International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 16 (2024) pp. 322–346

Article

Polycystic Ovary Syndrome (PCOS) Disease Prediction Using Traditional Machine Learning and Deep Learning Algorithms

Aunik Hasan Mridul [1,*], Nowreen Ahsan [2], Syeda Sadia Alam [3], Sonia Afrose [4], Zakia Sultana [1] and Md. Tanvir Mahmud kafi [1]

[1] Computer Science and Engineering, Daffodil International University, Dhaka 1216, Bangladesh; zakiasultana.cse@diu.edu.bd (Z.S.); tanvir15-3811@diu.edu.bd (M.T.M.k.)
[2] Department of Data Science, Clarkson University, Potsdam, NY 13699, USA; noahsan@clarkson.edu
[3] Computer Science and Engineering, Metropolitan University, Sylhet 3100, Bangladesh; syeda.alam.juti@gmail.com
[4] Department of Nutrition and Food Engineering, Daffodil International University, Dhaka 1216, Bangladesh; Sonia34-315@diu.edu.bd
* Correspondence author: aunik15-2732@diu.edu.bd

Received date: 5 March 2024; Accepted date: 21 March 2024; Published online: 10 July 2024

Abstract: In recent years, there has been a noticeable rise in the prevalence of Polycystic Ovary Syndrome (PCOS), a complex endocrine disorder that affects a significant portion of the population, particularly women of reproductive age. PCOS is characterized by hormonal imbalances, irregular menstrual cycles, and the presence of multiple small cysts on the ovaries. Beyond its reproductive implications, PCOS is associated with various metabolic disturbances, including insulin resistance, obesity, dyslipidemia, and increased risk for type 2 diabetes and cardiovascular disease. To gain a comprehensive understanding of the disease's severity and its multifaceted impact on women's health, distinguishing between standard and affected diagnostic reports is imperative. In this study, we propose the application of algorithmic models to enable early detection and raise awareness of potential health risks associated with PCOS. Our approach is straightforward and well-suited for the prediction of uncomplicated cases of PCOS in real-world scenarios. Our dataset, sourced from various medical databases and clinical records, served as the foundation for our research. We employed a wide array of classifiers, including Artificial Neural Network (ANN), Recurrent Neural Network (RNN), Convolution Neural Network (CNN), Long Short-Term Memory Network (LSTM), Bi-Directional Long Short-Term Memory Network (BLSTM), Random Forest (RF), Logistic Regression (LR), Gradient Boosting (GB), K-Nearest Neighbor (KNN), Adaboost Classifier (ABC), Decision Tree (DT), Support Vector Machine (SVM), Quadratic Discriminant Analysis (QDA), Ridge Classifier (RC), Passive Aggressive (PA), Gaussian Naïve Bayes (GNB), and ensemble techniques, to comprehensively explore and evaluate the predictive capabilities of each model in identifying PCOS and its associated complications. The results yielded notable success, with the Boosted Random Forest (RF) and Support Vector Classifier (SVC) classifier emerging as the most accurate, boasting an impressive accuracy rate of 98.278%. Furthermore, the Stacking Classifier RDAS exhibited an accuracy of 99.32%. Our optimization efforts, which included hyperparameter tuning, further enhanced the performance of each classifier. Based on extensive experimentation and a review of contemporary research, our findings unequivocally endorse the Random Forest (RF) and Support Vector Classifier (SVC) boosting classifier as exceptionally proficient, demonstrating a remarkable accuracy rate of 99.32% in the precise prediction of PCOS disease.

Keywords: PCOS; bagging; boosting; ensemble; machine learning; deep learning

1. Introduction

Polycystic Ovary Syndrome (PCOS) stands as one of the most prevalent endocrine disorders affecting women worldwide. With its intricate interplay of hormonal imbalances, metabolic disturbances, and reproductive irregularities, PCOS presents a multifaceted clinical challenge. This syndrome, characterized by its complex pathophysiology and diverse clinical manifestations, poses significant diagnostic and management dilemmas for healthcare providers. As a syndrome with variable clinical presentation and evolving diagnostic criteria, PCOS has garnered increasing attention in recent decades, not only due to its high prevalence but also due to its profound impact on women's health and quality of life.

At its core, PCOS is a heterogeneous disorder, encompassing a spectrum of symptoms and metabolic abnormalities that extend beyond its reproductive manifestations. Historically recognized for its association with menstrual irregularities, anovulation, and infertility, PCOS now stands as a syndrome with systemic implications, affecting various organ systems and posing long-term health risks. The hallmark features of PCOS include hyperandrogenism, ovarian dysfunction, and polycystic ovarian morphology, although the presentation of these features can vary widely among affected individuals. Furthermore, the syndrome is frequently accompanied by metabolic disturbances, including insulin resistance, dyslipidemia, obesity, and an increased risk for type 2 diabetes mellitus and cardiovascular disease. These metabolic aberrations not only exacerbate the reproductive manifestations of PCOS but also contribute to its long-term morbidity and mortality, underscoring the importance of early detection and comprehensive management strategies.

Given the heterogeneous nature of PCOS and its varied clinical manifestations, accurate diagnosis and effective management pose significant challenges for healthcare providers. The diagnosis of PCOS relies on a combination of clinical, biochemical, and imaging criteria, as outlined by various expert consensus guidelines. However, the lack of a singular diagnostic test and the evolving nature of diagnostic criteria contribute to diagnostic uncertainty and may lead to underdiagnosis or misdiagnosis of the syndrome. Moreover, the clinical heterogeneity of PCOS necessitates a personalized approach to management, tailored to the individual patient's presentation, reproductive goals, and metabolic profile. Thus, there is a growing need for reliable predictive models and decision-support tools that can facilitate early detection, risk stratification, and personalized management of PCOS.

In recent years, the advent of machine learning and artificial intelligence (AI) has revolutionized the field of healthcare, offering innovative solutions for disease prediction, diagnosis, and prognostication. Leveraging vast datasets and sophisticated algorithms, machine learning models have demonstrated remarkable capabilities in identifying complex patterns and relationships within clinical data, thereby enabling accurate disease prediction and risk stratification. In the context of PCOS, machine learning holds immense promise as a tool for early detection, prognosis, and personalized management, offering the potential to enhance clinical decision-making and improve patient outcomes.

By harnessing the power of machine learning algorithms and integrating multidimensional clinical data, researchers aim to develop robust predictive models for PCOS that can effectively discriminate between affected and unaffected individuals, identify at-risk populations, and predict disease progression and associated complications. These models utilize a variety of input variables, including demographic data, clinical symptoms, hormonal profiles, imaging findings, and genetic markers, to generate predictive algorithms that are tailored to the heterogeneous nature of PCOS. Furthermore, machine learning techniques such as feature selection, dimensionality reduction, and ensemble learning enable the extraction of meaningful insights from complex datasets, enhancing the predictive accuracy and interpretability of the models.

The development of accurate predictive models for PCOS holds immense potential to revolutionize clinical practice, offering clinicians valuable tools for early detection, risk assessment, and personalized management of this complex syndrome. By leveraging the vast amount of clinical data available and harnessing the power of machine learning algorithms, researchers can unlock novel insights into the pathophysiology of PCOS, identify novel biomarkers, and develop targeted interventions aimed at mitigating the long-term health risks associated with the syndrome. Moreover, by empowering patients with timely diagnosis and personalized management strategies, predictive models for PCOS can improve patient outcomes, enhance quality of life, and reduce the burden of this prevalent endocrine disorder on healthcare systems globally.

Numerous academic institutions have endeavored to develop machine learning algorithms for disease identification, including PCOS. However, their methods were found to be inadequate and lacking in predictive accuracy. We propose our approach to enhance the body's capacity for disease prognosis. Machine learning methods are broadly categorized into supervised and unsupervised learning. Supervised learning utilizes labeled data to generate outputs from inputs based on predefined relationships, employing the dataset's training data. In contrast, unsupervised learning constructs models

using unlabeled data to uncover latent patterns and information. Our developed technique aims to predict the onset of PCOS in individuals, offering a valuable tool for early detection and proactive intervention.

Our research introduces a model for predicting PCOS, a condition on the rise and significantly affecting our society. In our resource-constrained setting, the scarcity of diagnostic tools and high costs of analysis pose challenges for identifying this ailment through traditional means. To address this issue, we turn to machine learning, leveraging its capabilities to analyze symptoms and enhance diagnostic accuracy, thereby offering a cost-effective solution for our community.

2. Related Works

Machine learning methods play a crucial role in identifying the intricate architecture of PCOS disease, focusing on the evaluation of patient diagnosis reports. Various techniques, such as GS, RF, LR, GB, KN, ABC, and DT, are employed for the exploratory analysis in this field. Researchers have extensively employed a range of models, as discussed in this segment, to enhance our understanding of PCOS.

PCOS condition is one of the most prevalent health issues [1] that young women experience. Women of reproductive age are affected by PCOS illness, a complex health issue that may be recognized by a variety of symptoms and markers. The first step towards receiving the right therapy for PCOS is accurate identification and detection of the condition. To identify PCOS patients, researchers used a variety of machine learning techniques, including logistic regression, random forest, SVM, CART, and naive bayes classification. After comparing the outcomes, the Random Forest algorithm performed well in PCOS diagnosis on the provided dataset, achieving 96% accuracy [2].

Machine learning algorithms were used to a dataset of 541 individuals, 177 of whom suffer from PCOS. There are 43 features in the dataset. Since each feature was not equally important, researchers ranked each feature based on its value using a feature selection model known as the univariate feature selection model. Ten highly ranked characteristics that may be utilized to forecast the PCOS illness are obtained by implementing this algorithm. Several algorithms were used to obtain a result after dividing the dataset into the train and test halves. These models comprise logistic regression classifiers, random forest classifiers, gradient boosting classifiers [3], and RFLR, an acronym for logistic regression plus random forest. Consequently, the suggested RFLR algorithm classified the PCOS patients using 10 highly rated characteristics with an accuracy score of 90.01% [4].

In 2021, a novel method was put up for the early diagnosis and identification of PCOS. The suggested model was built using catBoost and XGBRF. The univariate feature selection approach was used to choose the top 10 features after the data had undergone preprocessing. The MLP, decision tree, SVM, HRFLR, random forest, logistic regression, and gradient boosting classifiers were used to compare the accuracy results. Based on the results, catBoost outperformed with a 95% accuracy score, whereas XGBRF performed with an 89% accuracy score. Other classifiers' accuracy values ranged from 76% to 85%. The most effective model for PCOS illness early detection was the catBoost method [5].

Studies have shown that morphological, biochemical, clinical, and methodological factors all play a role in the diagnosis of PCOS [6,7]. Ultrasonography and other modern technologies have made the excess follicle a crucial marker of polycystic ovarian morphology (PCOM). The majority of researches have been using the inception of twelve follicles (diameter measuring 2 to 9 mm) per whole ovary since 2003. That seems out of date today, but [8]. Variations in the volume of the ovaries or their spacing may also be recognized as reliable markers of PCOS morphology. Their efficacy in comparison to overweight and excess follicles is yet unclear, nevertheless.

Researchers examined the traits and features of female genes linked to PCOS in a certain pattern and order for the first time. Participating in the prediction method were the 233 PCOS patients. In order to predict PCOS by discovering novel genes, researchers employed machine learning techniques such decision trees and SVM with a variety of kernel characteristics (linear, polynomial, RBF), as well as k-nearest neighbor (KNN). Out of all these classifiers, SVM (linear) was the highest in terms of accuracy, scoring 80%, whereas KNN accuracy ranged from 57% to 79% [9].

A statistic states that one in three to four out of ten women are now experiencing PCOS difficulty. The authors suggested an automated method that can identify and forecast PCOS illness for medical therapy in order to detect and predict PCOS in the first phase. Five machine learning models were used by the authors: logistic regression, random forest, SVM, k-neighbors, and Gaussian naïve Bayes. They applied models to a dataset of forty-one characteristics. Through the use of statistics, the top 30 features were chosen. The random forest model's accuracy was found to be 90% after comparing the output of all five models, whereas the other models' accuracy ranged from 86% to 89%. The recommended method to identify and forecast PCOS was the random forest model [10].

A mixed machine learning approach for classifying gene expression in bioinformatics was suggested [11]. The suggested genetic model is predicated on an artificial bee colony (ABC) and a cuckoo search algorithm. Using the six benchmark gene expression dataset, a naïve bayes classifier was constructed.

When compared to feature selection methods that have already been published, the study results in good accuracy performance. A fresh paradigm for the categorization of cancer based on gene expression was suggested [12]. For the classification job, the ABC-based modified metaheuristics optimization strategy was used.

This work [13] suggested a new immune infiltrate and possible biomarker for PCOS diagnosis. The logistic regression and support vector machine models based on machine learning were the suggested method. The models were trained and tested using the five datasets. The suggested model's accuracy score for PCOS detection was 91%. The research aids in the presentation of an original framework for analysis. This work presented a modified PCOS-related genes analysis based on mutational landscape screening [14]. For examination, the nsSNP gene data associated with PCOS out of the 27 were chosen.

The comparative analysis showed in Table 1.

Tablel 1. Comparative analysis with previous work.

SL No	Author Name	Used Algorithm	Best Accuracy with Algorithm
1.	M. M. Hassan and T. Mirza [2]	Gaussian Naïve Bayes (NB), Random Forest (RF), Support Vector Classifier (SVC), CART and Logistic Regression (LR)	RF = 96%
2.	S. Bharati, P. Podder and M. R. H. Mondal [4]	Random Forest (RF), Logistic Regression (LR) and RFLR	RFLR = 90.01%
3.	Bhat, S. A. [5]	XGBRF and catBoost	XGBRF = 89%, CatBoost = 95%
4.	X.-Z. Zhang, Y.-L. Pang, X. Wang and Y.-H. Li [9]	K- Nearest Neighbor Classifier (KNN), Support Vector Classifier (SVC)	SVC= 80%
5.	V. Thakre [10]	Gaussian Naïve Bayes (NB), Random Forest (RF) and Logistic Regression (LR)	RF =90%
6.	S. Dhar, S. Mridha and P. Bhattacharjee [14]	Logistic Regression (LR), Support Vector Classifier (SVC)	LR, SVC = 91%
7.	Our Proposed model	RF, LR, GB, KKN, ABC, and DT and GS, Bagging, Boosting, Stacking, Voting, ANN, CNN, RNN, LSTM, BLSTM	Boosted RF and SVC = 99.32%

3. Classifier and Ensemble Models

In this paper, a supervised learning method based on training and testing was utilized. The classification model was constructed using the training dataset, where the algorithm learned patterns and relationships within the data. Subsequently, the trained model was applied to the testing dataset to predict outcomes or classify new instances. The specific deep learning and machine-learning algorithm employed in this study will be elaborated upon in the subsequent sections. Some of the classifiers we've developed include ANN, CNN, RNN, LSTM, BLSTM, SVM, GNB, RF, LR, GB, KN, ABC, RC, PA, QDA and DT methods.

