Analysis of Genetic Mutations Using Nature-Inspired Optimization Methods and Classification Approach

Anuradha Thakare, Pradnya Narkhede, and Sahil S. Adrakatti

1 Introduction

Cancer is a complex disease caused by genetic mutations that can occur in different parts of the body, leading to abnormal growth of cells and tumor formation. Identifying and classifying genetic mutations is essential for cancer diagnosis and treatment, as it can provide valuable information on the specific type of cancer and its potential response to treatment. However, detecting and analyzing genetic mutations can be difficult due to the vast number of possible mutations and their complex interactions. Early cancer detection can improve the chances of successful treatment and increase the chances of survival. The classification of genetic mutations can provide insights into the specific type of cancer and its potential response to treatment, allowing for personalized and targeted therapies. Failure to detect and classify genetic mutations can lead to misdiagnosis, inappropriate treatment, and poor clinical outcomes [1–3].

Nature-inspired optimization methods, such as Genetic Algorithms (GA) [4] and Particle Swarm Optimization (PSO) [5] are computational algorithms inspired by natural phenomena such as swarms, genetic evolution, and neural networks. These methods have shown promise in solving complex optimization problems, including feature selection and classification in cancer diagnosis. By leveraging the power of these optimization methods, it is possible to improve the accuracy and efficiency of classification models for genetic mutations in cancer patients. Additionally, these methods can be used to identify new biomarkers and potential targets for cancer treatment, leading to better patient outcomes.

A. Thakare (✉) · P. Narkhede · S. S. Adrakatti
Pimpri Chinchwad College of Engineering, Pune, Maharashtra, India
e-mail: anuradha.thakare@pccoepune.org; pradnya.narkhede@pccoepune.org

A. M. Anter et al. (eds.), *Nature-Inspired Methods for Smart Healthcare Systems and Medical Data*, https://doi.org/10.1007/978-3-031-45952-8_3

2 Related Research

The [6] article overviews the genetic alterations contributing to cancer development. It discusses the types of somatic mutations, copy number alterations, and structural variations that can lead to oncogenes' activation or tumor suppressor genes'inactivation. The authors also highlight the importance of identifying genetic alterations in cancer diagnosis, prognosis, and treatment and the challenges and opportunities for precision medicine.

The [7] Paper discusses an overview of cancer genomics, from discovering oncogenes and tumor suppressor genes to developing personalized medicine. The authors discuss the advances in genomic technologies, such as next-generation sequencing that have enabled the identification of genetic alterations in tumors. They also highlight the challenges and opportunities for using genomic information to guide cancer diagnosis, prognosis, and treatment.

In [8], this review article provides an overview of the genomic alterations contributing to cancer development, including somatic mutations, copy number alterations, and structural variations. The authors discuss emerging technologies for detecting and analyzing tumor genetic alterations, such as single-cell sequencing and liquid biopsy. They also highlight the challenges and opportunities for using genomic information to guide cancer diagnosis, prognosis, and treatment.

In [9], the review article provides an overview of the genetic mutation that contribute to breast cancer development, including structural variations, copy number alterations, and somatic mutations. The authors discuss the clinical implications of genetic testing for breast cancer diagnosis and treatment, including targeted therapies and immunotherapies.

In [10], the review article provides an overview of the molecular profiling of cancer, including identifying genetic alterations that drive cancer biology and using genomic information for personalized medicine. The authors discuss the advances in genomic technologies, such as whole-genome sequencing and transcriptomic that have enabled the identification of genetic alterations in tumors. They also highlight the challenges and opportunities for using genomic information to guide cancer diagnosis, prognosis, and treatment.

The [11] review article discusses the applications of machine learning in cancer prediction and prognosis. It provides an overview of various machine learning techniques, including neural networks, random forests, decision trees, and support vector machines, their applications in cancer classification using genetic mutations.

In [12] research article proposes a deep learning approach for classifying cancer types based on copy number alterations. The authors developed a convolutional neural network model and applied it to genomic data from 13 cancer types. They demonstrated that their approach achieved high accuracy in cancer classification and outperformed other machine learning methods.

In [13] This review article discusses the applications of machine learning in predicting the pathogenicity of genetic variants associated with cancer. The authors provide an overview of various machine learning techniques, including Deep

learning, random forests, decision trees, and support vector machines, and their applications in cancer genetics.

In [14] This research article proposes machine learning models for predicting oncogenic mutations in cancer patients. The authors developed logistic regression models and applied them to genomic data from cancer patients. They demonstrated that their approach achieved high accuracy in predicting oncogenic mutations and outperformed other machine learning methods.

In [15] This review article discusses the applications of machine learning in identifying driver mutations in cancer genomics. The authors provide an overview of various machine learning techniques, including neural networks, random forests, and support vector machines, and their applications in identifying driver mutations.

In [16] This research article proposes a deep learning approach for classifying genetic variants in cancer genes. The authors developed a deep neural network model and applied it to genomic data from cancer patients. They demonstrated that their approach achieved high accuracy in classifying genetic variants and outperformed other machine learning methods.

A genetic algorithm-based feature selection method for cancer classification utilizing microarray gene expression data was suggested in the study by Shahla Nosrati et al. The authors utilized a support vector machine (SVM) classifier to categorise cancers and a genetic algorithm to choose the most pertinent genes from the microarray data. The proposed strategy beat existing feature selection approaches in terms of accuracy and effectiveness when tested on six different cancer datasets, according to the results [17].

For detecting cancer driver genes, Xiao-Li Li et al. proposed a hybrid optimization technique. To find driver genes linked to cancer, the authors combined the genetic algorithm (GA) with particle swarm optimization (PSO). The proposed method beat previous optimization algorithms in terms of accuracy and stability when evaluated on four cancer datasets, according to the results [18].

A hybrid artificial bee colony optimization algorithm for cancer classification utilizing gene expression data was suggested in the publication by M. A. Arvind et al. The most informative genes from the microarray data were chosen by the authors using the artificial bee colony (ABC) algorithm, and the cancer types were categorized using a support vector machine (SVM) classifier. Three different cancer datasets were used to test the suggested strategy, and the results revealed that it performed better than other feature selection approaches in terms of accuracy and stability [19].

Ant colony optimization (ACO) was suggested by R. Balamurugan et al. as a method for improving gene expression data for the categorization of cancer. The authors utilized a decision tree classifier to categories cancers and an ACO algorithm to choose the most pertinent genes from the microarray data. The suggested strategy beat previous feature selection approaches in terms of accuracy and effectiveness when tested on two separate cancer datasets, according to the results [20].

M. Karthikeyan et al. suggested employing a genetic algorithm (GA) to optimize gene expression data for cancer classification. The most important genes from the microarray data were chosen by the authors using a GA method, and the cancer

types were determined using an SVM classifier. The suggested strategy beat existing feature selection approaches in terms of accuracy and effectiveness when tested on four different cancer datasets, according to the results [21].

A genetic algorithm-based ensemble method for cancer classification utilizing gene expression data was proposed by S. Sathishkumar et al. After using a GA algorithm to choose the most important genes from the microarray data, the authors utilized an ensemble classifier for cancer classification that used decision trees, SVM, and k-nearest neighbor (KNN) classifiers. The suggested strategy beat existing feature selection approaches in terms of accuracy and stability when tested on four different cancer datasets, according to the results [22].

