

(b) Graphical Analysis of Models trained with GridSearchCV hyper - parameter optimizer

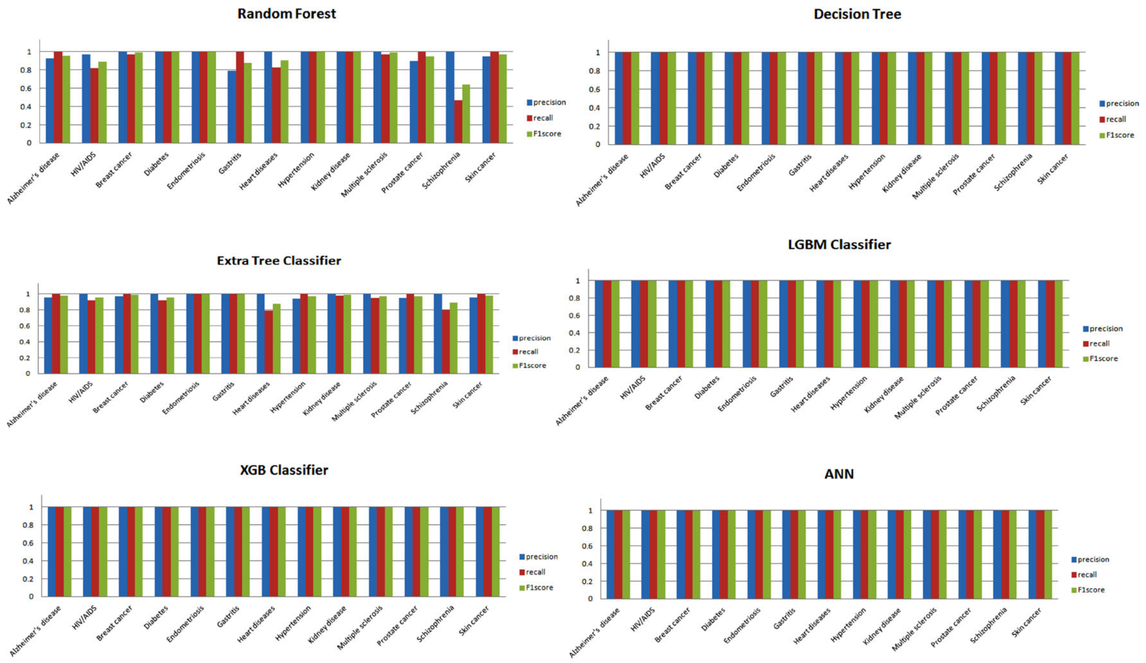
Fig. 7 **a** Graphical analysis of models trained without hyperparameter optimizer, **b** graphical analysis of models trained with grid search CV hyperparameter optimizer, **c** graphical analysis of models trained with random search hyperparameter optimizer, **d** graphical analysis of models trained with hyperband hyperparameter optimizer, **e** graphical analysis of models trained with genetic search hyperparameter optimizer



(c) Graphical Analysis of Models trained with Random Search hyper -parameter optimizer

(d) Graphical Analysis of Models trained with Hyperband hyper -parameter optimizer

Fig. 7 continued



(e) Graphical Analysis of Models trained with Genetic Search hyper-parameter optimizer

Fig. 7 continued

gastritis, breast cancer, and prostate cancer all have a recall value of 1.00. On the contrary, the highest F1 score value was computed by endometriosis and gastritis with the same value of 1.00.

5 Conclusion

In the past, determining the optimal value of any model was a labor-intensive task that demanded significant effort, time, and expertise. Thus, metaheuristic techniques prove invaluable in helping us to discover optimal solutions quickly and efficiently. The combination of metaheuristic techniques with machine learning and deep learning classifiers has consistently proven to be highly effective, reliable, and time-efficient. As a result, we employed six machine learning models and carefully adjusted their hyperparameters using four hyperparameter optimizers, taking into account the available information. The models under the random search optimizer demonstrated exceptional precision, accuracy, recall, and F1 score values. It was observed that the models performed more effectively after undergoing fine-tuning with hyperparameter optimizers compared to when they were not.

For future research, there is room to explore potential improvements by building upon the promising findings obtained from combining metaheuristic techniques and hyperparameter optimization in machine learning models. One possible way to improve is by investigating ensemble methods that use the capabilities of multiple models to develop a classifier that is more resilient and precise. Ensemble techniques, such as bagging and boosting, have demonstrated their effectiveness in mitigating overfitting and enhancing predictive performance.

Moreover, the integration of advanced models, coupled with optimization techniques, holds significant potential for augmenting the system's performance in predicting multiple diseases. This synergistic approach enhances the overall predictive capabilities of the system.

Conducting a comprehensive analysis of the misclassifications and understanding their root causes in case of models on incorporating with hyperband optimizer as well as using them by default is crucial for resolving the problem of zero precision, recall, and F1 score values for specific classes. This analysis could offer valuable insights for creating specialized strategies to handle the datasets as well as overcome challenges unique to various classes.