Artificial Neural Network (ANN)

Artificial Neural Networks (ANNs) are a fundamental component of machine learning, inspired by the intricate structure and functioning of the human brain. ANNs are versatile models that excel in tasks ranging from pattern recognition to complex decision-making. Comprising interconnected nodes, or artificial neurons, organized into layers, ANNs process information through weighted connections, mimicking the synaptic strengths in biological neural networks. In an ANN, the input layer receives data, and subsequent hidden layers transform this input using learned weights. The output layer then produces the final prediction or classification. What sets ANNs apart is their ability to adapt and learn from data.

Through a process known as training, ANNs adjust the weights between neurons based on the provided data and the desired output. This adaptability enables ANNs to generalize patterns and make predictions on new, unseen data. Deep learning, a subset of machine learning, has gained prominence with the development of deep neural networks, characterized by multiple hidden layers. This depth allows ANNs to automatically extract hierarchical features from data, making them powerful tools for tasks such as image and speech recognition. ANNs have proven effective in diverse domains, from natural language processing to medical diagnostics, showcasing their significance in advancing artificial intelligence and solving complex real-world problems.

Recurrent Neural Network (RNN)

Recurrent Neural Networks (RNNs) represent a category of artificial neural networks designed for processing sequential and temporal data. What sets RNNs apart from traditional feedforward neural networks is their unique ability to capture dependencies and patterns within sequences, making them particularly effective for tasks involving time-series data, natural language processing, and speech recognition. At the core of RNN architecture is the concept of recurrent connections, allowing information to persist within the network across different time steps. This recurrence enables RNNs to maintain a memory of previous inputs, making them well-suited for tasks where context and sequential relationships are crucial. Despite their conceptual strength, traditional RNNs suffer from challenges such as the vanishing gradient problem, limiting their ability to capture long-range dependencies effectively. To address this, variations like Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) have been introduced. These variants incorporate sophisticated gating mechanisms, facilitating better information flow over extended sequences. RNNs find applications in diverse domains, ranging from natural language processing tasks like language modeling and machine translation to time-series analysis in finance and healthcare. While effective, the evolving landscape of neural network architectures continues to refine and extend the capabilities of RNNs, ensuring they remain instrumental in modeling sequential data and understanding temporal relationships.

Convolutional Neural Network (CNN)

Convolutional Neural Networks (CNNs) represent a class of deep learning models designed for processing structured grid data, particularly images. They have gained immense popularity for their remarkable success in image recognition, classification, and feature extraction tasks. Unlike traditional neural networks, CNNs are equipped with specialized layers, such as convolutional and pooling layers, which enable them to automatically and adaptively learn hierarchical representations of input data. The core innovation of CNNs lies in the convolutional layers, where filters or kernels systematically slide across input images, capturing local patterns and features. This spatial hierarchy allows CNNs to recognize complex patterns by learning low-level features in the initial layers and progressively combining them to form higher-level abstractions in subsequent layers. Pooling layers further contribute to translation invariance by reducing spatial dimensions while retaining essential information. CNNs have revolutionized computer vision, demonstrating unparalleled performance in tasks like image classification, object detection, and facial recognition. Their applications extend beyond images to fields like natural language processing and speech recognition, showcasing the versatility and efficiency of CNN architectures. The success of CNNs underscores their significance in pushing the boundaries of machine learning and artificial intelligence, making them a foundational technology in modern computational advancements.

Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) designed to address the challenge of learning and remembering long-term dependencies in sequential data. Introduced by Hochreiter and Schmidhuber in 1997, LSTMs have become a cornerstone in the field of deep learning, particularly for tasks involving sequential information, such as natural language processing and time series prediction. What sets LSTMs apart from traditional RNNs is their unique architecture, featuring memory cells with self-connected gates. These gates enable the network to regulate the flow of information, selectively remembering or forgetting past states, thereby mitigating the vanishing gradient problem associated with standard RNNs. The architecture includes input, forget, and output gates, allowing LSTMs to capture and retain relevant information over extended sequences. LSTMs excel in modeling temporal dependencies, making them well-suited for tasks where understanding context and capturing long-range dependencies is crucial. Their ability to effectively handle vanishing gradient issues has made LSTMs instrumental in diverse applications, from speech recognition to machine translation. With their capacity to retain contextual information over extended periods, LSTMs have significantly contributed to the advancement of deep learning models for sequential data analysis.

Bidirectional Long Short-Term Memory (BLSTM)

Bidirectional Long Short-Term Memory (BLSTM) is a sophisticated neural network architecture designed to capture intricate dependencies and patterns in sequential data, making it particularly effective in tasks involving time series or sequential information. A variant of the Long Short-Term Memory (LSTM) network, BLSTM enhances predictive capabilities by processing input data in both forward and backward directions. The key innovation lies in its bidirectional nature, enabling the model to consider past and future context simultaneously. This bidirectional processing is crucial in understanding temporal relationships and dependencies within a sequence, making BLSTM well-suited for applications such as natural language processing, speech recognition, and medical time series analysis. By incorporating memory cells and gating mechanisms, BLSTM can effectively capture long-range dependencies in sequential data, mitigating issues like vanishing gradients that often hinder traditional recurrent neural networks. This bidirectional approach allows the model to learn from past and future context, improving its ability to predict and analyze sequential patterns. BLSTM has proven valuable in various domains where understanding the context of data over time is essential, making it a powerful tool in the realm of deep learning and sequential data analysis.

Random Forest

The Random Forest classifier is a powerful and versatile machine learning algorithm that has gained immense popularity for both classification and regression tasks. It operates by creating an ensemble of decision trees, where each tree is constructed using a random subset of the training data and a subset of the available features. This technique introduces variability and decorrelates the individual trees, mitigating overfitting and improving the model's generalization performance. In classification, the Random Forest combines the results from these decision trees through a majority vote, while in regression, it computes the average of the individual tree predictions. One of the key advantages of Random Forest lies in its ability to handle high-dimensional data, maintain robustness against outliers, and provide feature importance for model interpretability. The algorithm is less prone to overfitting compared to single decision trees, thanks to its inherent bagging (Bootstrap Aggregating) and feature bagging components. Random Forest is particularly useful when dealing with complex and noisy datasets, and it's less sensitive to hyperparameter tuning than other algorithms. Additionally, the Random Forest can identify influential features and provide insights into their contribution to the model's predictive power. Its robust performance, scalability, and flexibility have made it a popular choice across various domains, including finance, healthcare, and image analysis. However, the trade-off for its power and versatility is increased computational cost and complexity, which can be a consideration for real-time or resource-constrained applications. Nonetheless, the Random Forest remains a reliable workhorse in machine learning, delivering accurate predictions and valuable insights for diverse problem-solving scenarios [15].

Decision Tree

To assign a classification to an instance, we start by examining the feature represented by the base of the tree node. Then, we follow a branch of the structure that corresponds to the value of that feature. The Decision Tree technique, which just needs two number Classes, is one of the most effective and well-known prediction techniques. Each inner node of a decision tree, a structure of data with an ordered structure where every node in the leaf hierarchy denotes a distinct class, represents an attribute test. On the basis of decision trees, a tree structure known as DT is frequently utilized. The approach may be used to solve classification and regression issues. As the tree grows from the root node, the "splitting" procedure is utilized to select the “Best Features” or “Best Attributes” from the prospective characteristics pool. It is typical to compute two extra metrics, “Entropy”, as indicated in (1), and “Data Gain”, as mentioned in (2), in order to find the “Best Attribute” [16]. Entropy analyzes the consistency of a dataset, whereas collecting data measures the pace at changes that occur in the volatility of attributes.

$$E(D) = -P\,(positive) log 2P\,(positive) - P(negative) \log 2P\,(negative) \quad (1)$$

$$Gain\,(Attribute\ X) = Entropy\,(decision\ Attribute\ Y) - Entropy\,(X, Y) \quad (2)$$

Naïve Bayes

The term “GNB” refers to a group of Bayes’ Theorem-based algorithms for classification that calculate the probability of an event happening given the probability that another event could also happen. Each algorithm in this group is predicated on the fundamental tenet that any two attributes being identified are unrelated to each other (Equation (3)).

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (3)$$

The constant value is taken to represent a Gaussian distribution for every characteristic in Gaussian NB. The term "Normal distribution" is often used interchangeably w_(i^th).

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma y^2}} \exp\left(-\frac{\left(x_i - \mu_y\right)^2}{2\pi\sigma y^2}\right) \quad (4)$$

Logistic Regression

Logistic Regression is a widely utilized and interpretable machine learning classifier that excels in binary and multiclass classification tasks. Unlike linear regression, which predicts continuous values, logistic regression models the probability of an instance belonging to a particular class using the logistic function (sigmoid). It estimates the odds of an event occurring and maps them to a range between 0 and 1, allowing it to provide clear class separation. The model is trained by minimizing the logistic loss or cross-entropy loss through iterative optimization techniques like gradient descent. Logistic Regression is advantageous for its simplicity, quick training, and ease of interpretation. It can handle both linear and non-linear relationships between features and the target variable through polynomial or interaction terms. While primarily a binary classifier, it can be extended to multiclass problems through techniques like one-vs-rest or softmax regression. One limitation is its susceptibility to overfitting when dealing with high-dimensional data or complex relationships, which can be mitigated through regularization techniques like L1 (Lasso) or L2 (Ridge) regularization. Despite its simplicity, logistic regression is a valuable tool in various domains, including healthcare (predicting disease outcomes), finance (credit risk assessment), and natural language processing (text classification), and it serves as a foundational model in many machine learning pipelines due to its transparency and effectiveness [17,18].

Support Vector Machine

Regression and classification problems may both be resolved using the Support Vector Classifier (SVC). However, categorization issues are where artificial intelligence is most frequently applied. The SVM approach looks for a straight line, or judgment limit, that divides the region into categories in all n variables in order to properly categorize fresh data points. A hyperplane is this highest utility bound. Using SVM, which chooses the most extreme locations and vectors, a hyperplane may be created. As a result, the word "support vector", which is used to describe these severe situations, is where the technique's name, "support vector machine", comes from [18].

Gradient Boosting

The Gradient Boosting Classifier is a powerful and versatile machine learning algorithm that excels in predictive modeling, particularly in classification tasks. It operates by iteratively building a strong predictive model through the combination of multiple weak models, typically decision trees, in a sequential manner. At each iteration, the algorithm focuses on the misclassified data points from the previous stage, assigning them greater importance. This iterative process allows the algorithm to continuously refine its predictions, ultimately creating a robust ensemble model. One of the key advantages of the Gradient Boosting Classifier is its ability to handle complex, high-dimensional data and capture intricate relationships between variables. By combining the outputs of multiple weak learners, it can achieve superior predictive performance. However, this power comes at a computational cost, and training a Gradient Boosting model can be more time-consuming compared to some other algorithms. To mitigate the risk of overfitting, careful hyperparameter tuning and cross-validation are essential when implementing Gradient Boosting. The choice of the learning rate, the number of boosting iterations (trees), and the maximum depth of trees are critical factors that influence the model's performance. In practice, Gradient Boosting is widely used in various fields, including data mining, finance, and biology, due to its effectiveness in addressing complex classification challenges and producing accurate results. Its versatility and robustness make it a valuable tool for both beginners and experienced data scientists aiming to tackle a wide range of classification tasks [19].

K-Nearest

The K-Nearest Neighbors (KN) classifier is a widely used and intuitive machine learning algorithm for classification tasks. It operates on the principle that similar data points tend to belong to the same class. In the KN algorithm, an input data point is classified based on the majority class among its K nearest neighbors in the feature space. The choice of K, the number of neighbors to consider, is a critical hyperparameter that impacts the algorithm's performance. KN is non-parametric and does not make strong assumptions about the underlying data distribution, making it applicable in various scenarios. Its

simplicity and ease of implementation make it a popular choice for introductory machine learning tasks. However, KN's computational efficiency can be a limitation for large datasets, as it requires calculating distances between the data point in question and all other data points in the dataset. Moreover, KN's performance is sensitive to the choice of distance metric, and the curse of dimensionality can affect its accuracy as the number of features or dimensions increases. To address these challenges, techniques such as feature selection, dimensionality reduction, and careful hyperparameter tuning are often employed in conjunction with KN. Despite its limitations, KN remains a valuable tool for many classification problems, particularly when the dataset is manageable in size and the algorithm's assumptions align well with the underlying data distribution.

Adaboost

The AdaBoost (Adaptive Boosting) classifier is a powerful ensemble learning method designed to enhance the performance of weak classifiers by combining them into a robust and accurate model. AdaBoost operates iteratively, sequentially adjusting the weight of each training instance based on the accuracy of the previous weak classifiers. This means that instances that are misclassified receive higher weights, allowing subsequent weak classifiers to focus on them and improve their classification accuracy. The final prediction is then made by combining the weighted outputs of these weak classifiers. One of AdaBoost's strengths lies in its adaptability to different classification problems, as it can work with a wide range of base classifiers, typically decision stumps or shallow decision trees. It's particularly effective in addressing complex datasets and overcoming issues such as overfitting, as it gives more emphasis to challenging data points during training. Moreover, AdaBoost is known for its ability to handle high-dimensional feature spaces effectively. While AdaBoost is a powerful algorithm, it's not immune to outliers or noisy data, which can adversely affect its performance. However, its capacity to mitigate these issues is strengthened by its sequential learning process. By leveraging AdaBoost's combination of weak learners, it often results in a strong and accurate classifier that is widely used in various fields, including face detection, text classification, and bioinformatics, where high performance and adaptability are essential.

Quadratic Discriminant Analysis

Quadratic Discriminant Analysis (QDA) is a statistical classification technique used in machine learning and pattern recognition. It is an extension of Linear Discriminant Analysis (LDA) and is particularly applicable when the assumption of equal covariance matrices among classes is not met. In QDA, each class is characterized by its own covariance matrix, providing a more flexible model that can better capture the underlying distribution of the data. The goal of QDA is to find the decision boundaries that best separate different classes by estimating the probability distributions of the input features for each class. QDA models the likelihood of a data point belonging to a specific class using a quadratic decision boundary, allowing for more complex relationships between variables compared to linear boundaries. QDA involves estimating the mean and covariance matrix for each class and then using Bayes' rule to calculate the posterior probability of a data point belonging to each class. During classification, the class with the highest posterior probability is assigned to the data point. While QDA can be effective in capturing non-linear decision boundaries, it requires estimating more parameters, and if the number of features is large, it may lead to overfitting. Choosing between QDA and LDA depends on the underlying distribution of the data and the specific assumptions about covariance matrices. Overall, QDA is a valuable tool for classification problems when class-specific covariances are unequal, and it offers a more flexible approach compared to LDA.