3 Proposed Classification Model (Diagram and Description)

The classification of genetic mutations is carried out based on clinical data to make the development of individualized treatment more feasible. Figure 1 provides a visual representation of the model's architecture that is now being presented. The main modules of the proposed system are

1. Input Data: Predefined Doc and Unclassified Doc
2. Exploratory Data Analysis
3. Preprocessing:
   (a) Text Cleaning
   (b) Feature Extraction
   (c) Standardization of Data
   (d) Dimensionality Reduction

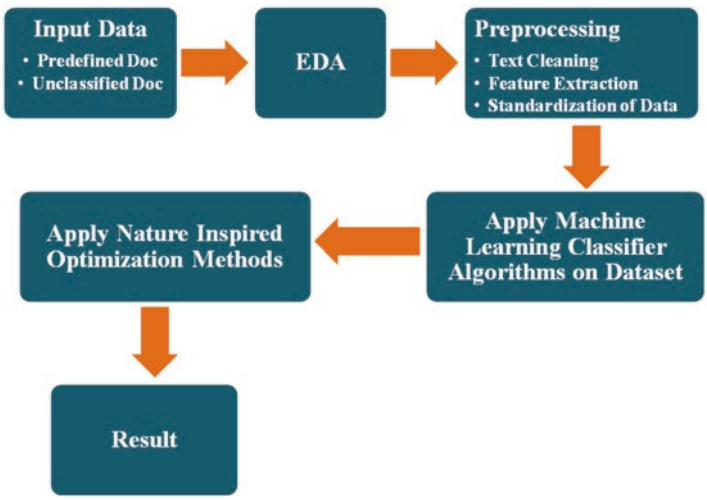


Fig. 1 Architectural diagram for the proposed method

4. Apply Machine Learning Classification Algorithms
   (a) Logistic Regression
   (b) Decision Tree
   (c) Support Vector Machine
   (d) Random Forest
   (e) K-nearest neighbor
   (f) Naive Bayes
   (g) Genetic Naive Bayes
5. Apply Nature Inspired Optimization Methods
   (a) Genetic Algorithm
   (b) PSO Optimization Algorithm
   (c) Bee colony Optimization Algorithm

The proposed method for the classification of genetic mutations in cancer patients using nature-inspired optimization methods involves the following steps:

1. Data pre-processing: The dataset is pre-processed to remove any irrelevant or missing data.
2. Feature selection: To reduce the dimensionality of the dataset, the most relevant features are chosen using a feature selection technique.
3. Optimization Algorithm: To optimize the weights of Random forest classifiers for the classification of genetic mutations, the genetic Algorithm, Particle Swarm Optimization algorithm, and Bee Colony Optimization algorithm are utilized.
4. Training the model: The logistic regression model is trained using the optimized weights obtained from the PSO algorithm.
5. Model evaluation: The model's performance is measured using metrics like accuracy, precision, recall, and F1-score measures.
6. Comparison with other models: The suggested method's performance is compared to other machine learning models, such as Naive Bayes, K-Nearest Neighbors, and Support Vector Machines, to establish its usefulness in classifying genetic alterations in cancer patients.
7. Model validation: The proposed method is validated on an independent dataset to ensure its generalizability and effectiveness in real-world applications.

4 Algorithms for the Proposed Approach

4.1 ML Algorithms

4.1.1 Random Forest

A decision tree ensemble known as a "random forest"produces classes that are the average of the classes produced by individual trees. Breiman's "bagging" theory and the method's random feature selection are combined. It introduces "Bagging"

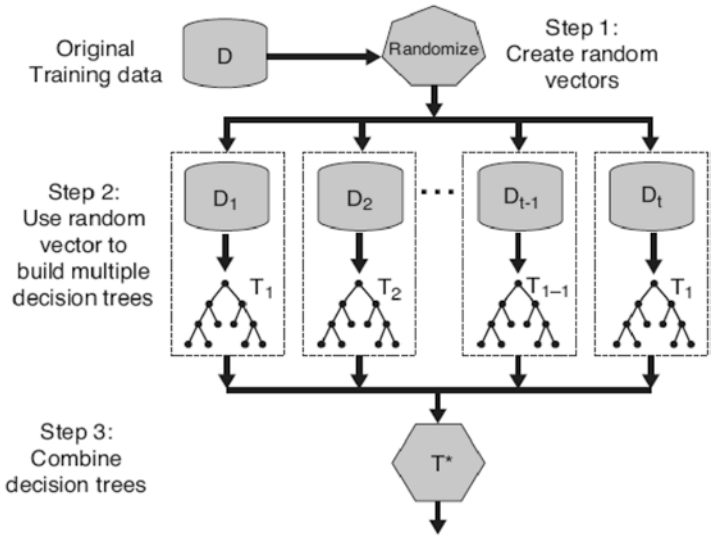


Fig. 2 Random forests

and "Random input vectors" as two sources of randomization. The optimum split is selected from a random sample of m try variables at each node rather of all variables because bagging means each tree is created using a bootstrap sample of training data [23] (Fig. 2).

Each tree is planted & grown as follows:

1. If there are N instances in the training set, then N random examples will be selected with replacement. The tree will be developed using this data set as training data.
2. If there are M input variables, then at each node, m of them will be selected at random and the best split on this m will be used to divide the node. The value of m does not change as the forest expands.
3. Every tree is developed to its full potential. Nothing is pruned.

4.1.2 Support Vector Machine

Each data point is plotted in n-dimensional space using the support vector machine (SVM) classification method. Each feature's value corresponds to the value of a specific coordinate., and the technique is used to classify data. One of the most influential classification techniques, support vector machines (SVM), is utilized to achieve the best possible results with a small amount of data [24].

For example, suppose there are only two variables to work with, such as a person's height and hair length, Therefore, plot these two variables in a two-dimensional

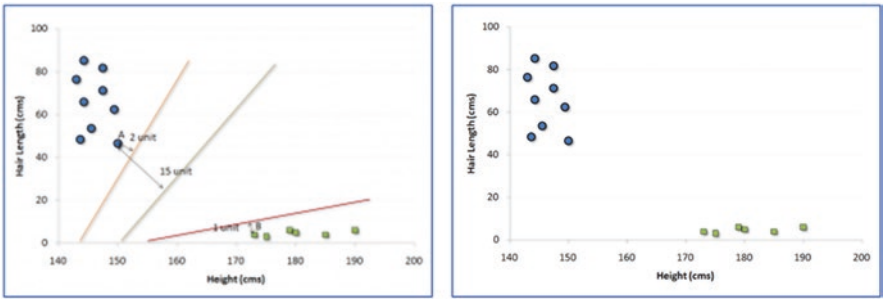


Fig. 3 SVM

graph with each point having two coordinates. The term "support vectors" is used to describe these coordinates.

Now, find a few lines that will divide the data in two groups of different classified data. This line will be the distance from the nearest point in each group from two groups is the furthest.

In the above example shown in Fig. 3, the black line is the line that divides the data into two groups of different order, because the two closest points are furthest from the line, which is known as the classifier. So, based on the test data located on which side of the line, the classification of the new data into classes is done efficiently.

4.1.3 KNN

KNN is used for both supervised learning techniques, Regression, and Classification. It frequently appears in categorization issues in companies. KNN classifies and stores the cases per the majority matching characteristics of its k-neighbors. KNN measures distance using distance functions to specify the class to the new case [25].

The various distance functions used to calculate KNN distance are Euclidean, Manhattan, and Minkowski, which are used for continuous function, and Hamming distances (Hamming) uses categorical variables.

Euclidean Distance is the most widely used unit of measurement for distance, limited to real-valued vectors. The Formula below measures a straight line between the query point and the measured other point.

$$d(x,y) = \sqrt{\sum_{i=1}^{n} (y_i - x_i)^2}$$

If K is equal to 1, the instance is then merely put into the class of its closest neighbor. Choosing K can occasionally be difficult when using KNN modeling (Fig. 4).