In addition, it is advised to explore state-of-the-art learning algorithms, such as advanced deep learning architectures, to capture intricate patterns in the data. Investigating architectures like transformer models or neural architecture search techniques could provide valuable insights into enhancing the models' capacity to comprehend complex relationships within the data. These enhancements aim to optimize the performance and dependability of the models, tackling existing constraints and facilitating their more efficient application in practical scenarios.

Funding Not applicable.

Declarations

Conflict of interest Authors declare no conflict of interest.

References

1. Lauri J, Dutta S, Grassia M, Ajwani D (2020) Learning fine-grained search space pruning and heuristics for combinatorial optimization. https://arxiv.org/abs/2001.01230
2. Linardatos P, Papastefanopoulos V, Kotsiantis S (2020) Explainable ai: a review of machine learning interpretability methods. Entropy 23(1):18
3. Gandomi AH, Yang XS, Alavi AH (2013) Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. Eng Comput 29:17–35
4. Singh P, Kottath R (2021) An ensemble approach to meta-heuristic algorithms: comparative analysis and its applications. Comput Ind Eng 162:107739
5. Xu Y, Liu X, Cao X, Huang C, Liu E, Qian S, Zhang J (2021) Artificial intelligence: a powerful paradigm for scientific research. Innov 2(4):100179
6. Karimi-Mamaghan M, Mohammadi M, Pasdeloup B, Meyer P (2023) Learning to select operators in meta-heuristics: an integration of Q-learning into the iterated greedy algorithm for the permutation flowshop scheduling problem. Eur J Oper Res 304(3):1296–1330
7. Kumar Y, Koul A, Sisodia PS, Shafi J, Kavita V, Gheisari M, Davoodi MB (2021) Heart failure detection using quantum-enhanced machine learning and traditional machine learning techniques for internet of artificially intelligent medical things. Wirel Commun Mob Comput 2021:1–16
8. Sarker IH (2021) Machine learning: algorithms, real-world applications and research directions. SN Comput Sci 2(3):160
9. Brownlee J (2016) Supervised and unsupervised machine learning algorithms. Mach Learn Mastery 16(03)
10. Gupta A, Koul A, Kumar Y (2022) Pancreatic cancer detection using machine and deep learning techniques. In: 2022 2nd international conference on innovative practices in technology and management (ICIPTM). IEEE, vol 2, pp 151–155
11. Sarker IH (2021) Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions. SN Comput Sci 2(6):420
12. Kumar Y, Kaul S, Sood K (2019) Effective use of the machine learning approaches on different clouds. In: Proceedings of International Conference on Sustainable Computing in Science, Technology and Management (SUSCOM), Amity University Rajasthan, Jaipur-India
13. Babu GS, Rao ST, Rao RR (2022) Automated assessment for Alzheimer's disease diagnosis from MRI images: meta-heuristic assisted deep learning model. Int J Imaging Syst Technol 32(2):544–563
14. Faris H, Aljarah I, Mirjalili S (2016) Training feedforward neural networks using multi-verse optimizer for binary classification problems. Appl Intell 45:322–332
15. Aljarah I, Al-Zoubi AM, Faris H, Hassonah MA, Mirjalili S, Saadeh H (2018) Simultaneous feature selection and support vector machine optimization using the grasshopper optimization algorithm. Cogn Comput 10:478–495
16. Tao Z, Huiling L, Wenwen W, Xia Y (2019) GA-SVM based feature selection and parameter optimization in hospitalization expense modeling. Appl Soft Comput 75:323–332
17. Faris H, Mirjalili S, Aljarah I (2019) Automatic selection of hidden neurons and weights in neural networks using grey wolf optimizer based on a hybrid encoding scheme. Int J Mach Learn Cybern 10:2901–2920
18. Mirjalili S (2015) How effective is the Grey Wolf optimizer in training multi-layer perceptrons. Appl Intell 43:150–161
19. Amirsadri S, Mousavirad SJ, Ebrahimpour-Komleh H (2018) A Levy flight-based grey wolf optimizer combined with back-propagation algorithm for neural network training. Neural Comput Appl 30:3707–3720
20. Hu A, Razmjooy N (2021) Brain tumor diagnosis based on metaheuristics and deep learning. Int J Imaging Syst Technol 31(2):657–669
21. Eshtay M, Faris H, Obeid N (2018) Improving extreme learning machine by competitive swarm optimization and its application for medical diagnosis problems. Expert Syst Appl 104:134–152
22. Shankar K, Manickam P, Devika G, Ilayaraja M (2018) Optimal feature selection for chronic kidney disease classification using deep learning classifier. In: 2018 IEEE international conference on computational intelligence and computing research (ICCIC). IEEE, pp 1–5
23. Chitradevi D, Prabha S, Prabhu AD (2021) Diagnosis of Alzheimer disease in MR brain images using optimization techniques. Neural Comput Appl 33:223–237
24. Canayaz M (2021) MH-COVIDNet: Diagnosis of COVID-19 using deep neural networks and meta-heuristic-based feature selection on X-ray images. Biomed Signal Process Control 64:102257
25. Roostaee S, Ghaffary HR (2016) Diagnosis of heart disease based on meta heuristic algorithms and clustering methods. J Electr Comput Eng Innov (JECEI) 4(2):105–110
26. Nadimi-Shahraki MH, Zamani H, Mirjalili S (2022) Enhanced whale optimization algorithm for medical feature selection: a COVID-19 case study. Comput Biol Med 148:105858
27. Rashid TA, Fattah P, Awla DK (2018) Using accuracy measure for improving the training of LSTM with metaheuristic algorithms. Procedia Comput Sci 140:324–333
28. Elgamal ZM, Yasin NBM, Tubishat M, Alswaitti M, Mirjalili S (2020) An improved harris hawks optimization algorithm with simulated annealing for feature selection in the medical field. IEEE Access 8:186638–186652
29. Oyelade ON, Ezugwu AES, Mohamed TI, Abualigah L (2022) Ebola optimization search algorithm: A new nature-inspired metaheuristic optimization algorithm. IEEE Access 10:16150–16177
30. Karimnahas (2018) Medical data. Kaggle. https://www.kaggle.com/datasets/karimnahas/medicaldata
31. Chaudhary A, Kolhe S, Kamal R (2016) An improved random forest classifier for multi-class classification. Inf Process Agric 3(4):215–222
32. Alzubi J, Nayyar A, Kumar A (2018) Machine learning from theory to algorithms: an overview. In: Journal of physics: conference series. IOP Publishing, vol 1142, p 012012
33. Ilyas H, Ali S, Ponum M, Hasan O, Mahmood MT, Iftikhar M, Malik MH (2021) Chronic kidney disease diagnosis using decision tree algorithms. BMC Nephrol 22(1):1–11
34. Lu Y, Ye T, Zheng J (2022) Decision tree algorithm in machine learning. In: 2022 IEEE international conference on advances in electrical engineering and computer applications (AEECA). IEEE, pp 1014–1017
35. Ampomah EK, Qin Z, Nyame G (2020) Evaluation of tree-based ensemble machine learning models in predicting stock price direction of movement. Information 11(6):332