Ridge Classifier

Ridge Classifier is a linear classification algorithm that extends the traditional linear models by incorporating L2 regularization, also known as Ridge regularization. This regularization term is added to the standard linear regression cost function, aiming to prevent overfitting and improve the generalization of the model. In the context of classification, Ridge Classifier is often used for binary or multiclass classification tasks. It applies Ridge regularization to the coefficients of the linear decision boundary, encouraging them to be small. This regularization term is proportional to the squared L2 norm of the coefficients, penalizing large values. The regularization term introduces a trade-off between fitting the training data well and keeping the model parameters small. The strength of regularization is controlled by a hyperparameter, commonly denoted as alpha. Higher values of alpha increase the regularization strength, leading to a simpler model with smaller coefficients. The Ridge Classifier is part of a family of classifiers that leverage regularization techniques to enhance model stability and prevent overfitting. It is particularly useful when dealing with datasets that have multicollinearity issues, where features are correlated. Ridge regularization helps stabilize the model by distributing the impact of correlated features more evenly. Scikit-learn, a popular machine learning library in Python, provides an implementation of Ridge Classifier, making it accessible for practitioners to apply in various classification scenarios.

Passive Aggressive Classifier

The Passive-Aggressive (PA) Classifier is an online learning algorithm designed for dynamic and large-scale datasets. Operating in a lazy learning fashion, it updates its model incrementally as it encounters new data, making it suitable for scenarios with evolving information. The algorithm processes one training example at a time and updates its model when mistakes occur during predictions. The update rule, guided by an aggressiveness parameter, adjusts the model to rectify errors, with higher aggressiveness leading to quicker adaptations. This versatility allows the Passive-Aggressive algorithm to be applied to both classification and regression tasks. Its applications span various domains, including natural language processing and text classification, making it a valuable tool for scenarios with continuous and substantial data influx. Different variants, such as PA-I and PA-II, offer flexibility in adapting to specific characteristics of the data and learning requirements.

Ensemble Learning Algorithms

Ensemble learning refers to the technique of combining multiple machine learning models to improve overall predictive performance and robustness [20].

Bagging Classifier

Bagging is a powerful technique that reduces variance and improves the stability of machine learning algorithms, with a particular focus on decision tree algorithms. By creating multiple subsets of the training data through bootstrapping and training separate models on each subset, bagging helps mitigate issues like overfitting and handling missing variables. The predictions from these individual models are then combined using techniques such as majority voting or averaging to generate an ensemble model. This ensemble model, formed through the combination of diverse model predictions, exhibits enhanced performance and robustness. Bagging is a valuable tool for improving the reliability and accuracy of machine learning algorithms, providing a more robust solution for classification tasks [21].

Boosting Classifier

Boosting is a powerful technique that leverages a weighted average to combine multiple algorithms, transforming weak learners into strong learners and enhancing the accuracy of independent models. This technique focuses on creating loss functions that guide the learning process of the individual models. The concept of boosting is illustrated in highlighting the iterative nature of the algorithm. In our study, we employ the boosting method during the training and testing phase to construct a hybrid model that benefits from the strengths of each individual model. The equation for the boosting algorithm, which captures the iterative nature of the model construction, is depicted below. By utilizing boosting, we can effectively improve the accuracy and performance of our models by iteratively adjusting the weights and combining the predictions of multiple weak learners into a more robust and accurate ensemble model [21,22].

Stacking Classifier

Stacking, also known as stacked generalization, is a distinctive approach in machine learning. It involves exploring multiple models for solving the same problem. The core idea is to address a learning problem by employing various models, with each model focusing on a specific aspect of the problem rather than the entire problem. The crucial aspect is that each of these individual models can produce intermediate predictions. Consequently, we can train a second model that learns the same target using these intermediate predictions. This second model, as the name suggests, is intended to be "stacked" on top of the others. The ultimate goal is to enhance overall performance and typically achieve a model that outperforms each individual intermediate model. Ultimately, stacking trains a single model that aggregates the outputs of multiple algorithms and generates a new prediction. In terms of efficiency, stacking often outperforms any single model [23]. It can be illustrated using logistic regression as a combiner approach to integrate all the existing predictions into a final prediction.

Voting Classifier

A voting classifier is a combination of multiple individual classifiers that predict which class will receive the majority of votes, effectively making predictions through a “majority rules” approach. This technique involves developing several models that predict outcomes, and the final prediction is based on the collective votes from these models. The specific algorithm used for the voting classifier is depicted in the calculations provided in references [24–26].

Flow Chart

we harnessed the power of a process diagram to predict PCOS disease effectively. Our initial steps revolved around the presentation of the training and testing datasets for our system, followed by the implementation of critical data pre-processing methods such as Standard Scaler Transform, Categorical to Numeric conversion, and Feature Selection. The allocation of 80% -20% for training and testing

ensured a robust evaluation process. Subsequently, we executed various deep learning and machine learning algorithms and meticulously assessed their results. To elevate our predictive accuracy to its maximum potential, we turned to ensemble algorithms, encompassing techniques like bagging, boosting, stacking, and voting. This allowed us to extract the most from the combined algorithms and derive results that were comprehensively analyzed. The models employed in this phase were subjected to outcome analysis to determine their effectiveness in predicting PCOS disease. The model, illustrated in Figure 1, encapsulated our research journey, offering insights into the most effective techniques employed in our study.

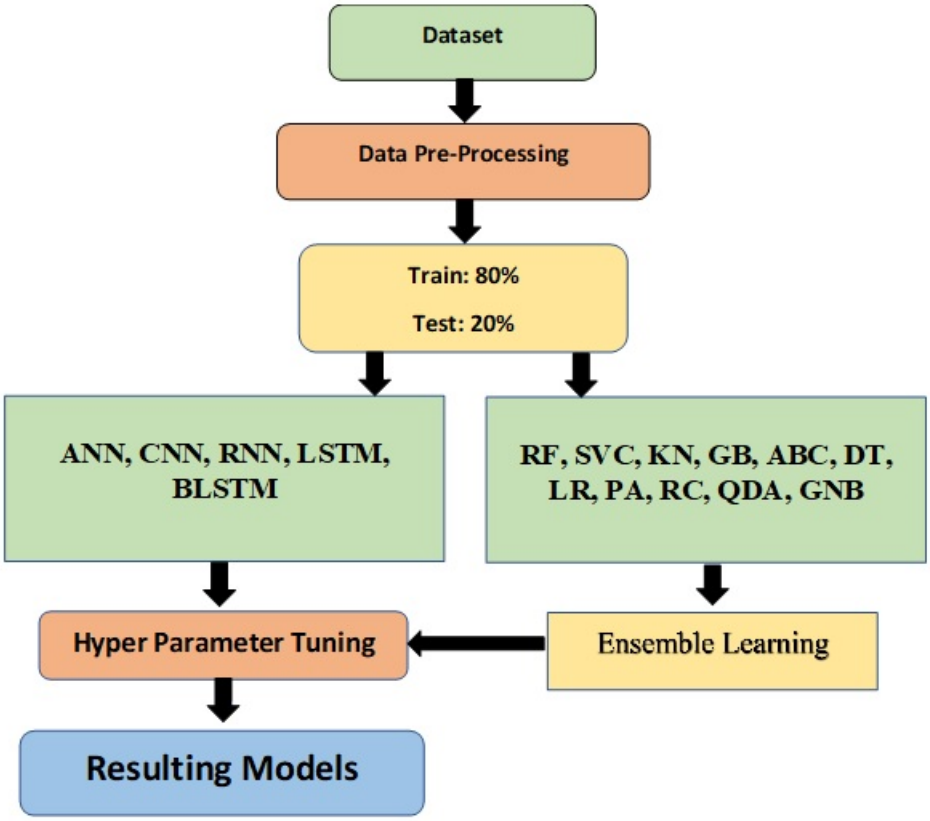


Figure 1. Methodology of PCOS Disease Prediction.

Data collection and Pre-processing

The dataset, sourced from Kaggle [27], comprising 540 rows and 46 columns. Among these columns, the PCOS attribute played a pivotal role in categorizing the prevalence of PCOS disease, while each individual trait proved crucial for identifying this condition. Patients were classified into two groups denoted by 0 and 1, representing the occurrence and absence of PCOS, with 363 individuals belonging to the former category and 177 to the latter, as depicted in Figure 2. In the Figure 2, the dataset is not balanced. The color red defines the negative of PCOS and blue defines the positive. Figure 3 shows the count chart of target column after balancing. The dataset was further divided into two segments: the training set and the test set. In the training set, 80% of the applicants were selected, while the remaining 20% constituted the test set, facilitating comprehensive model development and assessment for PCOS prediction. We have used ADASYN to balance the target column.

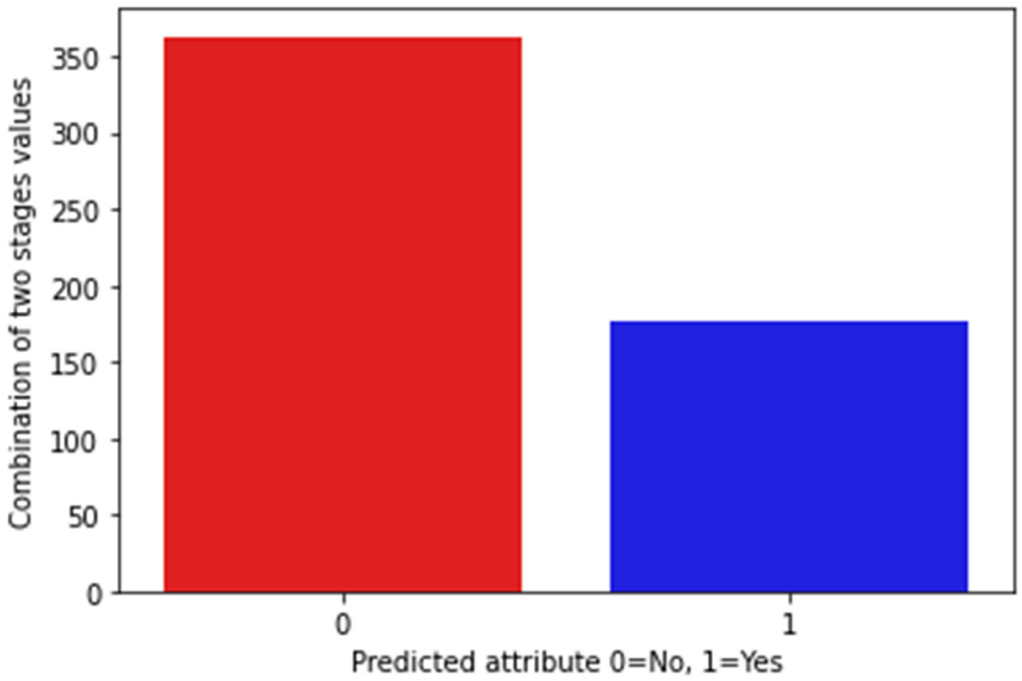


Figure 2. Target column count plot before ADASYN.

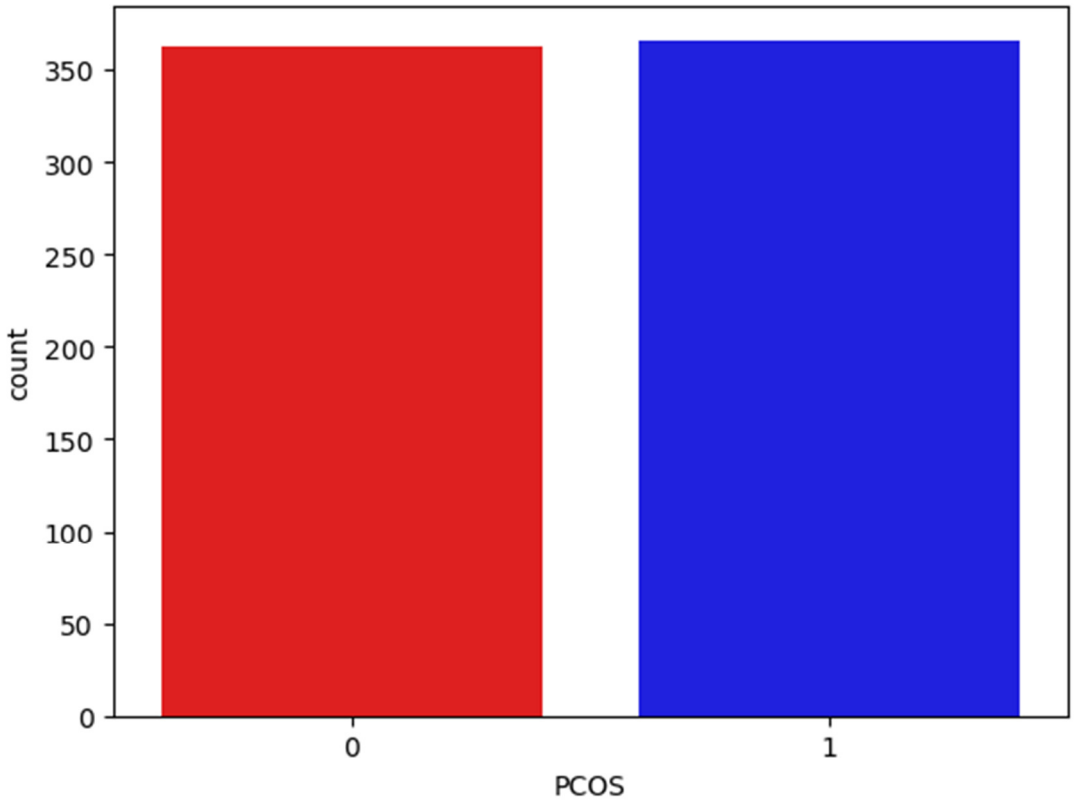


Figure 3. Target column count plot after ADASYN.