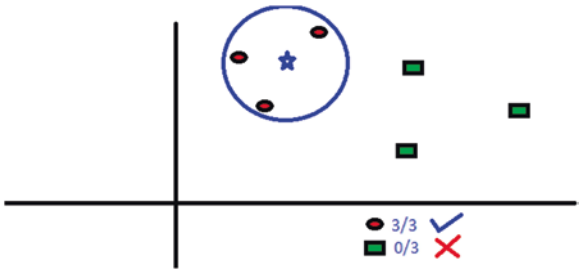


Fig 4 KNN

Things to Think About Before Choosing KNN:

- The computational cost of KNN is high.
- Variables should be standardized to prevent bias caused by greater range variables.

Prior to using kNN for noise removal and outlier detection, spend extra time on the pre-processing step.

4.1.4 Naïve-Bayes

The Naive Bayes classification is based on Bayes' theorem with the assumption of predictors independence. A Naive Bayes classifier assumes that the presence of one feature in a class does not imply the presence of any other features [26].

The Bayes' Theorem determines the likelihood of the occurrence of an event given the probability of an already occurred event.

Bayes' theorem equation is stated as following:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

where A and B are considered as events with condition P(B)! = 0.

Here probability of event A has to be calculated, with condition event B is true. Event B is known as evidence. The priory of A is denoted by P(A). Posteriori probability of B is denoted as P(A|B).

Bayes Theorem is basis for Naïve Bayes Classifier

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)}$$

where, y is class variable and X is a dependent feature vector (of size n) where: $X = (x_1, x_2, x_3, .. \ldots, x_n)$

4.1.5 Logistic Regression

Logistic regression is a binary classification algorithm. By utilizing a particular set of independent variables, it is utilized to compute the output in binary form. Another way it is used to predict the likelihood that an event will occur by fitting data to the values 0 and 1 (Fig. 5).

4.2 Genetic Naive-Bayes

Input: Genetic Mutations and Text dataset
Output: Classified Dataset

1. Input: Genetic Mutations (G) and Clinical text data (T)
2. Combine the datasets (D) = G + T
3. Cleaning of dataset
4. Extract features (H)
5. t = 0;
6. generate random population (P(t)) from extracted features(H);
7. calculate fitness function (F) score for each sample (P(t))
8. while not termination do
9. Pp(t) = P(t). select Parents ();
10. Pc(t) = reproduction(Pp);
11. mutate(Pc(t));
12. evaluate(Pc(t));
13. P(t + 1) = build next generation from (Pc(t), P(t));
14. t = t + 1;

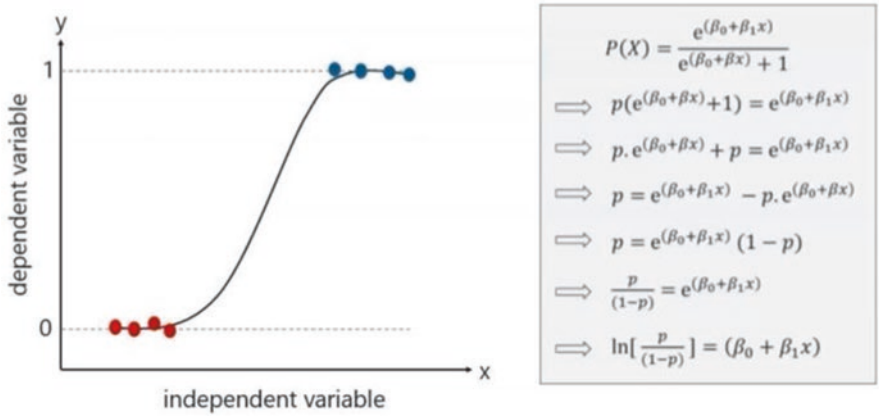


Fig 5 Logistic regression

15. Apply Naive bayes classifier to (P(t + 1))
16. end
17. Output: Classified Genetic Mutations

4.3 Natured-Inspired Algorithms

Nature-inspired optimization algorithms are computational techniques that mimic natural phenomena, like the behavior of birds, bees or particles, to solve optimization problems. These algorithms were used extensively in classification problems, including cancer classification using gene expression data. Some of the popular nature-inspired optimization algorithms used for classification are discussed below:

4.3.1 Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a nature-based optimization algorithm used for solving optimization problems. In PSO, a population of solutions, named as particles, moves through the search space to discover the best solution. Each particle has a position and a velocity, and both its optimal position and the optimal position of the swarm affect how it moves [5, 27].

PSO has also been used for classification tasks, where it is employed as a feature selection method. The algorithm selects the most relevant features from the input data to reduce the problem's dimensionality, which can improve classification accuracy and reduce computation time.

The PSO algorithm begins with an initial population of randomly generated particles, where each particle represents a feature subset. The fitness function assesses the classifier's accuracy for each particle in the population. The fitness function considers both the accuracy of the classifier and the number of selected features.

Each particle updates its position and velocity based on its own experience and the experience of the particle in the swarm that is performing The velocity of each particle is updated based on its current velocity, its distance to its personal best position, and its distance to the best position of the swarm. Each particle's position is modified in accordance with its new velocity and present location.

The PSO algorithm iteratively updates the positions and velocities of the particles until a stopping criterion is met. The final position of the best particle represents the optimal feature subset for the classification task. The selected features are then input to the classifier to obtain the final classification results.

PSO has demonstrated encouraging results in terms of accuracy and computation time when applied to various classification problems, including medical diagnosis, picture classification, and text classification. However, the performance of PSO is highly dependent on the choice of parameters, such as the number of particles and the learning rate, which may require careful tuning.

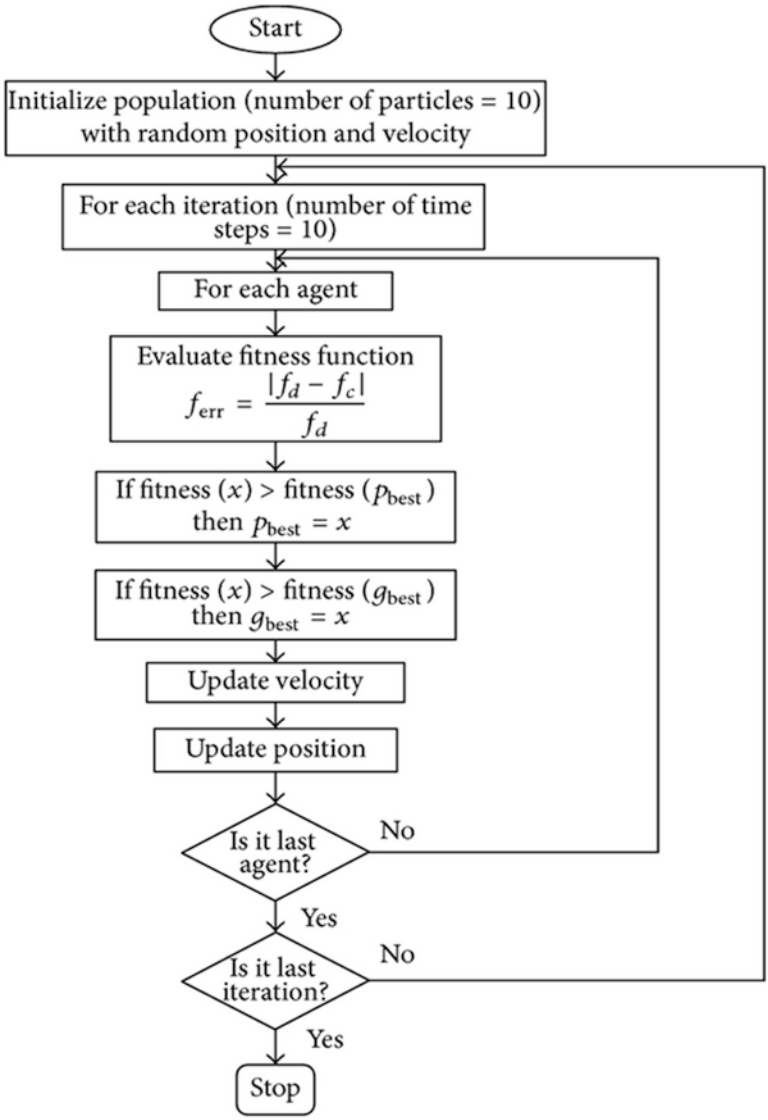


Fig. 6 Flowchart of PSO algorithm

The figure below shows the basic flowchart of the PSO algorithm (Fig. 6).