36. Alfian G, Syafrudin M, Fahrurrozi I, Fitriyani NL, Atmaji FTD, Widodo T, Rhee J (2022) Predicting breast cancer from risk factors using SVM and extra-trees-based feature selection method. Computers 11(9):136
37. Mienye ID, Sun Y (2022) A survey of ensemble learning: concepts, algorithms, applications, and prospects. IEEE Access 10:99129–99149
38. Kumar GD, Deepa V, Vineela N, Emmanuel G (2022) Detection of Parkinson's disease using LightGBM Classifier. In: 2022 6th International conference on computing methodologies and communication (ICCMC). IEEE, pp 1292–1297
39. Trognon A, Cherifi YI, Habibi I, Demange L, Prudent C (2022) Using machine-learning strategies to solve psychometric problems. Sci Rep 12(1):18922
40. Micheli-Tzanakou E (2011) Artificial neural networks: an overview. Netw Comput Neural Syst 22(1–4):208–230
41. Gallo C (2015) Artificial neural networks tutorial. In: Encyclopedia of information science and technology, Third Edition. IGI Global, pp 6369–6378
42. Liashchynskyi P, Liashchynskyi P (2019) Grid search, random search, genetic algorithm: a big comparison for NAS. https://arxiv.org/abs/1912.06059
43. Bergstra J, Bengio Y (2012) Random search for hyper-parameter optimization. J Mach Learn Res 13(2)
44. Kumar Y, Gupta S (2023) Deep transfer learning approaches to predict glaucoma, cataract, choroidal neovascularization, diabetic macular Edema, DRUSEN and healthy eyes: an experimental review. Arch Computat Methods Eng 30:521–541. https://doi.org/10.1007/s11831-022-09807-7
45. Wang J, Xu J, Wang X (2018) Combination of hyperband and Bayesian optimization for hyperparameter optimization in deep learning. https://arxiv.org/abs/1801.01596
46. Sipper M, Fu W, Ahuja K, Moore JH (2018) Investigating the parameter space of evolutionary algorithms. BioData Min 11:1–14
47. Koul A, Bawa RK, Kumar Y (2022) Artificial intelligence techniques to predict the airway disorders illness: a systematic review. Arch Comput Methods Eng 1–34
48. Kaur S, Kumar Y, Koul A et al (2022) A systematic review on metaheuristic optimization techniques for feature selections in disease diagnosis: open issues and challenges. Arch Computat Methods Eng. https://doi.org/10.1007/s11831-022-09853-1
49. Kumar Y, Gupta S, Singla R et al (2022) A systematic review of artificial intelligence techniques in cancer prediction and diagnosis. Arch Computat Methods Eng 29:2043–2070. https://doi.org/10.1007/s11831-021-09648-w
50. Bhardwaj P, Bhandari G, Kumar Y et al (2022) An investigational approach for the prediction of gastric cancer using artificial intelligence techniques: a systematic review. Arch Computat Methods Eng 29:4379–4400. https://doi.org/10.1007/s11831-022-09737-4

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.