4. Experimental Results

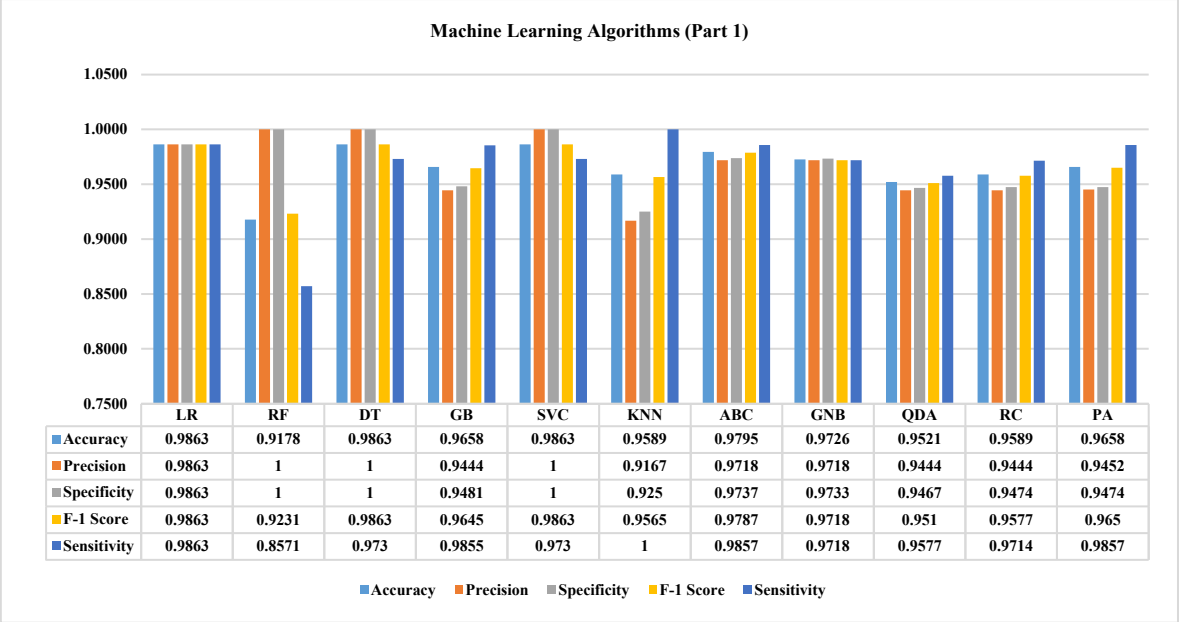
In this phase of the study, the evaluation of existing models played a pivotal role in assessing the efficiency of the proposed model targeting PCOS using the designated dataset [25]. The process commenced with the initial implementation of the chosen dataset, followed by a rigorous examination to identify and rectify missing or erroneous data points, ensuring the dataset's integrity. A diverse range of

machine learning algorithms was subsequently deployed, and their performances meticulously analyzed. For the proposed algorithms, a comprehensive assessment was conducted through confusion matrices, which included key metrics such as Accuracy, Precision, Recall, and F-1 Score, Specificity providing a holistic view of their predictive capabilities. Additionally, traditional algorithms underwent the same scrutiny, further enabling a comparative analysis. The evaluation extended to exploring the potential of different ensemble techniques, incorporating bagging, boosting, stacking, and voting, to leverage the collective strengths of multiple models for enhanced prediction accuracy. A total of five deep learning and eleven distinct traditional classifiers were harnessed, and the resulting outcomes, thoroughly assessed, facilitated the identification of the most effective approaches for predicting PCOS disease. This comprehensive evaluation process served as a critical step in gauging the performance of the proposed model and fine-tuning its predictive accuracy for practical application. Here FPR means False Positive Rate, FNR means False Negative Rate, NPV means Negative Predective Value, FDR means False Discovery Rate and MCC means Mathews Correction Coefficient.

The results obtained from the evaluation of various machine learning algorithms for predicting the presence of a certain condition reveal valuable insights into their performance characteristics showed in Table 2. Firstly, it is evident that several algorithms, including Logistic Regression (LR), Decision Tree (DT), Support Vector Classifier (SVC), and Adaptive Boosting Classifier (ABC), demonstrate consistently high levels of accuracy, precision, specificity, and F1 score, all above 0.95. These models exhibit robustness in their ability to correctly classify both positive and negative instances, indicating their suitability for reliable prediction tasks. On the other hand, Random Forest (RF), Gradient Boosting (GB), K-Nearest Neighbors (KNN), Gaussian Naïve Bayes (GNB), Quadratic Discriminant Analysis (QDA), Ridge Classifier (RC), and Passive Aggressive (PA) classifiers also perform reasonably well, albeit with slight variations in their performance metrics. For instance, RF achieves perfect precision and specificity, suggesting its capability to avoid false positives, but its sensitivity and F1 score are relatively lower compared to LR, DT, and SVC. Similarly, GB exhibits high sensitivity but slightly lower precision compared to LR and DT. KNN, while achieving high sensitivity, also exhibits a higher false positive rate (FPR), indicating potential for improvement in its classification boundaries. Furthermore, it is noteworthy that some algorithms, such as LR, DT, SVC, and ABC, achieve perfect precision, implying their ability to avoid false positives entirely. This characteristic is particularly desirable in medical diagnostics, where minimizing false positives is crucial to prevent unnecessary interventions or treatments. However, it is essential to consider the trade-offs between sensitivity and specificity when interpreting these results, as a balance between the two is often necessary to optimize overall model performance showed in Figures 4 and 5. Overall, the analysis highlights the diverse performance profiles of different machine learning algorithms and underscores the importance of selecting an appropriate model based on the specific requirements and constraints of the predictive task at hand. Additionally, it emphasizes the need for further investigation into the underlying factors influencing algorithm performance and the potential for ensemble techniques or model stacking to harness the strengths of multiple classifiers and improve overall predictive accuracy. The AUC-ROC curve is showed in Figure 6.

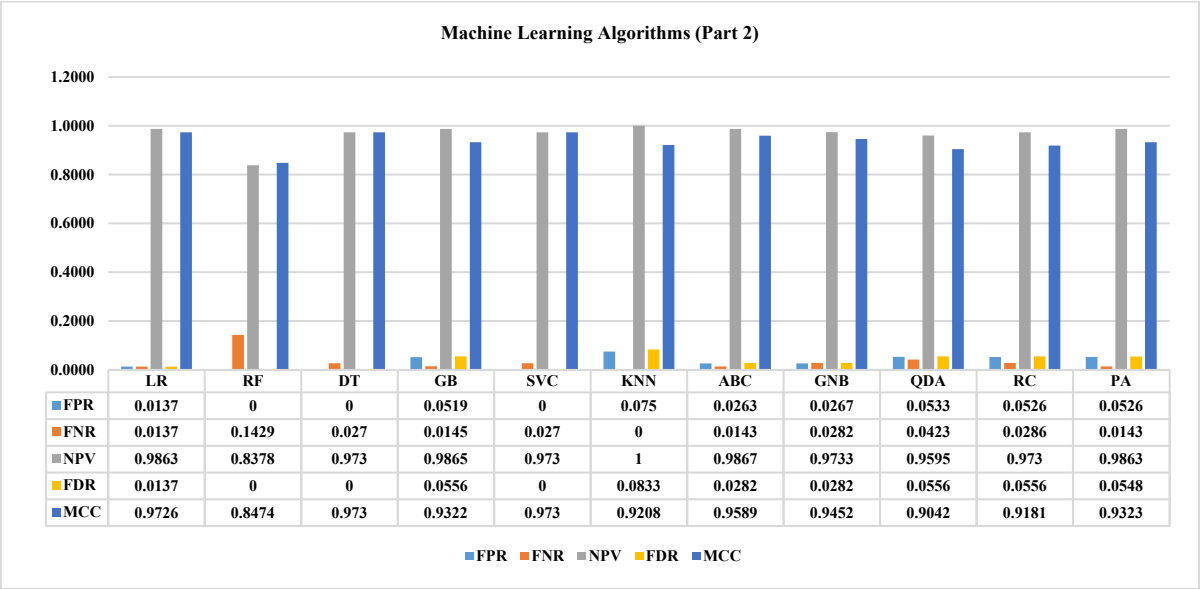
Table 2. Comparative analysis of machine learning algorithms.

Algorithms	Accuracy	Precision	Specificity	F-1 Score	Sensitivity	FPR	FNR	NPV	FDR	MCC
LR	0.9863	0.9863	0.9863	0.9863	0.9863	0.0137	0.0137	0.9863	0.0137	0.9726
RF	0.9178	1	1	0.9231	0.8571	0	0.1429	0.8378	0	0.8474
DT	0.9863	1	1	0.9863	0.973	0	0.027	0.973	0	0.973
GB	0.9658	0.9444	0.9481	0.9645	0.9855	0.0519	0.0145	0.9865	0.0556	0.9322
SVC	0.9863	1	1	0.9863	0.973	0	0.027	0.973	0	0.973
KNN	0.9589	0.9167	0.925	0.9565	1	0.075	0	1	0.0833	0.9208
ABC	0.9795	0.9718	0.9737	0.9787	0.9857	0.0263	0.0143	0.9867	0.0282	0.9589
GNB	0.9726	0.9718	0.9733	0.9718	0.9718	0.0267	0.0282	0.9733	0.0282	0.9452
QDA	0.9521	0.9444	0.9467	0.951	0.9577	0.0533	0.0423	0.9595	0.0556	0.9042
RC	0.9589	0.9444	0.9474	0.9577	0.9714	0.0526	0.0286	0.973	0.0556	0.9181
PA	0.9658	0.9452	0.9474	0.965	0.9857	0.0526	0.0143	0.9863	0.0548	0.9323



	LR	RF	DT	GB	SVC	KNN	ABC	GNB	QDA	RC	PA
Accuracy	0.9863	0.9178	0.9863	0.9658	0.9863	0.9589	0.9795	0.9726	0.9521	0.9589	0.9658
Precision	0.9863	1	1	0.9444	1	0.9167	0.9718	0.9718	0.9444	0.9444	0.9452
Specificity	0.9863	1	1	0.9481	1	0.925	0.9737	0.9733	0.9467	0.9474	0.9474
F-1 Score	0.9863	0.9231	0.9863	0.9645	0.9863	0.9565	0.9787	0.9718	0.951	0.9577	0.965
Sensitivity	0.9863	0.8571	0.973	0.9855	0.973	1	0.9857	0.9718	0.9577	0.9714	0.9857

Figure 4. Comparative analysis of Machine Learning algorithms (part 1).



	LR	RF	DT	GB	SVC	KNN	ABC	GNB	QDA	RC	PA
FPR	0.0137	0	0	0.0519	0	0.075	0.0263	0.0267	0.0533	0.0526	0.0526
FNR	0.0137	0.1429	0.027	0.0145	0.027	0	0.0143	0.0282	0.0423	0.0286	0.0143
NPV	0.9863	0.8378	0.973	0.9865	0.973	1	0.9867	0.9733	0.9595	0.973	0.9863
FDR	0.0137	0	0	0.0556	0	0.0833	0.0282	0.0282	0.0556	0.0556	0.0548
MCC	0.9726	0.8474	0.973	0.9322	0.973	0.9208	0.9589	0.9452	0.9042	0.9181	0.9323

Figure 5. Comparative analysis of Machine Learning algorithms (part 2).

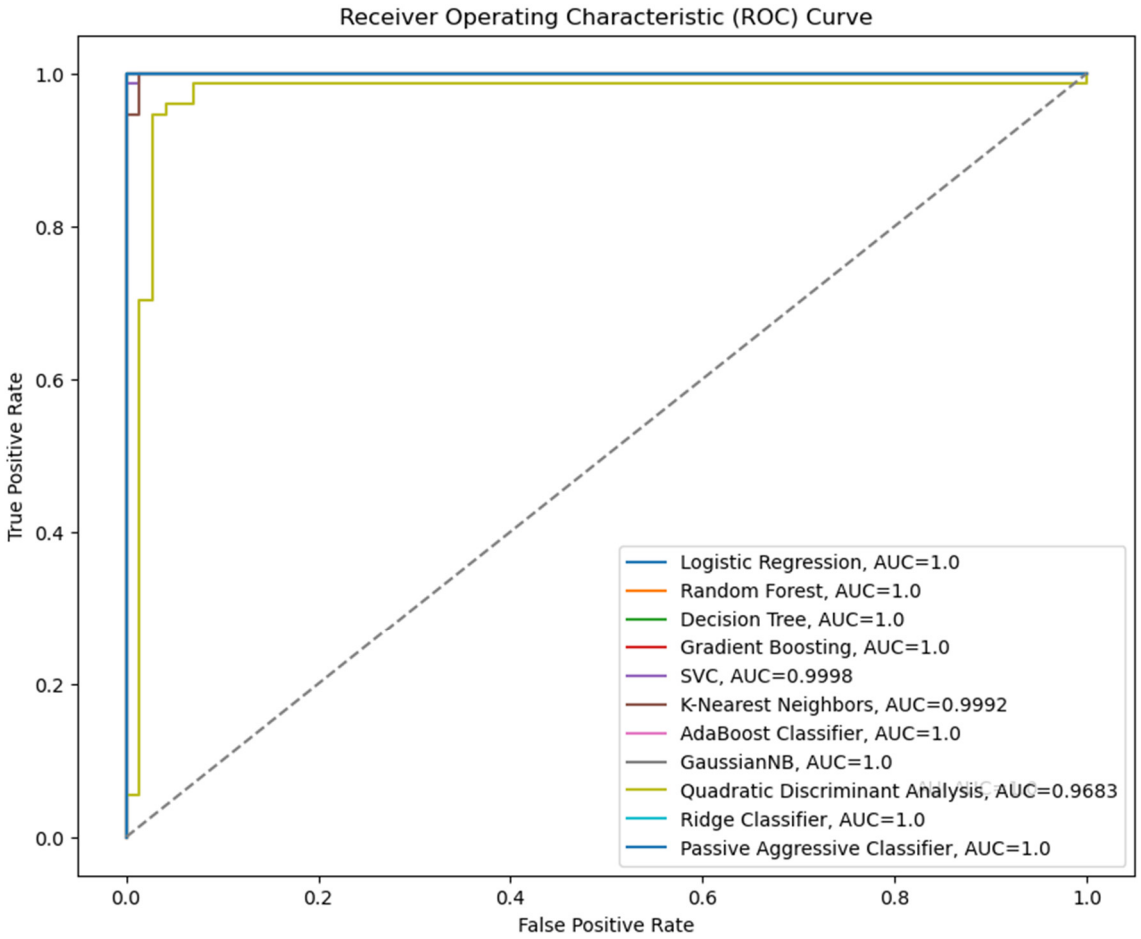


Figure 6. AUC-ROC Curve Comparative analysis of Machine Leaning algorithms.

The bagging ensemble classifiers, including Logistic Regression (LR), Random Forest (RF), Decision Tree (DT), Gradient Boosting (GB), Support Vector Machine (SVM), K-Nearest Neighbors (KN), Adaboost Classifier (ABC), Gaussian Naïve Bayes (GNB), Quadratic Discriminant Analysis (QDA), Ridge Classifier (RC), and Passive Aggressive Classifier (PA), were evaluated on various metrics.