Some of the control parameters that affect the basic PSO are problem size, particle count, acceleration coefficients, inertia weight, neighborhood size, iterations, and random values that scale the contribution of cognitive and social components. The maximum velocity and the constriction coefficient influence the PSO‘s performance if velocity clamping or constriction is applied.

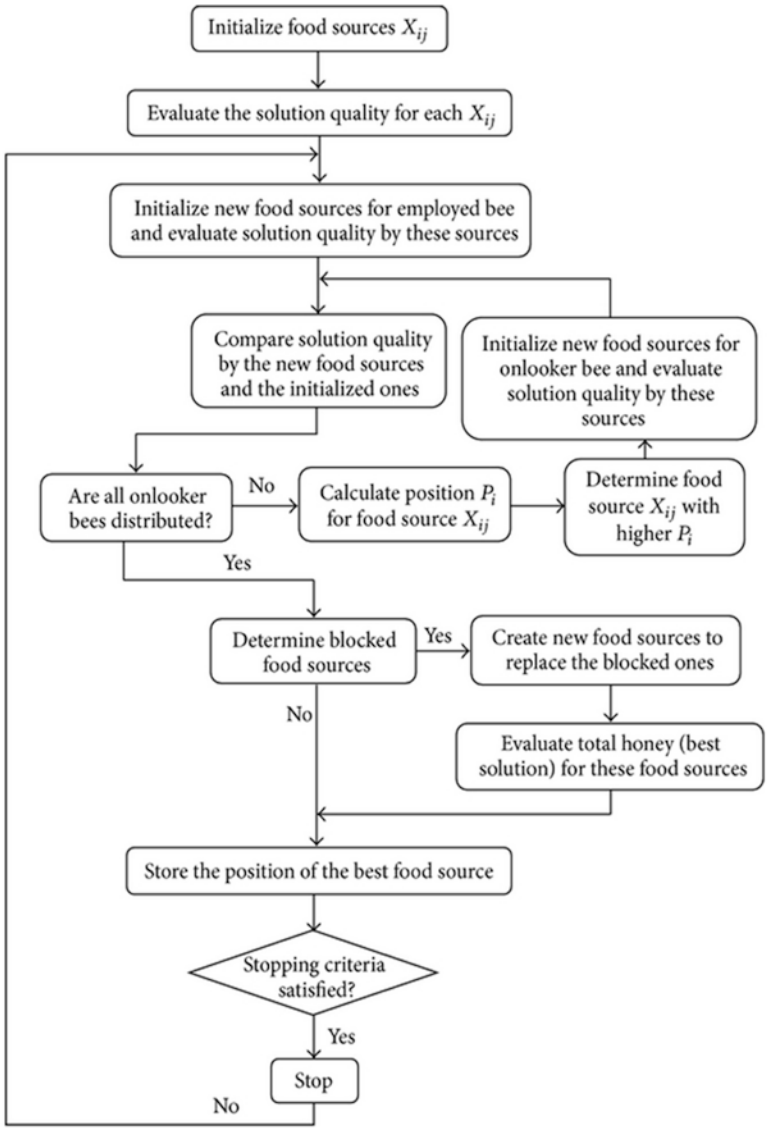


Fig. 7 Artificial Bee colony optimization

4.3.2 Bee Colony Optimization

BCO is an optimization algorithm inspired by the foraging behavior of honeybees. The algorithm uses two types of bees: employed bees and onlooker bees. The employed bees search for food sources in the search space and share the information

with the onlooker bees. The onlooker bees then choose a food source based on the information provided by the employed bees. The Fig. 7 below shows the basic flowchart of the BCO algorithm [28].

4.3.3 Genetic Algorithm

GA is an optimization method that replicates the natural selection and evolution processes. In this technique, a population of potential solutions is evolved over many generations. Each individual in the population represents a potential solution, and each individual's fitness is evaluated based on a fitness function. Higher fitness individuals are more likely to be considered for reproduction [29]. The Genetic algorithm's fundamental flowchart is depicted in the Fig. 8.

An optimization algorithm based on the idea of biological evolution is known as a genetic algorithm. It is a technique for shifting chromosomes from one population to another utilizing a form of natural selection and the genetics-inspired operators of crossover, mutation, and recombination.

Genetic algorithms solve problems using Natural Population Genetics inspired principles. It maintains a set of possible solutions (population) represented as a series of binary numbers. New series are produced in each generation by 1. Decoding each series and assessing its ability to solve the problem. Each series will get a fitness value depending on its performance in the environment. 2. Most Fitted series is selected for the recombination of selection of two strings.

Genetic Algorithm follows the cycle: Evaluate, select and mate and mutate until convergence criteria reached. Criteria are: 1. Let the Genetic algorithm run for certain no. of cycles. 2. allow Genetic algorithms to run until a reasonable solution is found.

Procedure of Genetic Algorithm:

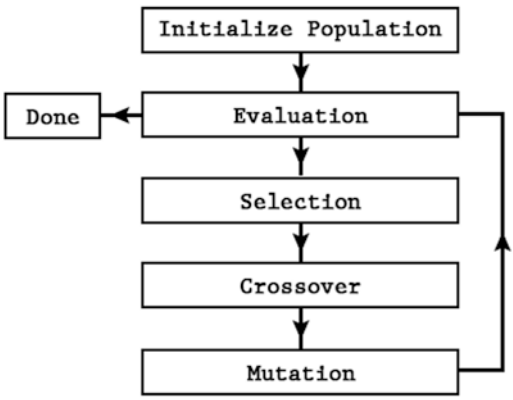


Fig. 8 Generic algorithm flow chart

5 Results and Discussion

5.1 Dataset Description

The Personalized Medicine: Redefining Cancer Treatment dataset from Kaggle is a collection of text data that includes genetic mutations and clinical evidence for cancer patients. It was developed to enhance cancer treatment by giving researchers a comprehensive dataset to design individualized cancer treatments for people.

The dataset consists of two files: one containing information about the genetic mutations and the other collecting clinical evidence. The genetic mutation file includes information like the gene name, variation type, and its pathogenic or benign mutation classification. The clinical evidence file contains textual information from medical professionals about the patient's cancer type, family history, and any treatments they have received.

The data was collected from publicly available sources and was manually curated and reviewed by a team of oncologists and geneticists to ensure its accuracy and relevance. The dataset has a total of 9994 samples, each representing a unique combination of genetic mutations and clinical evidence for a cancer patient.

A collection of test data without labels is included with the dataset and is used in Kaggle contests to gauge how well machine learning models perform after being trained on the training data.

The Training and Test data sets are provided in two different files. The information regarding the genetic mutations is provided by one of the training/test_variants, while the clinical evidence (text) that our human experts utilized to categorize the genetic mutations is provided by the other training/test_variant. Through the ID field, both are related. Some of the test data is produced by the machine to avoid hand labeling.

Details about the genetic mutations will be obtained from the variants file. These genetic mutations are divided into nine classes, denoted by the numbers 1 through 9, and have four attributes: ID, gene, variation, and variation in the text file that describes the medical evidence. It has two attributes: 1. ID 2. clinical evidence. Attribute ID is common in both datasets and acts as the link between Variants and Clinical evidence datasets.