The results obtained from the evaluation of bagging classifiers provide valuable insights into their performance characteristics across various metrics showed in Table 3. Firstly, it is evident that bagging classifiers, such as Logistic Regression (LR), Decision Tree (DT), Support Vector Classifier (SVC), Adaptive Boosting Classifier (ABC), Gaussian Naïve Bayes (GNB), and Quadratic Discriminant Analysis (QDA), demonstrate robust performance with accuracy scores ranging from approximately 95% to 97.26%. This indicates their ability to correctly classify instances across the dataset. Looking at precision, which measures the proportion of true positive predictions among all positive predictions made by the classifier, it is observed that most bagging classifiers achieve high precision scores, exceeding 90%. Notably, DT, ABC, and GNB achieve precision scores above 98%, indicating their ability to minimize false positive predictions effectively. Specificity, which measures the proportion of true negative predictions among all negative instances, is consistently high across all classifiers, with scores ranging from approximately 90% to 100%. This indicates that bagging classifiers are capable of accurately identifying negative instances of the target condition. F1 score, a harmonic mean of precision and sensitivity, is also high for most classifiers, indicating a balance between precision and sensitivity in their predictions. However, it is essential to consider the trade-offs between these metrics, as optimizing one may adversely affect the other. Sensitivity, or recall, measures the proportion of true positive predictions among all actual positive instances showed in Figures 7 and 8. While most classifiers achieve sensitivity scores above 90%, some, such as SVC and GNB, exhibit slightly lower sensitivity scores compared to others. Overall, the analysis demonstrates that bagging classifiers, including LR, DT, SVC, ABC, GNB, and QDA, exhibit strong performance across multiple evaluation metrics. However, it is essential to consider the specific requirements and constraints of the prediction task when selecting the most suitable classifier. Additionally, further optimization and fine-tuning of parameters may help enhance the performance of these classifiers and improve their predictive accuracy. The AUC-ROC curve

is showed in Figure 9.

Table 3. Comparative analysis of bagging ensemble algorithms.

Algorithms	Accuracy	Precision	Specificity	F-1 Score	Sensitivity	FPR	FNR	NPV	FDR	MCC
LR	0.9726	0.973	0.9722	0.973	0.973	0.0278	0.027	0.9722	0.027	0.9452
RF	0.911	0.9067	0.9028	0.9128	0.9189	0.0972	0.0811	0.9155	0.0933	0.8219
DT	0.9658	0.9859	0.9861	0.9655	0.9459	0.0139	0.0541	0.9467	0.0141	0.9323
GB	0.9589	0.9722	0.9722	0.9589	0.9459	0.0278	0.0541	0.0541	0.0278	0.9182
SVC	0.9726	1	1	0.973	0.9474	0	0.0526	0.9459	0	0.9467
KNN	0.9521	0.9583	0.9589	0.9517	0.9452	0.0411	0.0548	0.9459	0.0417	0.9042
ABC	0.9658	0.9861	0.9859	0.966	0.9467	0.0141	0.0533	0.9459	0.0139	0.9323
GNB	0.9658	0.9595	0.9589	0.966	0.9726	0.0411	0.0274	0.9722	0.0405	0.9316
QDA	0.9521	0.9444	0.9467	0.951	0.9577	0.0533	0.0423	0.9595	0.0556	0.9042
RC	0.9452	0.9452	0.9452	0.9452	0.9452	0.0548	0.0548	0.9452	0.0548	0.8904
PA	0.9521	0.9459	0.9589	0.9524	0.9589	0.0548	0.0411	0.9583	0.0541	0.9042

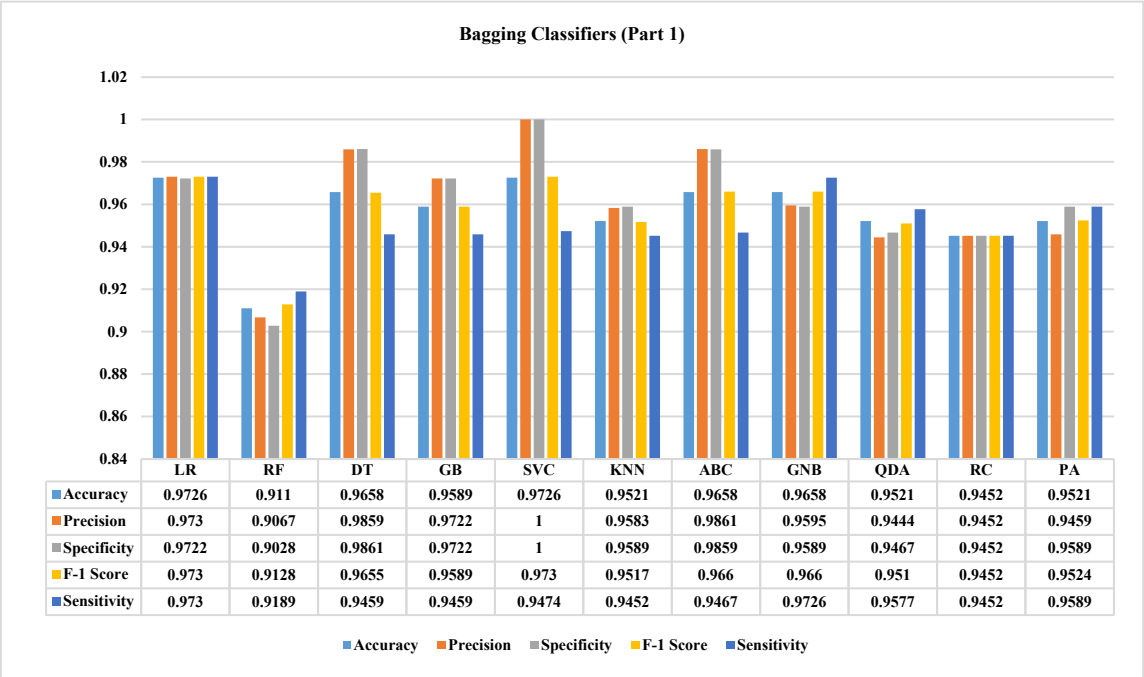


Figure 7. Comparative analysis of Bagging Classifiers (part 1).

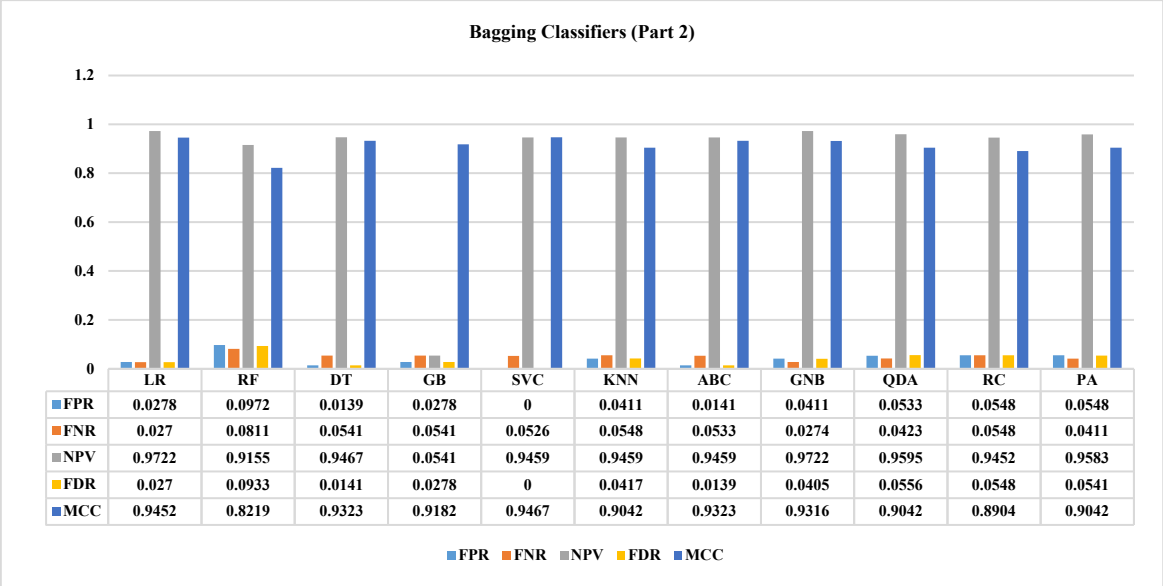


Figure 8. Comparative analysis of Bagging Classifiers (part 2).

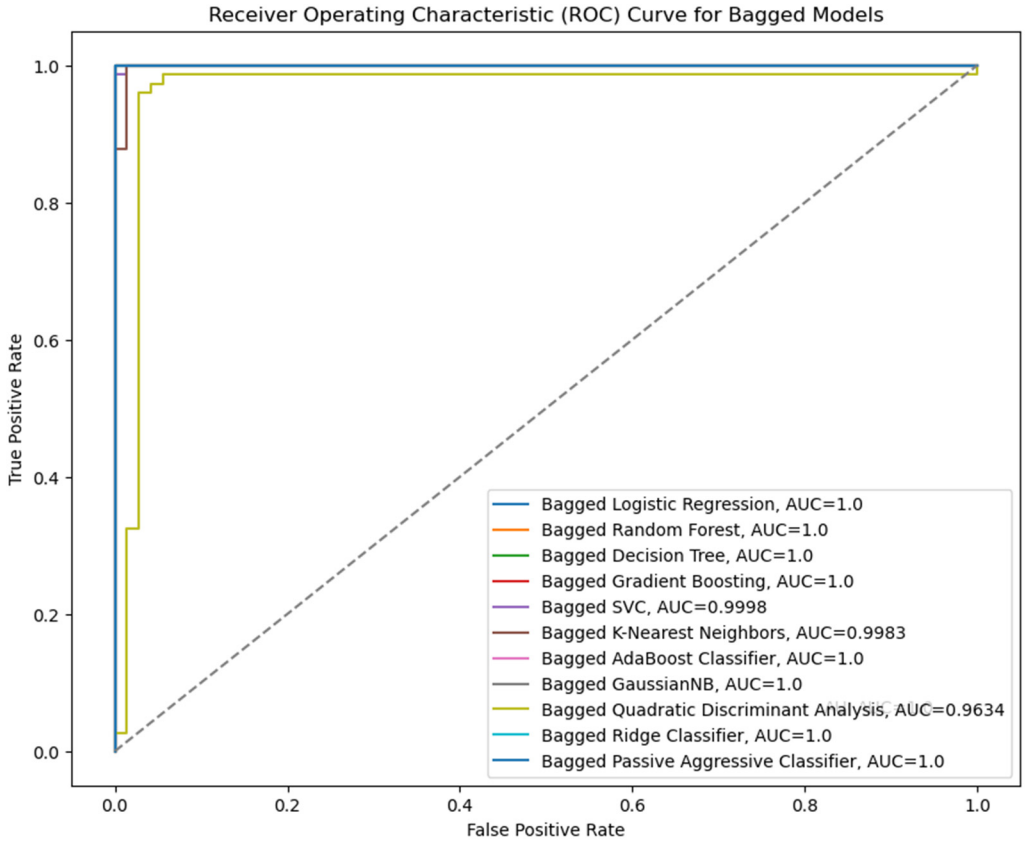


Figure 9. AUC-ROC Curve Comparative analysis of Bagging Classifiers.

The results obtained from the evaluation of boosting classifiers reveal their performance across various evaluation metrics showed in Table 4. Firstly, it is observed that boosting classifiers, including Logistic Regression (LR), Random Forest (RF), Decision Tree (DT), Gradient Boosting (GB), Support Vector Classifier (SVC), and Adaptive Boosting Classifier (ABC), demonstrate high accuracy scores, ranging from approximately 96.58% to 99.32%. This indicates their ability to correctly classify instances across the dataset with a high level of accuracy. Looking at precision, which measures the proportion of true positive predictions among all positive predictions made by the classifier, it is noted that most boosting classifiers achieve precision scores exceeding 94.44%. Notably, RF and DT achieve perfect precision scores of 100%, indicating their ability to minimize false positive predictions effectively. Specificity, which measures the proportion of true negative predictions among all negative instances, is consistently high across all classifiers, with scores ranging from approximately 94.81% to 100%. This indicates that boosting classifiers are capable of accurately identifying negative instances of the target condition. F1 score, a harmonic mean of precision and sensitivity, is also high for most classifiers, indicating a balance between precision and sensitivity in their predictions. However, it is essential to consider the trade-offs between these metrics, as optimizing one may adversely affect the other. Sensitivity, or recall, measures the proportion of true positive predictions among all actual positive instances. Most boosting classifiers achieve sensitivity scores above 94.67%, indicating their ability to correctly identify positive instances of the target condition showed in Figures 10 and 11. Overall, the analysis demonstrates that boosting classifiers, including LR, RF, DT, GB, SVC, and ABC, exhibit strong performance across multiple evaluation metrics. However, it is essential to consider the specific requirements and constraints of the prediction task when selecting the most suitable classifier. Additionally, further optimization and fine-tuning of parameters may help enhance the performance of these classifiers and improve their predictive accuracy. The AUC-ROC curve is showed in Figure 12.

Table 4. Comparative analysis of Boosting ensemble algorithms.

Algorithms	Accuracy	Precision	Specificity	F-1 Score	Sensitivity	FPR	FNR	NPV	FDR	MCC
LR	0.9726	0.9583	0.9605	0.9718	0.9857	0.0395	0.0143	0.9865	0.0417	0.9455
RF	0.9932	1	1	0.9931	0.9863	0	0.0137	0.9865	0	0.9864
DT	0.9863	1	1	0.9863	0.973	0	0.027	0.973	0	0.973
GB	0.9658	0.9444	0.9481	0.9645	0.9855	0.0519	0.0145	0.9865	0.0556	0.9322
SVC	0.9932	0.9861	0.9867	0.993	1		0	1	0.0139	0.9864
ABC	0.9658	0.9861	0.9859	0.966	0.9467	0.0141	0.0533	0.9459	0.0139	0.9323

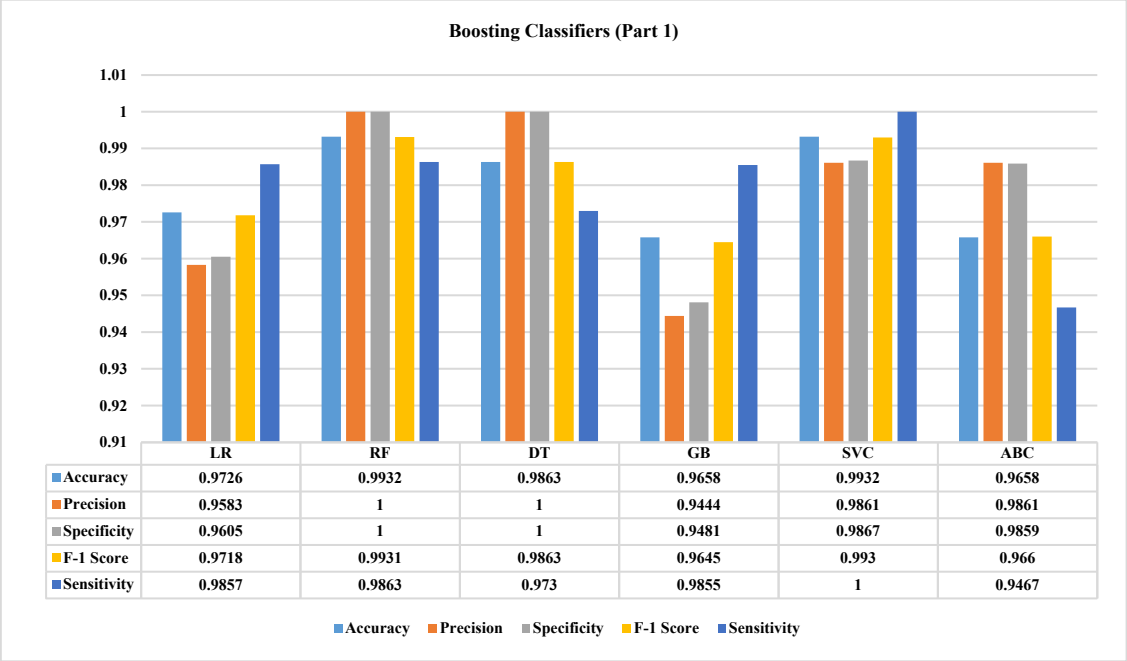


Figure 10. Comparative analysis of Boosting Classifiers (part 1).