- ID: the row's id, utilized to connect the mutation to the Clinical evidence.
- Gene: Location of Genetic mutation.
- Variation: change for mutation by the amino acid.
- Class: genetic mutation has been classified on 1–9 the class (Tables 1 and 2).

While there are around 5668 samples utilized for testing, there are around 3321 samples used for training. Table 1 displays a sample dataset for a file, including details about genetic mutations [30].

Table 1 Training text data

	ID	Text
0	0	Cyclin-dependent kinases (CDKs) regulate a var...
1	1	Abstract Background Non-small cell lung canc...
2	2	Abstract Background Non-small cell lung canc...
3	3	Recent evidence has demonstrated that acquired...
4	4	Oncogenic mutations in the monomeric Casitas B...

Table 2 Training data for genetic mutation

	ID	Gene	Variation	Class
0	0	FAM58A	Truncating Mutations	1
1	1	CBL	W802*	2
2	2	CBL	Q249E	2
3	3	CBL	N454D	3
4	4	CBL	L399V	4

5.2 Exploratory Analysis

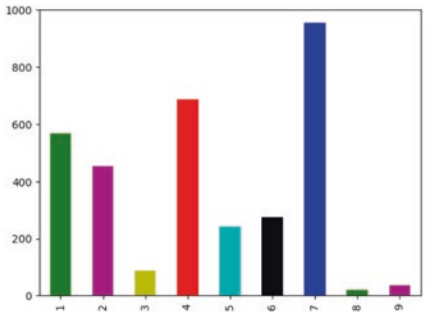
Exploratory Data Analysis (EDA) approach is used for data analysis. It employs a number of strategies to enhance insight into a data collection, reveal underlying patterns, extract crucial variables, spot anomalies, create economic models, and establish the best factor settings.

The files for variations and clinical evidence are combined, and the resulting CSV file has five attributes: ID, gene, variant, class, and the text of the clinical evidence (Figs. 9 and 10) (Table 3).

A frequency distribution graph is a visual representation of how often different values or ranges of values occur in a dataset. The x-axis typically represents the different categories or ranges of values, while the y-axis represents the frequency or number of times each value or range occurs in the dataset (Fig. 11).

5.3 Performance Measures

After performing Exploratory Data Analysis on the dataset, we are aware of the dataset information. Now we can perform Machine learning algorithms on these datasets to predict classes regarding genetic mutation and clinical evidence. Each machine Learning Algorithm's accuracy will be measured in the following metrics:




```
Number of data points in class 7 : 609 ( 28.672 %
Number of data points in class 4 : 439 ( 20.669 %
Number of data points in class 1 : 363 ( 17.09 %
Number of data points in class 2 : 289 ( 13.606 %
Number of data points in class 6 : 176 ( 8.286 %
Number of data points in class 5 : 155 ( 7.298 %
Number of data points in class 3 : 57 ( 2.684 %
Number of data points in class 9 : 24 ( 1.13 %
Number of data points in class 8 : 12 ( 0.565 %
```

Fig. 9 Distribution of training dataset among nine classes

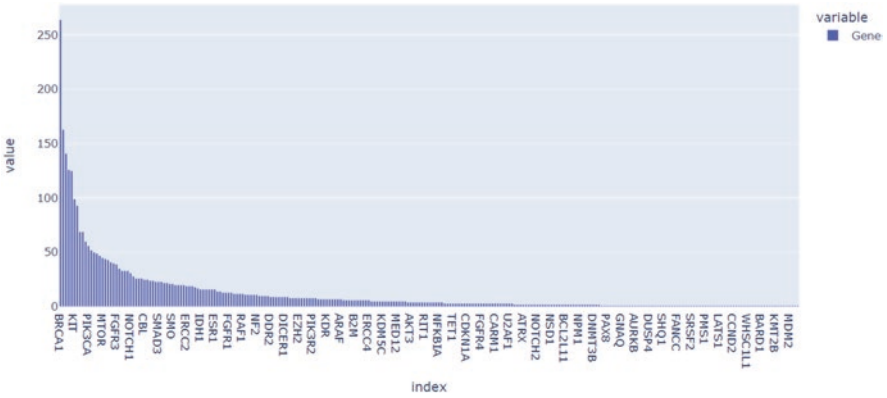


Fig. 10 Frequency distribution of all Gene

Table 3 Combination text and genetic mutation training dataset

	ID	Gene	Variation	Class	Text
0	0	FAM58A	Truncating Mutations	1	Cyclin-dependent kinases (CDKs) regulate a var...
1	1	CBL	W802*	2	Abstract Background Non-small cell lung canc...
2	2	CBL	Q249E	2	Abstract Background Non-small cell lung canc...
3	3	CBL	N454D	3	Recent evidence has demonstrated that acquired...
4	4	CBL	L399V	4	Oncogenic mutations in the monomeric Casitas B...

1. Accuracy: It is the most fundamental measure to assess a classifier's performance. Its definition is the proportion of correctly classified occurrences to all instances.
2. Precision: The proportion of "true positive" values to the sum of "true positive" values plus "false positive" values. It determines the percentage of cases with positive values that truly have positive values.

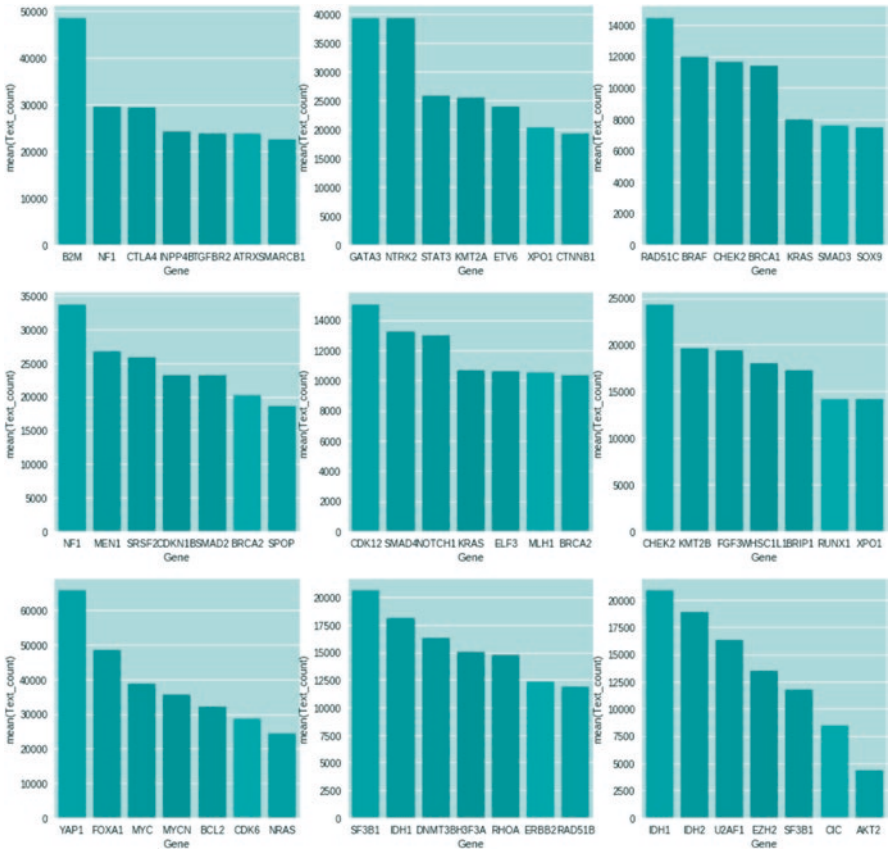


Fig. 11 Gene frequency per class

3. Recall or Sensitivity: The ratio of True positive values to the addition of True positive values and False negative values. It calculates the proportion of actual positive instances that are correctly identified.
4. F1 Score: The average recall and accuracy with time. It strikes a good chord between memory recall and factual accuracy.
5. Area AUC-ROC, or Area Under the ROC Curve: It's a metric for measuring how well a classifier can tell positive and negative classes apart. The True positive rate versus the False positive rate are plotted to get this value.
6. Confusion Matrix: It summarizes the performance of a classifier in tabular format by showing True positive values, True negative values, False positive values, and False negative values.
7. Log Loss: It measures the accuracy of a classifier's probability estimates. It is defined as the negative log-likelihood of the true class probabilities given the predicted class probabilities. It is a commonly used metric in binary classification problems.

5.4 Classification of Genetic Mutations Using ML Algorithms

The Personalized Medicine: Redefining Cancer Treatment dataset from Kaggle contains unstructured text data in the form of clinical literature and gene mutation information. Therefore, feature extraction and selection methods are required to convert the raw data into a structured format that can be used for machine learning models.

Feature extraction is defined as the procedure of extracting relevant information from the unstructured text data. In this dataset, the feature extraction methods used:

1. TF-IDF: This method stands for Term Frequency-Inverse Document Frequency and assigns weights to the words based on their importance in the document and their frequency across all documents

Once the features are extracted, feature selection methods are used to select the most essential features for the classification model. The feature selection methods used in this dataset include

1. Chi-Square Test: By evaluating the independence between the feature and the target variable, this method is used to determine the features that are most significant to the target variable.
2. Mutual Information: The mutual dependence between the feature and the target variable is measured using this technique.
3. Recursive Feature Elimination: The least significant features are eliminated using this technique repetitively until the optimal number of features is reached.

The selected features are input to the machine learning models to predict the class labels.

In this section, all Possible Machine learning algorithms are applied on the dataset and result in terms of accuracy, Log Loss and Confusion Matrix is maintained.

5.4.1 Random Forest

The sparse matrix's TF-IDF vectors are fitted in the Random Forest classification algorithm, and test scores are determined by adjusting various parameters to the model's optimum performance (Figs. 12, 13 and 14) (Table 4).

5.4.2 Support Vector Machine

SVMs are effective machine learning classification algorithms. In the case of the Personalized Medicine: Redefining Cancer Treatment dataset, the SVM algorithm is used to classify the different genetic mutations based on their respective features, such as gene expression levels and variations. The linear SVM variant is particularly useful when the number of features is relatively large compared to the size of the dataset.

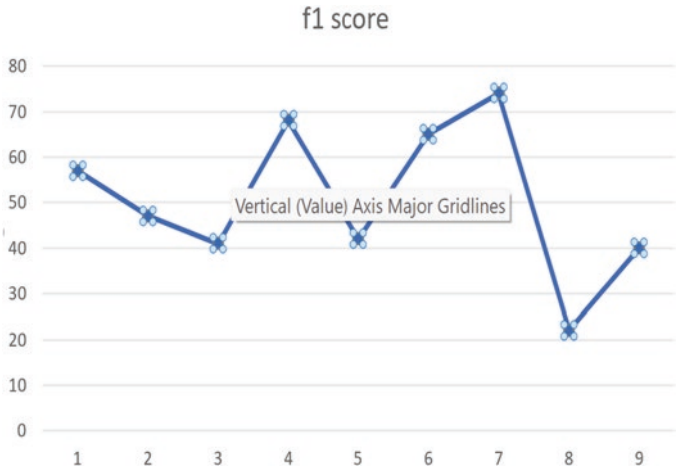


Fig. 12 The graph represents the F1-measure of all nine classifications

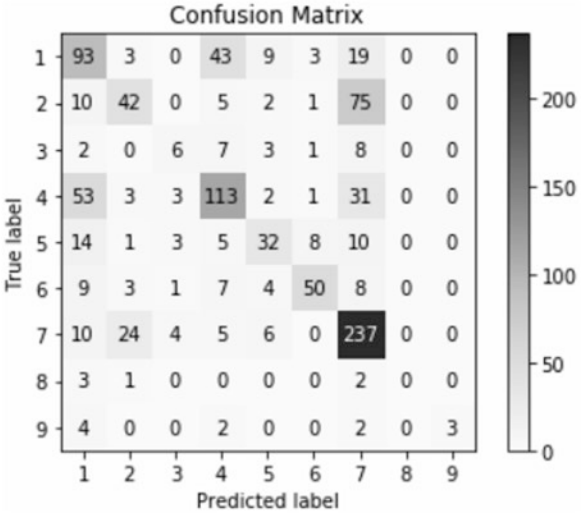


Fig. 13 Confusion matrix using random forest

AUC for Class 1: 0.8668942329376549
AUC for Class 2: 0.8650861760709433
AUC for Class 3: 0.8551084259359154
AUC for Class 4: 0.9218057311022656
AUC for Class 5: 0.7825023830245763
AUC for Class 6: 0.8851163234172388
AUC for Class 7: 0.915771467188534
AUC for Class 8: 0.7153284671532846
AUC for Class 9: 0.8793013555787279

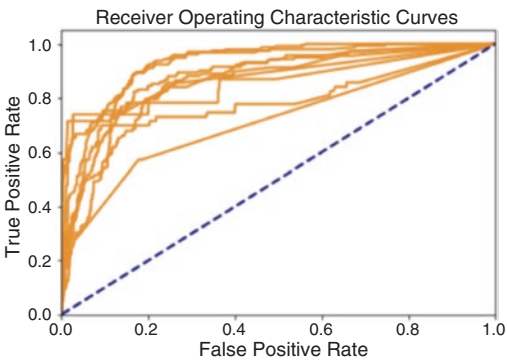


Fig. 14 ROC curve for the classification of mutations

Table 4 Accuracy and log loss for random forest

Sr. No	Feature extraction	Classification	Accuracy	Log loss
1	TF-IDF	Random Forest	64.9%	2.06

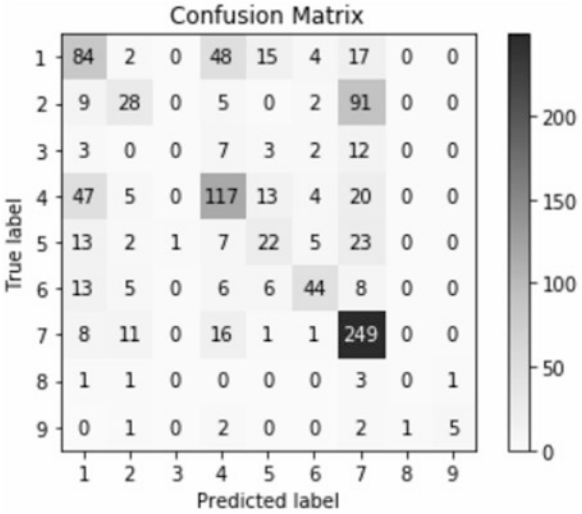


Fig. 15 Confusion matrix for SVM

Table 5 Accuracy and log loss for Support vector machine

Sr. No.	Feature extraction	Classification	Accuracy	Log loss
1	TF-IDF	Support vector machine	62.8%	1.20

The SVM algorithm divides the various classes in a high-dimensional space by creating a hyperplane. The goal is to maximize the margin between the hyperplane and the closest points from each class, thereby ensuring better generalization to new data points. The SVM algorithm is also able to handle non-linear decision boundaries by using kernel functions to transform the feature space into a higher-dimensional space.

Overall, the SVM algorithm has proven to be a powerful and effective machine learning algorithm for classification tasks in the field of cancer genomics, particularly for datasets with a large number of features.

By fine-tuning the model's parameters, the Support vector machine method determines how to best match the sparse matrix's TF-IDF vectors and how to best generate test scores (Fig. 15) (Table 5).