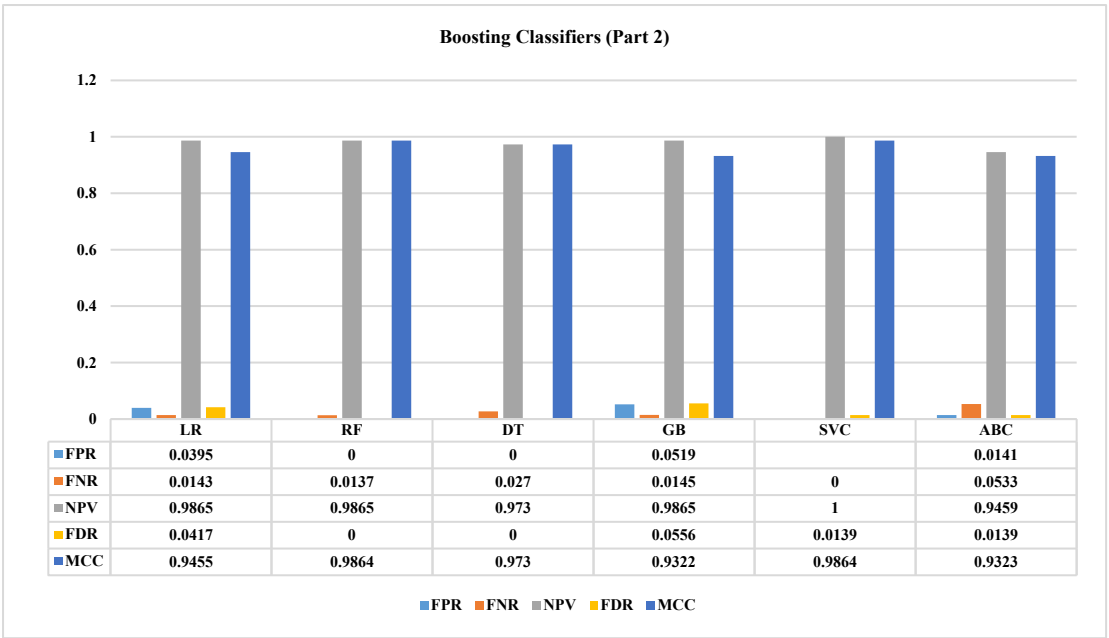


Figure 11. Comparative analysis of Boosting Classifiers (part 2).

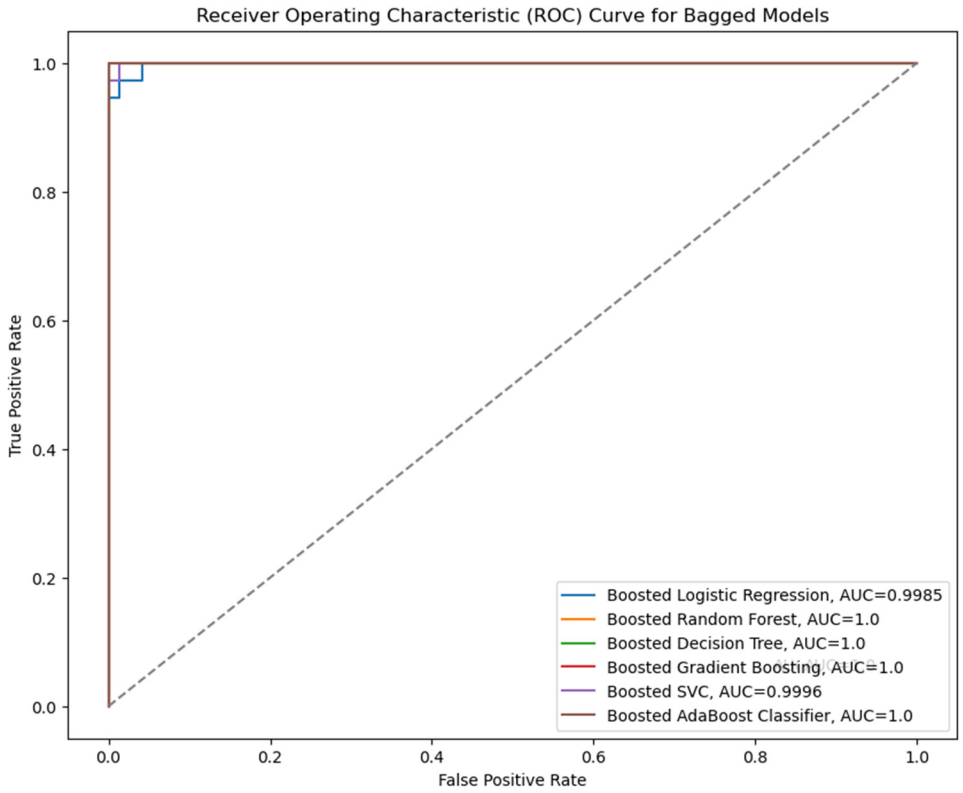
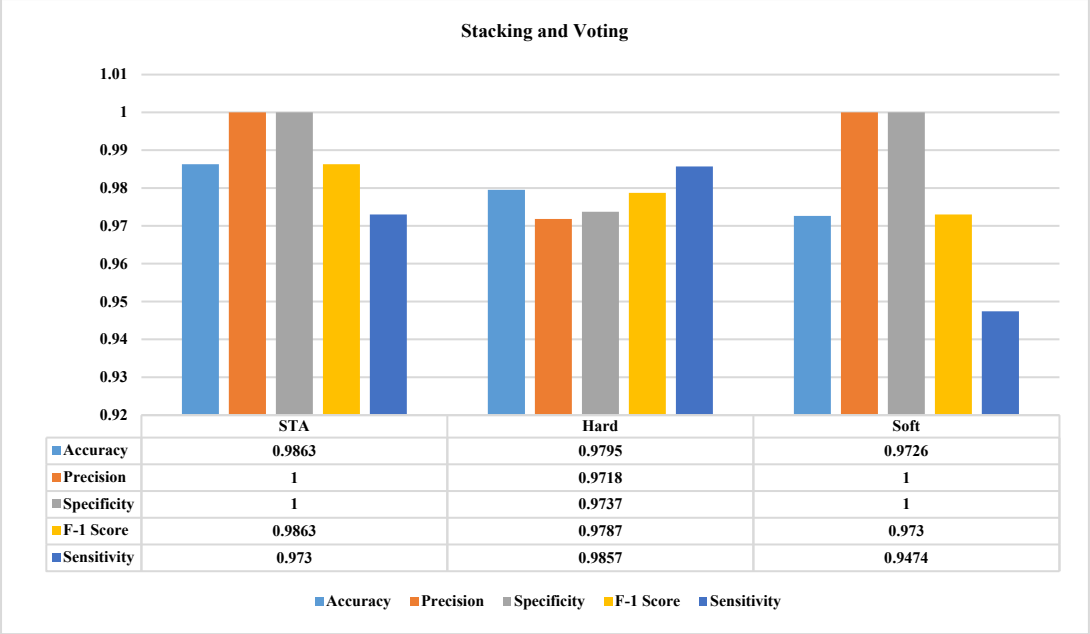


Figure 12. AUC-ROC Curve Comparative analysis of Boosting Classifiers.

The results from the stacking ensemble method, as well as the hard and soft voting classifiers, provide insights into their performance across various evaluation metrics showed in Table 5. Firstly, the stacking ensemble method (STA) achieves high accuracy, precision, and specificity scores, all reaching or exceeding 97.26%. This indicates the effectiveness of the stacking approach in combining the predictions of multiple base classifiers to produce accurate results. The hard voting classifier achieves a slightly lower accuracy of 97.95% compared to the stacking method, but it still demonstrates strong performance across all metrics. It achieves precision, sensitivity, and specificity scores above 97.18%, indicating its ability to make accurate predictions across both positive and negative instances. Similarly, the soft voting classifier also performs well, with an accuracy score of 97.26% and perfect precision and specificity scores. This suggests that the soft voting classifier effectively combines the probabilities predicted by the base classifiers to make accurate classifications. In terms of sensitivity, the stacking ensemble method and soft voting classifier achieve scores of 97.30% and 94.74%, respectively, indicating their ability to correctly identify positive instances. The hard voting classifier achieves a sensitivity score of 98.57%, demonstrating its effectiveness in accurately identifying positive instances. Overall, the results suggest that both the stacking ensemble method and the voting classifiers (both hard and soft) are effective in making accurate predictions for the given task showed in Figures 13 and 14. However, the choice between these methods may depend on factors such as computational complexity, interpretability, and the specific requirements of the application. Further analysis and experimentation may be necessary to determine the most suitable ensemble approach for the given problem domain. The AUC-ROC curve is showed in Figures 15 and 16.

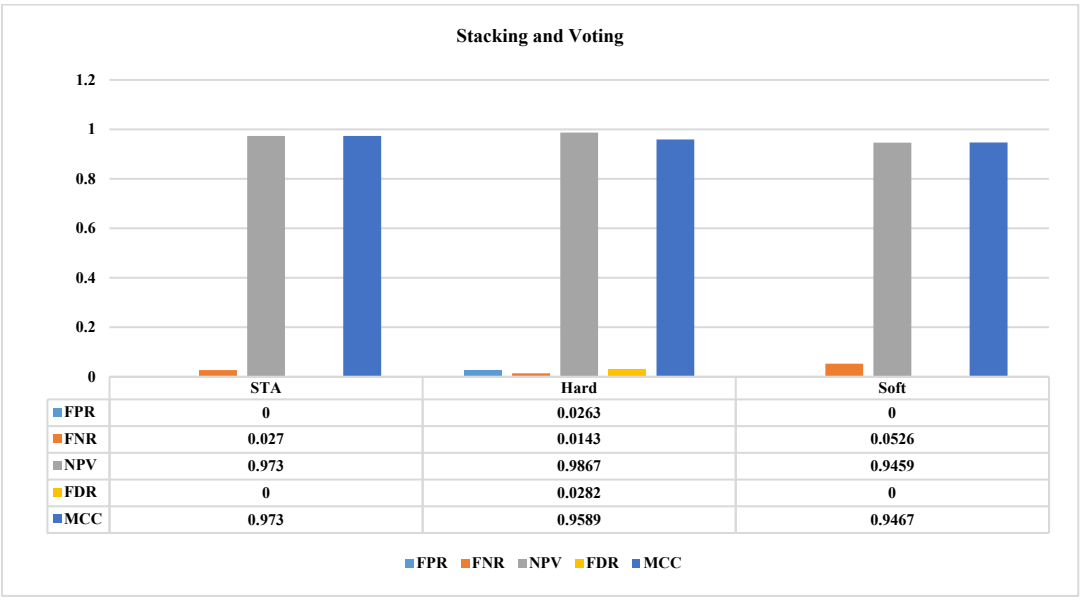
Table 5. Comparative analysis of Stacking and Voting ensemble algorithms.

Algorithms	Accuracy	Precision	Specificity	F-1 Score	Sensitivity	FPR	FNR	NPV	FDR	MCC
STA	0.9863	1	1	0.9863	0.973	0	0.027	0.973	0	0.973
Hard	0.9795	0.9718	0.9737	0.9787	0.9857	0.0263	0.0143	0.9867	0.0282	0.9589
Soft	0.9726	1	1	0.973	0.9474	0	0.0526	0.9459	0	0.9467



	STA	Hard	Soft
Accuracy	0.9863	0.9795	0.9726
Precision	1	0.9718	1
Specificity	1	0.9737	1
F-1 Score	0.9863	0.9787	0.973
Sensitivity	0.973	0.9857	0.9474

Figure 13. Comparative analysis of Stacking and Voting Classifiers (part 1).



	STA	Hard	Soft
FPR	0	0.0263	0
FNR	0.027	0.0143	0.0526
NPV	0.973	0.9867	0.9459
FDR	0	0.0282	0
MCC	0.973	0.9589	0.9467

Figure 14. Comparative analysis of Stacking and Voting Classifiers (part 2).