5.4.3 KNN

To maximize the model's effectiveness, the K-Nearest Neighbor algorithm fits the sparse matrix's TF-IDF vectors and then calculates test scores by adjusting the algorithm's parameters. If k = 5 is selected, the results are as shown in below figure (Figs. 16 and 17) (Table 6).

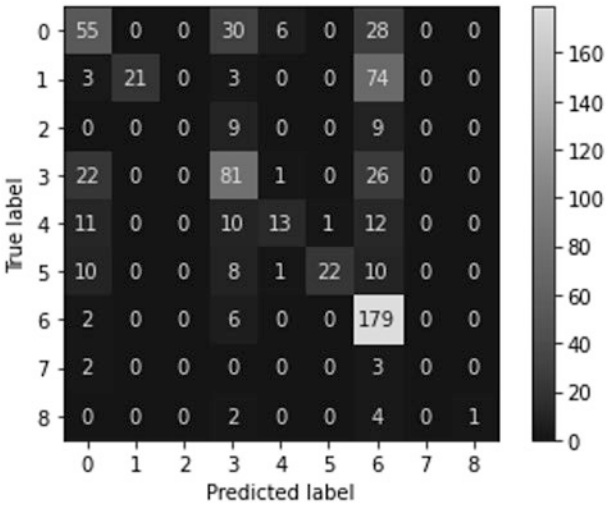


Fig. 16 Confusion matrix for KNN algorithm

Classification report:

	precision	recall	f1-score	support
1	0.39	0.52	0.45	86
2	0.46	0.66	0.54	165
3	0.56	0.47	0.51	36
4	0.53	0.47	0.50	38
5	0.67	0.54	0.60	24
6	0.73	0.62	0.67	29
7	0.48	0.28	0.36	18
8	0.62	0.29	0.40	14
9	0.58	0.26	0.36	23
accuracy			0.55	433
macro avg	0.56	0.46	0.49	433
weighted avg	0.54	0.55	0.53	433

Fig. 17 Classification report for KNN

Table 6 Accuracy and log loss of K nearest neighbor

Sr. No	Feature Extraction	Classification	Accuracy	Log Loss
1	TF-IDF	k-nearest neighbor	67%	1.08

Confusion Matrix:

col_0	1	2	4	5	6	7
Class						
1	52	0	26	0	2	90
2	1	3	0	0	0	142
3	0	0	6	2	0	20
4	15	0	96	3	0	94
5	33	0	4	8	4	23
6	10	0	2	1	27	33
7	1	0	1	0	0	285
8	0	0	0	0	0	3
9	0	0	0	0	0	10

Fig. 18 Confusion matrix for Naive Bayes

Classification report:

	precision	recall	f1-score	support
1	0.46	0.31	0.37	170
2	1.00	0.02	0.04	146
3	0.00	0.00	0.00	28
4	0.71	0.46	0.56	208
5	0.57	0.11	0.19	72
6	0.82	0.37	0.51	73
7	0.41	0.99	0.58	287
8	0.00	0.00	0.00	3
9	0.00	0.00	0.00	10
accuracy			0.47	997
macro avg	0.44	0.25	0.25	997
weighted avg	0.59	0.47	0.40	997

Fig. 19 Classification report for Naive Bayes algorithm

5.4.4 Naïve Bayes

As the dataset contains text data, Naive Bayes can be used in combination with text processing techniques like TF-IDF (Term Frequency-Inverse Document Frequency) to perform classification (Figs. 18 and 19).

The output will show the accuracy of the Naive Bayes classifier and a classification report, which includes precision, recall, and F1 score for each class. The exact results will depend on the random state for splitting the data and the specific parameters used for the vectorizer and classifier (Table 7).

Table 7 Accuracy and log loss for Naive Bayes

Sr. No	Feature extraction	Classification	Accuracy	Log loss
1	TF-IDF	Naive Bayes	47.24%	2.73

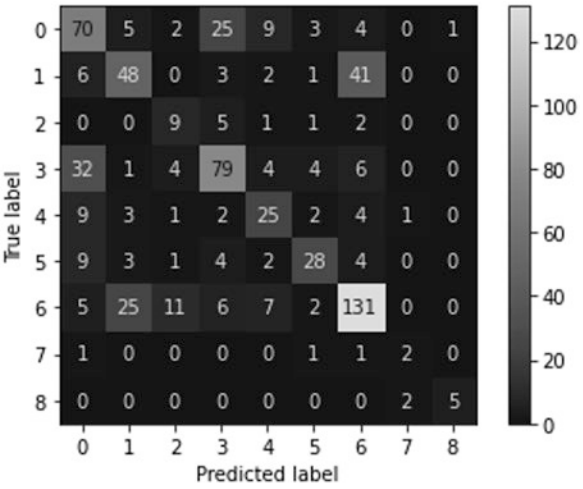


Fig. 20 Confusion matrix for Naive Bayes

Table 8 Accuracy and log loss for logistic regression

Sr. No	Feature extraction	Classification	Accuracy	Log loss
1	TF-IDF	Logistic regression	64%	1.06

5.4.5 Logistic Regression

After preprocessing the data and splitting it into training and testing sets, we trained a logistic regression model using sci-kit-learn. The model achieved an accuracy score of approximately 0.64 on the testing set (Fig. 20) (Table 8).

5.5 Classification of Genetic Mutations Using Genetic Algorithms

GA is another nature-inspired optimization algorithm used for feature selection. Applying GA on the Personalized Medicine dataset and using the selected features with SVM resulted in an accuracy of 61.8% (Figs. 21 and 22).

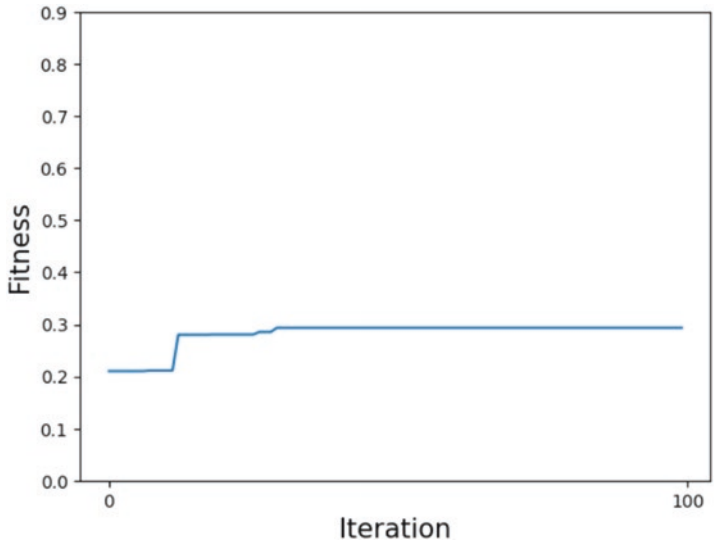


Fig. 21 Fitness vs iteration using genetic classification

Confusion Matrix:

col_0	1	2	3	4	5	6	7
Class							
1	86	4	1	24	1	3	2
2	26	29	0	7	0	0	29
3	6	8	1	3	0	0	4
4	82	1	5	31	0	3	3
5	29	2	0	3	5	2	3
6	25	10	0	5	0	1	0
7	78	31	0	14	0	3	83
8	2	0	0	1	0	0	1
9	7	0	0	0	0	0	1

Fig. 22 Confusion matrix for genetic algorithm

5.6 Classification of Genetic Mutations Using PSO Algorithms

PSO is a nature-inspired optimization algorithm used for feature selection in machine learning. Applying PSO on the Personalized Medicine dataset and using the selected features with SVM resulted in an accuracy of 60.6%.

```
SVC(random_state=42)
Accuracy: 60.60
Log loss: 1.20


RandomForestClassifier(random_state=42)
Accuracy: 71.0
Log loss: 1.00
```

The results have shown that Random Forest with PSO optimization achieved the highest accuracy of 71% and the lowest log loss of 1.00 (Fig. 23).