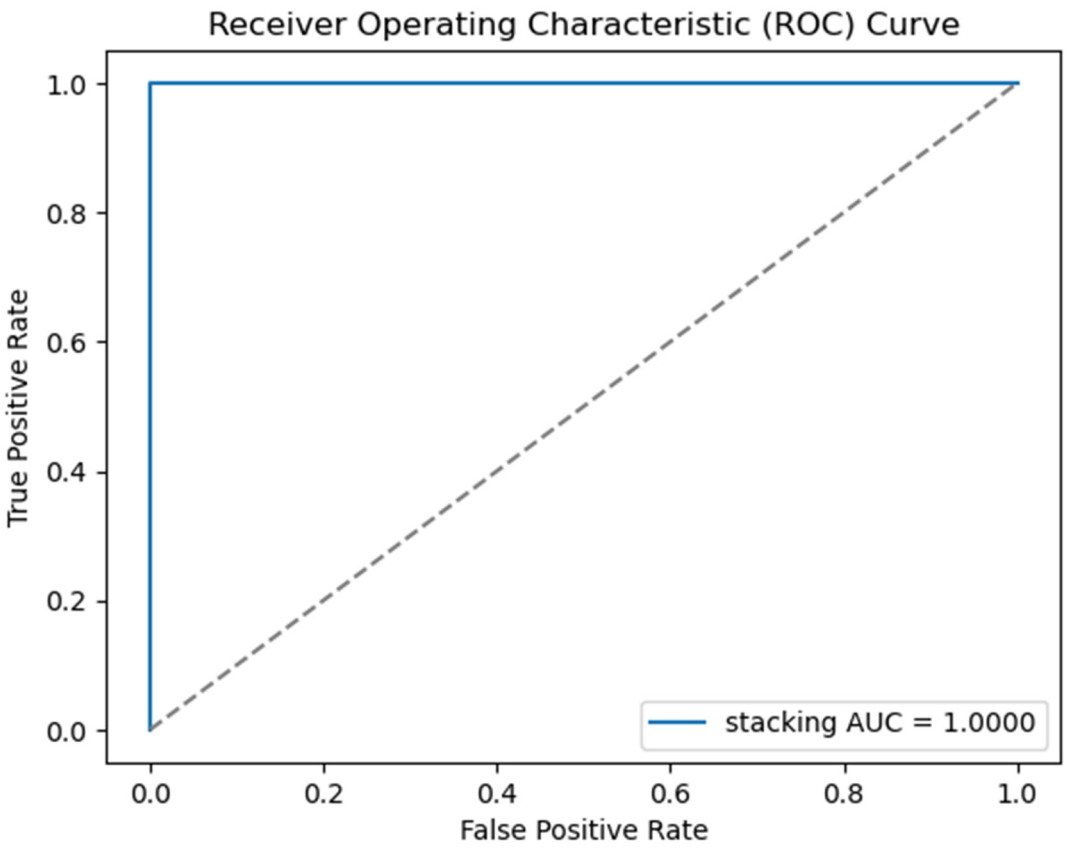


Figure 15. AUC-ROC Curve Comparative analysis of Stacking.

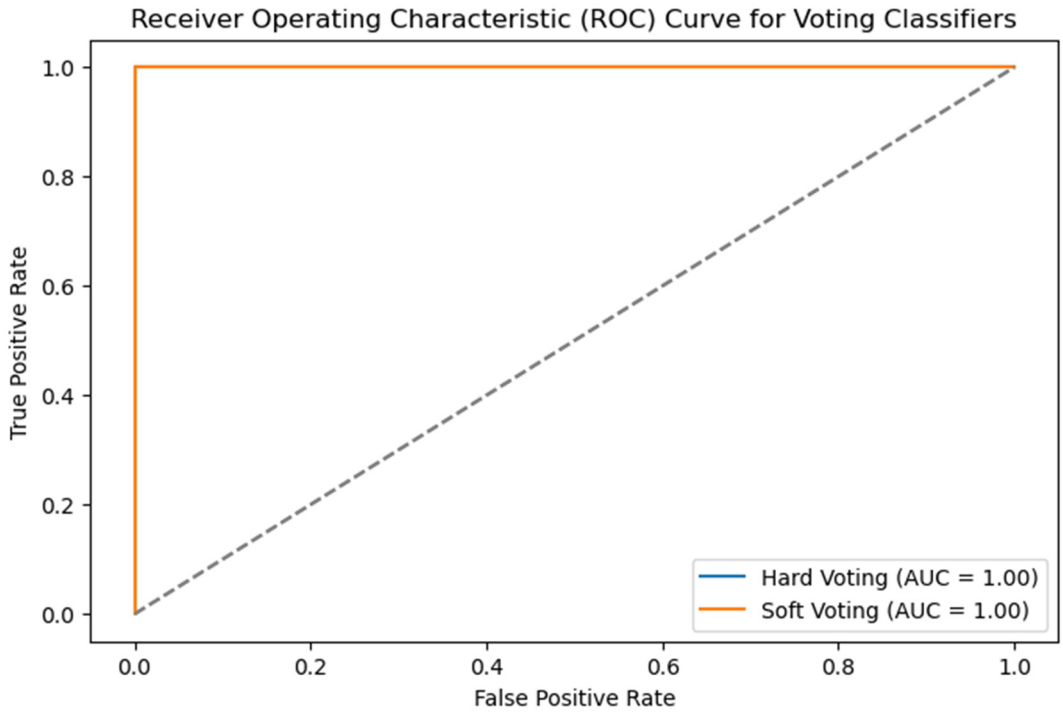


Figure 16. AUC-ROC Curve Comparative analysis of Hard and Soft Voting.

The results from the deep learning algorithms showcase varying degrees of performance across different evaluation metrics showed in Table 6. The Artificial Neural Network (ANN) demonstrates exceptionally high accuracy, precision, specificity, and F-1 score, all reaching or exceeding 98.63%. This

suggests that the ANN model effectively learns and generalizes patterns from the data, leading to accurate predictions. Similarly, the Recurrent Neural Network (RNN) achieves impressive performance, with high precision, sensitivity, and specificity scores, all above 97.18%. The RNN's ability to capture temporal dependencies in sequential data makes it well-suited for tasks involving time-series or sequential data, resulting in robust predictive performance. On the other hand, the Convolutional Neural Network (CNN) exhibits slightly lower performance compared to the ANN and RNN, with an accuracy of 89.04%. While the CNN performs well in terms of precision and sensitivity, its specificity is comparatively lower, indicating potential challenges in correctly identifying negative instances. The Long Short-Term Memory Network (LSTM) and Bi-Directional Long Short-Term Memory Network (BLSTM) also demonstrate strong performance, with accuracy scores of 94.52% and 91.10%, respectively showed in Figures 17 and 18. These models excel in capturing long-term dependencies and sequential patterns in the data, leading to effective predictions. Overall, the deep learning algorithms, particularly ANN and RNN, showcase remarkable performance in accurately predicting the target variable. Their ability to learn complex patterns and relationships within the data makes them valuable tools for various predictive modeling tasks. However, the choice of the appropriate deep learning architecture may depend on factors such as the nature of the data, computational resources, and specific requirements of the application. The AUC-ROC curve is showed in Figure 19.

Table 6. Comparative analysis of Deep Learning algorithms.

Algorithms	Accuracy	Precision	Specificity	F-1 Score	Sensitivity	FPR	FNR	NPV	FDR	MCC
ANN	0.9863	1	1	0.9863	0.973	0	0.027	0.973	0	0.973
RNN	0.9795	0.9718	0.9737	0.9787	0.9857	0.0263	0.0143	0.9867	0.0282	0.9589
CNN	0.8904	0.9028	0.9028	0.8904	0.8784	0.0972	0.1216	0.8784	0.0972	0.7812
LSTM	0.9452	0.9167	0.9231	0.9429	0.9706	0.0769	0.0294	0.973	0.0833	0.8917
BLSTM	0.911	0.8611	0.8765	0.9051	0.9538	0.1235	0.0462	0.9595	0.1389	0.8255

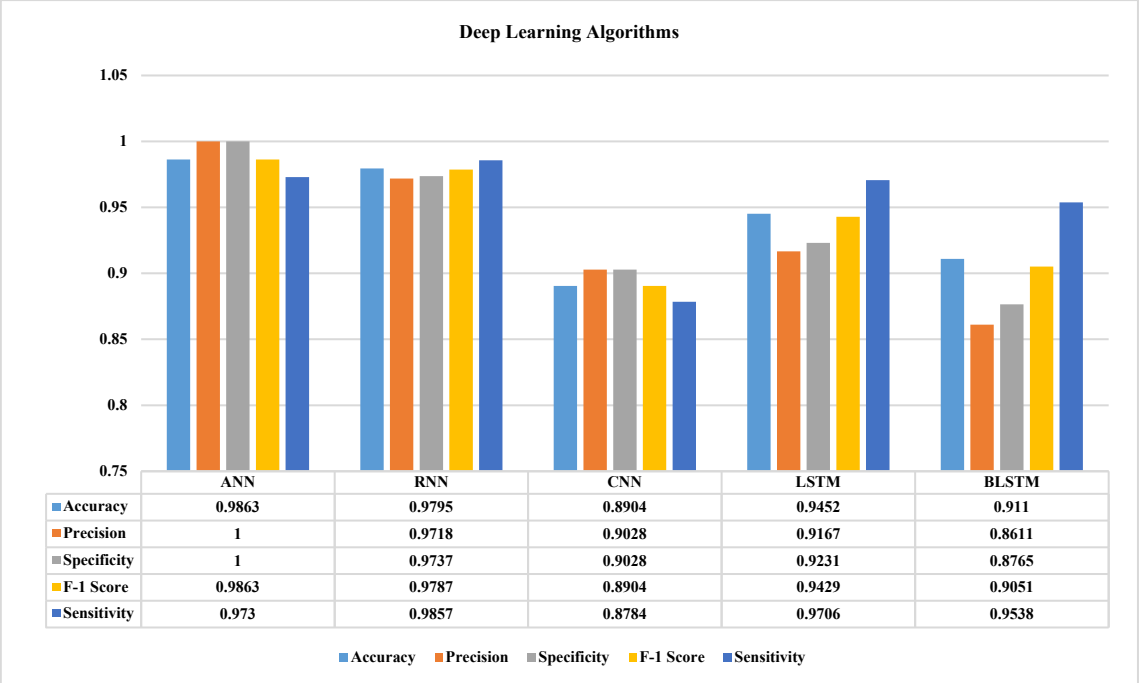
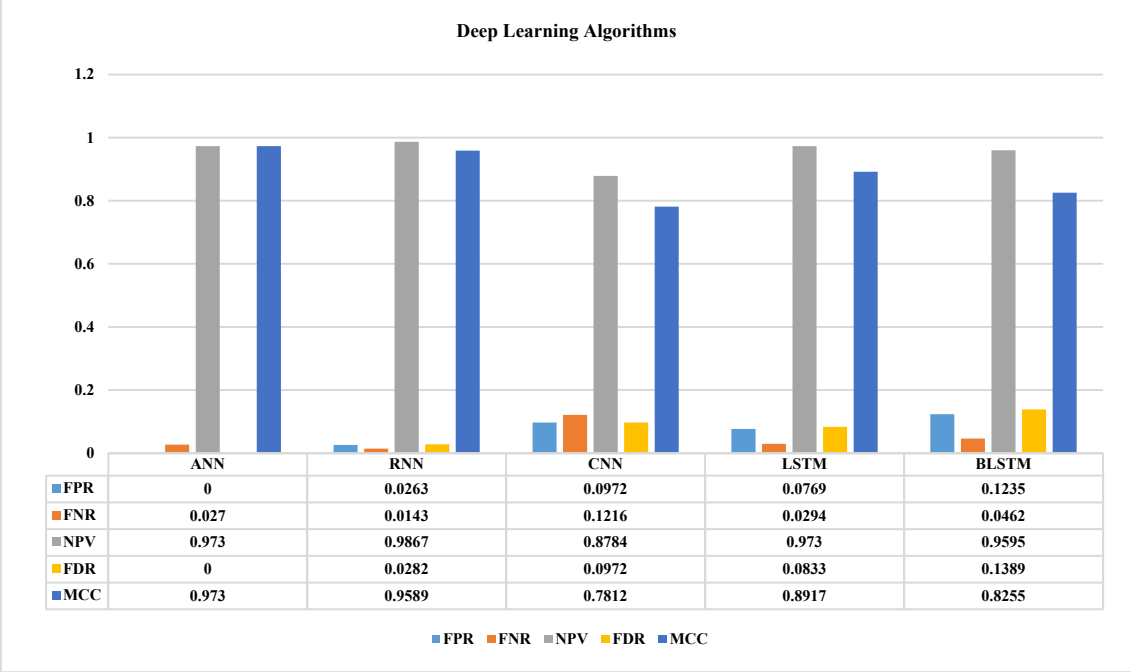


Figure 17. Comparative analysis of Deep Leaning Algorithms (part 1).



	ANN	RNN	CNN	LSTM	BLSTM
FPR	0	0.0263	0.0972	0.0769	0.1235
FNR	0.027	0.0143	0.1216	0.0294	0.0462
NPV	0.973	0.9867	0.8784	0.973	0.9595
FDR	0	0.0282	0.0972	0.0833	0.1389
MCC	0.973	0.9589	0.7812	0.8917	0.8255

Figure 18. Comparative analysis of Deep Leaning Algorithms (part 2).