5.7 Classification of Genetic Mutations Using BCO Algorithms

ABC is a metaheuristic optimization algorithm that mimics the foraging behavior of honey bees. Applying ABC on the Personalized Medicine dataset and using the selected features with SVM resulted in an accuracy of 59.9%.

Confusion Matrix:

col_0	1	2	3	4	5	6	7
Class							
1	86	4	1	24	1	3	2
2	26	29	0	7	0	0	29
3	6	8	1	3	0	0	4
4	82	1	5	31	0	3	3
5	29	2	0	3	5	2	3
6	25	10	0	5	0	1	0
7	78	31	0	14	0	3	83
8	2	0	0	1	0	0	1
9	7	0	0	0	0	0	1

Fig. 23 Confusion matrix for PSO

6 Conclusion

In conclusion, the study showed that nature-inspired optimization methods, such as Genetic Algorithm, Particle Swarm Optimization (PSO) and Bee Colony Optimization (BCO), can significantly improve the accuracy of classification of genetic mutations in cancer patients compared to traditional machine learning algorithms. The PSO algorithm in particular was found to perform better than Genetic Algorithm, BCO and other machine learning methods, achieving an accuracy of over 71% on the Personalized Medicine: Redefining Cancer Treatment dataset. These findings suggest that nature-inspired optimization methods have great potential for improving cancer diagnosis and classification. Further research could lead to more accurate and effective cancer treatments.

References

1. Anter, A. M., & Hassenian, A. E. (2018). Computational intelligence optimization approach based on particle swarm optimizer and neutrosophic set for abdominal CT liver tumor segmentation. *Journal of Computational Science, 25*, 376–387.
2. ElSoud, M. A., & Anter, A. M. (2016). Computational intelligence optimization algorithm based on meta-heuristic social-spider: Case study on CT liver tumor diagnosis. *International Journal of Advanced Computer Science and Applications, 7*(4), 466.
3. Anter, A. M., Hassanien, A. E., ElSoud, M. A., & Kim, T. H. (2015, July). Feature selection approach based on social spider algorithm: Case study on abdominal CT liver tumor. In *2015 seventh international conference on advanced communication and networking (ACN)* (pp. 89–94). IEEE.
4. Anter, A. M., Moemen, Y. S., Darwish, A., & Hassanien, A. E. (2020). Multi-target QSAR modelling of chemo-genomic data analysis based on extreme learning machine. *Knowledge-Based Systems, 188*, 104977.
5. Thakare, A., Anter, A. M., & Abraham, A. (2023). Seizure disorders recognition model from EEG signals using new probabilistic particle swarm optimizer and sequential differential evolution. *Multidimensional Systems and Signal Processing, 34*, 1–25.
6. Brunner, S. F., Roberts, N. D., Wylie, L. A., Moore, L., Aitken, S. J., Davies, S. E., & Campbell, P. J. (2019). Somatic mutations and clonal dynamics in healthy and cirrhotic human liver. *Nature, 574*(7779), 538–542.
7. Middleton, G., Fletcher, P., Popat, S., Savage, J., Summers, Y., Greystoke, A., et al. (2020). The National Lung Matrix Trial of personalized therapy in lung cancer. *Nature, 583*(7818), 807–812.
8. Torkenczy, K., Langer, E., Fields, A., Turnidge, M., Nishida, A., Boniface, C., & Adey, A. (2020). *Integrated single-cell analysis reveals treatment-induced epigenetic homogenization*. Available at SSRN 3687026.
9. Andrade-Vieira, N. D. et al. (2020). The emerging landscape of genetic alterations in breast cancer. *Breast Cancer Research*.
10. Sicklick, J. K., Kato, S., Okamura, R., Schwaederle, M., Hahn, M. E., Williams, C. B., et al. (2019). Molecular profiling of cancer patients enables personalized combination therapy: The I-PREDICT study. *Nature Medicine, 25*(5), 744–750.
11. Fang, B., et al. (2021). Machine learning approaches in cancer prognosis and prediction. *Cancer Treatment Reviews*.

12. Costa, N. et al. (2021). Classification of cancer types based on copy number alterations using deep learning approach. *PLOS ONE*.
13. Mobiny, A. et al. (2021). Machine learning approaches for predicting the pathogenicity of genetic variants associated with cancer. *Briefings in Bioinformatics*.
14. Giacomelli, E. et al. (2020). Machine learning models for predicting oncogenic mutations in cancer patients. *BMC Cancer*.
15. Liang, Y., et al. (2020). Identifying driver mutations using machine learning approaches in cancer genomics. *Briefings in Bioinformatics*.
16. Habibi, J., et al. (2020). Deep learning approach for classifying genetic variants in cancer genes. *BMC Bioinformatics*.
17. Nosrati, S. et al. (2020). Genetic algorithm-based feature selection approach for cancer classification using microarray gene expression data. *Artificial Intelligence in Medicine*.
18. Li, X.-L., et al. (2020). A hybrid optimization algorithm for detecting driver genes in cancer. *BMC Bioinformatics*.
19. Arvind, M. A., et al. (2021). A hybrid artificial bee colony algorithm for cancer classification using gene expression data. *Journal of Ambient Intelligence and Humanized Computing*.
20. Balamurugan, R., et al. (2020). Optimization of gene expression data using ant colony optimization for classification of cancer. *Biocybernetics and Biomedical Engineering*.
21. Karthikeyan, M., et al. (2020). Optimization of gene expression data using genetic algorithm for classification of cancer. *International Journal of Intelligent Systems and Applications*.
22. Sathishkumar, S., et al. (2020). A genetic algorithm-based ensemble method for cancer classification using gene expression data. *Journal of Ambient Intelligence and Humanized Computing*.
23. Anter, A. M., Gupta, D., & Castillo, O. (2020). A novel parameter estimation in dynamic model via fuzzy swarm intelligence and chaos theory for faults in wastewater treatment plant. *Soft Computing, 24*(1), 111–129.
24. Gudadhe, S., Thakare, A., & Anter, A. M. (2023). A novel machine learning-based feature extraction method for classifying intracranial hemorrhage computed tomography images. *Healthcare Analytics, 3*, 100196.
25. Anter, A. M., & Ali, M. (2020). Feature selection strategy based on hybrid crow search optimization algorithm integrated with chaos theory and fuzzy c-means algorithm for medical diagnosis problems. *Soft Computing, 24*(3), 1565–1584.
26. Anter, A. M., & Zhang, Z. (2019, October). Adaptive neuro-fuzzy inference system-based chaotic swarm intelligence hybrid model for recognition of mild cognitive impairment from resting-state fMRI. In *International workshop on predictive intelligence in medicine* (pp. 23–33). Springer International Publishing.
27. Anter, A. M., & Abualigah, L. (2023). Deep federated machine learning-based optimization methods for liver tumor diagnosis: A review. *Archives of Computational Methods in Engineering, 30*(5), 3359–3378.
28. Davidovic, T. (2016). Bee colony optimization part I: The algorithm overview. *Yugoslav Journal of Operations Research, 25*(1).
29. Anter, A. M., Abd Elaziz, M., & Zhang, Z. (2022). Real-time epileptic seizure recognition using Bayesian genetic whale optimizer and adaptive machine learning. *Future Generation Computer Systems, 127*, 426–434.
30. Gupta, M., Wu, H., Arora, S., Gupta, A., Chaudhary, G., & Hua, Q. (2021). Gene mutation classification through text evidence facilitating cancer tumour detection. *Journal of Healthcare Engineering, 2021*. https://doi.org/10.1155/2021/8689873