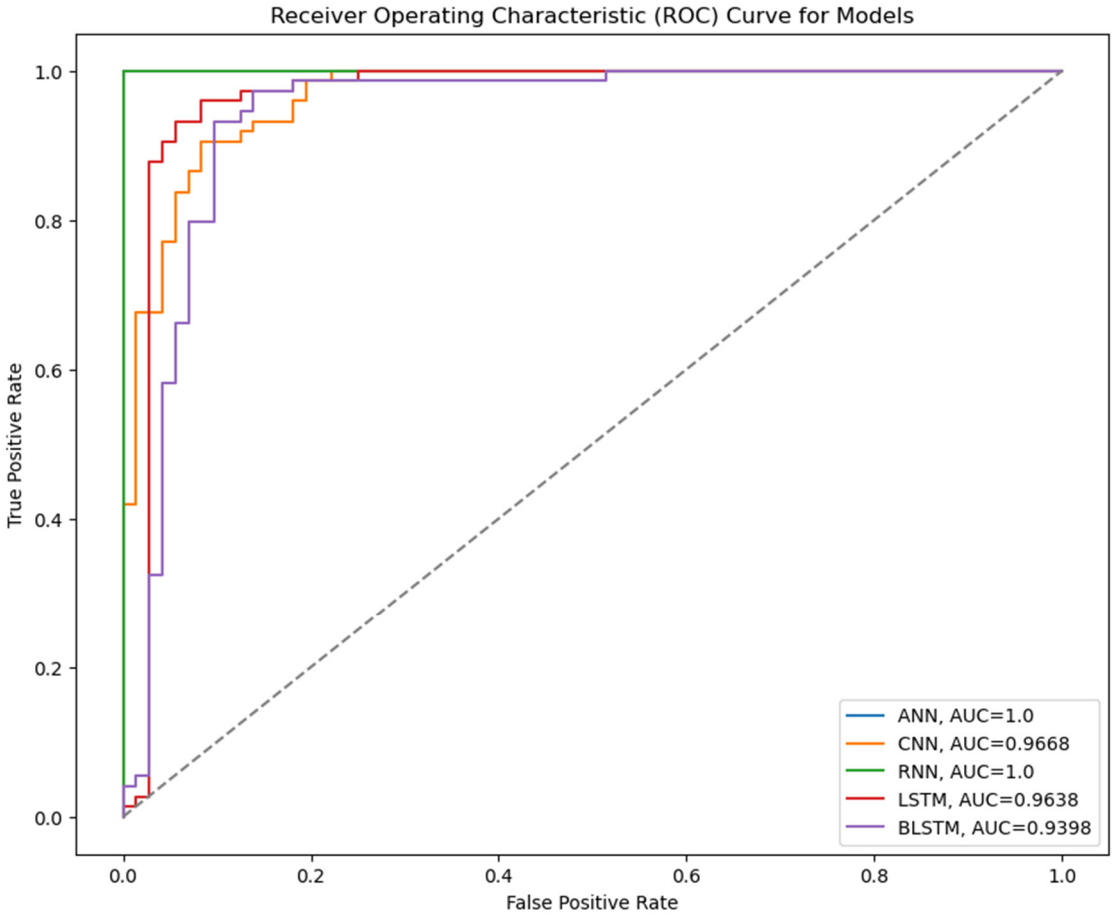
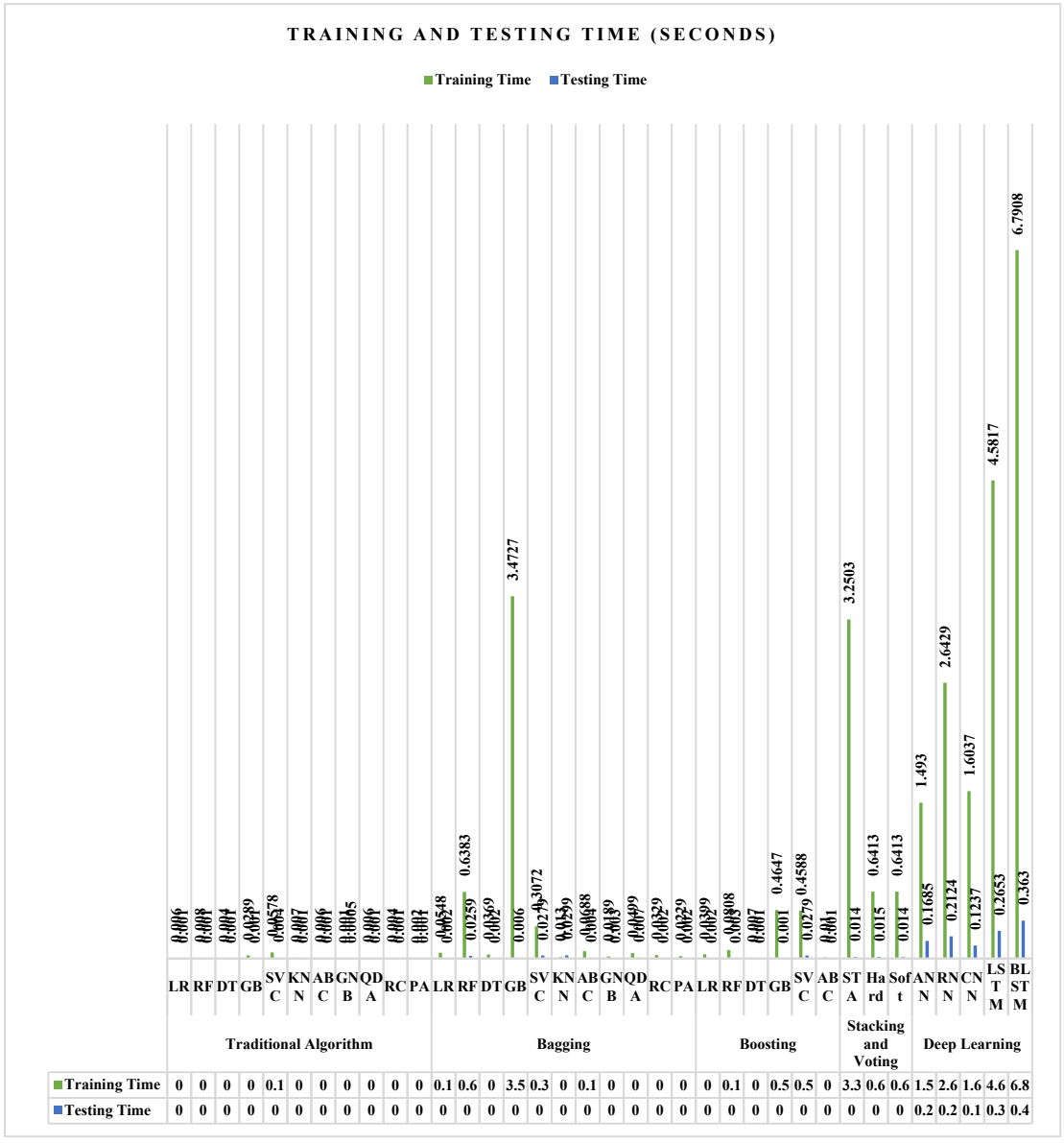


Figure 19. AUC-ROC Curve Comparative analysis of Deep Leaning Algorithms.

In this summary of training and testing times for various machine learning algorithms, it's evident that traditional algorithms such as Logistic Regression (LR), Random Forest (RF), and Decision Trees (DT) exhibit relatively short training and testing times, making them efficient choices for real-time

applications. Bagging and Boosting techniques, while offering improved performance, require longer training times, with Gradient Boosting (GB) being the most time-consuming among them. Stacking and Voting methods involve moderate training times, with Stacking showing the longest duration. Deep learning models, including Artificial Neural Networks (ANN), Recurrent Neural Networks (RNN), Convolutional Neural Networks (CNN), Long Short-Term Memory Networks (LSTM), and Bi-Directional LSTM (BLSTM), demonstrate longer training times due to their complex architectures and the need for extensive computation. However, they provide powerful capabilities for handling intricate data patterns, which can outweigh the increased training time in scenarios where accuracy and performance are paramount. The compilation time is showed in Figure 20.



	LR	RF	DT	GB	SVC	KNN	ABC	GNB	QDA	RC	PA	LR	RF	DT	GB	SVC	KNN	ABC	GNB	QDA	RC	PA	LR	RF	DT	GB	SVC	ABC	STA	Hard	Soft	ANN	RNN	CNN	LSTM	BLSTM
Training Time	0	0	0	0	0.1	0	0	0	0	0	0	0.1	0.6	0	3.5	0.3	0	0.1	0	0	0	0	0	0.1	0	0.5	0.5	0	3.3	0.6	0.6	1.5	2.6	1.6	4.6	6.8
Testing Time	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.2	0.2	0.1	0.3	0.4

Figure 20. Comparative analysis of Training and Testing Time (seconds).

5. Conclusions and Future Work

In conclusion, the application of algorithmic models for the prediction and early detection of Polycystic Ovary Syndrome (PCOS) holds significant promise in the realm of healthcare. Through the utilization of machine learning techniques, particularly classification algorithms, we have demonstrated the potential to accurately identify and predict PCOS cases based on a variety of clinical and diagnostic parameters. Our findings suggest that these models can play a crucial role in improving diagnostic accuracy, enabling timely intervention, and ultimately enhancing patient outcomes. By leveraging large datasets and advanced analytics, we can empower healthcare providers with valuable tools for risk assessment, personalized treatment planning, and proactive management of PCOS.

Moving forward, there are several avenues for further research and development in the field of PCOS prediction and management. Firstly, efforts should be directed towards the integration of multi-modal data sources, including genetic, hormonal, and imaging data, to enhance the predictive capabilities of machine learning models. Additionally, the development of interpretable and explainable AI techniques will be essential for facilitating clinical decision-making and fostering trust among healthcare professionals. Furthermore, longitudinal studies are needed to assess the long-term effectiveness and scalability of predictive models in real-world clinical settings. Collaborations between researchers, clinicians, and industry stakeholders will be instrumental in driving innovation and translating research findings into actionable insights that benefit patients affected by PCOS. Overall, the continued advancement of machine learning approaches holds the potential to revolutionize the diagnosis, treatment, and management of PCOS, ultimately improving the quality of life for individuals living with this complex condition.

Author Contributions

In this study the contribution of all authors was significant. Each of us were helpful to each other. We have collected the data and checked disease with Nutrition student Sonia Afrose. The data were checked and ensured analyzable by author Nowreen Ahsan. The methodology and evaluation was conducted by Aunik Hasan Mridul. The result analysis and evaluation was helped by Sayeda Sadia Alam and Md. Tanvir Mahmud kafi. The study was rechecked and reviewed by all of our authors.

Funding

The fund was collected by the authors.

Conflict of Interest Statement

There were no conflicts with another author works and no objection.

Data Availability Statement

The data was collected from kaggle [27]. The data was publicely available in online and ready to use.

References

1. D. Hu, W. Dong, X. Lu, H. Duan, K. He and Z. Huang, "Evidential mace prediction of acute coronary syndrome using electronic health records", *BMC Med. Informat. Decis. Making*, vol. 19, no. 2, pp. 9-17, 2019.
2. M. M. Hassan and T. Mirza, "Comparative analysis of machine learning algorithms in diagnosis of polycystic ovarian syndrome", *Int. J. Comput. Appl.*, vol. 175, pp. 42-53, Sep. 2020.
3. G. Du, L. Ma, J.-S. Hu, J. Zhang, Y. Xiang, D. Shao, et al., "Prediction of 30-day readmission: An improved gradient boosting decision tree approach", *J. Med. Imag. Health Informat.*, vol. 9, no. 3, pp. 620-627, 2019.
4. S. Bharati, P. Podder and M. R. H. Mondal, "Diagnosis of polycystic ovary syndrome using machine learning algorithms", *Proc. IEEE Region Symp. (TENSYMP)*, pp. 1486-1489, Jun. 2020.
5. Bhat, S. A. (2021). "Detection of Polycystic Ovary Syndrome Using Machine Learning Algorithms". Doctoral dissertation, Dublin, National College of Ireland.
6. S. Yang, X. Zhu, L. Zhang, L. Wang and X. Wang, "Classification and prediction of Tibetan medical syndrome based on the improved bp neural network", *IEEE Access*, vol. 8, pp. 31114-31125, 2020.
7. D. Dewailly, M. E. Lujan, E. Carmina, M. I. Cedars, J. Laven, R. J. Norman, et al., "Definition and significance of polycystic ovarian morphology: A task force report from the androgen excess and polycystic ovary syndrome society", Hum. *Reproduction Update*, vol. 20, no. 3, pp. 334-352, 2014.
8. A. Saravanan and S. Sathiamoorthy, "Detection of polycystic ovarian syndrome: A literature survey", *Asian J. Eng. Appl. Technol.*, vol. 7, pp. 46-51, Nov. 2018.
9. X.-Z. Zhang, Y.-L. Pang, X. Wang and Y.-H. Li, "Computational characterization and identification of human polycystic ovary syndrome genes", *Sci. Rep.*, vol. 8, no. 1, Dec. 2018.
10. V. Thakre, "PCOcare: PCOS detection and prediction using machine learning algorithms", *Biosci. Biotechnol. Res. Commun.*, vol. 13, no. 14, pp. 240-244, Dec. 2020.
11. R. M. Aziz, "Nature-inspired metaheuristics model for gene selection and classification of biomedical microarray data", *Med. Biol. Eng. Comput.*, vol. 60, no. 6, pp. 1627-1646, 2022.
12. R. M. Aziz, "Application of nature inspired soft computing techniques for gene selection: A novel frame work for classification of cancer", *Soft Comput.*, vol. 1, pp. 1-18, Apr. 2022.
13. Z. Na, W. Guo, J. Song, D. Feng, Y. Fang and D. Li, "Identification of novel candidate biomarkers and immune infiltration in polycystic ovary syndrome", *J. Ovarian Res.,* vol. 15, no. 1, pp. 1-13, 2022.
14. S. Dhar, S. Mridha and P. Bhattacharjee, "Mutational landscape screening through comprehensive in silico analysis for polycystic ovarian syndrome–related genes", *Reproductive Sci.*, vol. 29, no. 2, pp. 480-496, 2022.
15. Shorove Tajmen, Asif Karim, Aunik Hasan Mridul, Sami Azam, Pronab Ghosh, Alamin Dhaly, Md Nour Hossain. "A Machine Learning based Proposition for Automated and Methodical Prediction of Liver Disease". In *Proceedings of the April 2022 The 10th International Conference on Computer and Communications Management in Japan,* 29–31 July 2022.
16. Aunik Hasan Mridul, Md. Jahidul Islam, Mushfiqur Rahman, Mohammad Jahangir Alam, Asifuzzaman Asif. "A Machine Learning-Based Traditional and Ensemble Technique for Predicting Breast Cancer", In *Proceedings of the December, 2022. Conference: 22th International Conference on Hybrid Intelligent Systems (HIS 2022) online, Auburn, WA, USA*, 13-15 December 2022.

17. “Logistic Regression for Machine Learning”. Available at: https://www.capitalone.com /tech/machine-learning/what-is-logistic-regression/ (accessed on 6 August 2021).
18. G. Pronab, A., S. Tanjila Atik, S. Afrin, and M. Saifuzzaman. “Expert cancer model using supervised algorithms with a LASSO selection approach”. *International Journal of Electrical and Computer Engineering (IJECE)* 11, no. 3 (2021): 2631.
19. L. Aurélie and C. Croux. “Bagging and boosting classification trees to predict churn” *Journal of Marketing Research* 43, no. 2 (2006): 276-286.
20. C. Bentéjac, A. Csörgő, G. Martínez-Muñoz. “A comparative analysis of gradient boosting algorithms”. *ArtificialIntelligence Review* 54, no. 3 (2021): 1937-1967.
21. W. Yizhen, S. Jha, and K. Chaudhuri. “Analyzing the robustness of nearest neighbors to adversarial examples”. In *Proceedings of the International Conference on Machine Learning*, pp. 51335142. PMLR, 2018
22. H. Drucker, C. Cortes, L.D. Jackel, Y. LeCun, V. Vapnik. “Boosting and other ensemble methods”. *Neural* Computation 6, no. 6 (1994): 1289-1301.
23. A. Sharma, A. Suryawanshi. “A novel method for detecting spam email using KNN classification with spearman correlation as distance measure”. *International Journal of Computer Applications* 136, no. 6 (2016): 28-35.
24. M. Pasha, M. Fatima. “Comparative Analysis of Meta Learning Algorithms for Liver Disease Detection”. J. Softw. 12, no.12 (2017): 923-933.
25. I. Rakibul; A. Reddy Beeravolu, Md Al Habib Islam, Asif Karim, Sami Azam, and Sanzida Akter Mukti. “A Performance Based Study on Deep Learning Algorithms in the Efficient Prediction of Heart Disease”. In *Proceedings of the 2021 2nd International Informatics and Software Engineering Conference (IISEC)*, pp. 1-6. IEEE, 2021.
26. A. Hasan Mridul; N. Ahsan; M. Abu Saleh; A. Sonia Afrose. “Effective Early Hypothyroid Disease Prediction Using Traditional and Ensemble Machine Learning Algorithms”, *SoCPaR 2023*, December 2023.
27. P. Kottarathil, “Polycystic ovary syndrome (PCOS) |Kaggle”, 2022.