Metaheuristic Algorithms based Ensemble Models for Enhanced Chronic Disease Diagnosis

Submitted in partial fulfillment of the requirements

for the award of the degree of

DOCTOR OF PHILOSOPHY

Submitted by

Srinivas Arukonda

(Roll No. 721089)

Under the supervision of

Dr. Ramalingaswamy Cheruku

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY WARANGAL
TELANGANA - 506004, INDIA
DECEMBER 2023

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY WARANGAL
TELANGANA - 506004, INDIA

THESIS APPROVAL FOR Ph.D.

This is to certify that the thesis entitled, Metaheuristic Algorithms based Ensemble Models for Enhanced Chronic Disease Diagnosis, submitted by Mr. Srinivas Arukonda [Roll No. 721089] is approved for the degree of DOCTOR OF PHILOSOPHY at the National Institute of Technology Warangal,Telanagana, India

Examiner

Research Supervisor
Prof. Ramalingaswamy Cheruku
Assistant Professor (Gr-I)
Dept. of Computer Science and Engg.
NIT Warangal
India

Chairman
Prof. R. Padmavathy
Professor & Head
Dept. of Computer Science and Engg.
NIT Warangal
India

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY WARANGAL
TELANGANA - 506004, INDIA

CERTIFICATE

This is to certify that the thesis entitled, Metaheuristic Algorithms based Ensemble Models for Enhanced Chronic Disease Diagnosis, submitted in partial fulfillment of the requirement for the award of the degree of DOCTOR OF PHILOSOPHY to the National Institute of Technology Warangal, is a bonafide research work done by Mr. Srinivas Arukonda [Roll No. 721089] under my supervision. The contents of the thesis have not been submitted elsewhere for the award of any degree.

Research Supervisor
Prof. Ramalingaswamy Cheruku (Gr-I)
Dept. of Computer Science and Engg.
NIT Warangal
India

Place: NIT Warangal

Date: 14th December, 2023

DECLARATION

This is to certify that the work presented in the thesis entitled "*Metaheuristic Algorithms based Ensemble Models for Enhanced Chronic Disease Diagnosis*" is a bonafide work done by me under the supervision of Prof. Ramalingaswamy Cheruku was not submitted elsewhere for the award of any degree.

I declare that this written submission represents my ideas in my own words and where other's ideas or words have been included, I have adequately cited and referenced the sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented fabricated or falsified any idea/date/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Srinivas Arukonda
(Roll No. 721089)
Date:14th Dec 2023

ACKNOWLEDGMENTS

I would like to express my sincere gratitude and appreciation to my supervisor, Prof. Ramalingaswamy Cheruku for his invaluable guidance throughout the completion of this work. Their continuous support, timely feedback, and constructive discussions have played a pivotal role in helping me achieve my objectives. I am grateful for the ample time they dedicated to reviewing my work and providing insightful suggestions for improvement. Their mentorship not only shaped me as a researcher but also as an individual. I have been inspired by their words, actions, and values, which have demonstrated the qualities of a great teacher and a compassionate human being. Their unwavering dedication and commitment to excellence have left a profound impact on me. I aspire to embody these remarkable qualities throughout my life.

I would like to express my heartfelt gratitude to all the members of my Doctoral Scrutiny Committee (DSC), namely Prof. R.Padmavathy, Prof. U.S.N.Raju, Prof. Sanjaya Kumar Panda, and Prof. Venkatesh Gudipadu. Their valuable comments and suggestions during the oral presentations have greatly enriched my research work. I am truly fortunate to have had the opportunity to attend lectures by esteemed professors such as Prof. P. Radha Krishna, Prof. Sanjeevi, Prof. U.S.N. Raju, and Prof. Kadambari. Their knowledge and expertise have been instrumental in broadening my understanding of the field.

I am immensely thankful to Prof. P. Radha Krishna, Prof. Ravichandra Sadam, and Prof. R. Padmavathy Heads of Dept. of CSE and chairman of DSC, during my tenure for providing adequate facilities in the department for carrying out the oral presentations. I wish to express my thanks to all the esteemed faculty members of the Department of Computer Science and Engineering at NIT Warangal. I would also like to extend my heartfelt gratitude to Prof. N.V. Ramana Rao and Prof. Bidyadhar Subudhi, the Director of

NIT Warangal, for their unwavering support and encouragement throughout my research endeavor. I am truly fortunate to have been part of such a remarkable institution and to have received support from all these individuals who have contributed to my academic and personal development.

I am immensely grateful to my wife Ramya and beloved daughter Sayonah, my mother and father-in-law's, for their unwavering love, support, and prayers throughout my journey. Their constant encouragement and support have been my pillars of strength, providing me with the confidence and motivation to overcome challenges and pursue excellence.

I would like to express my heartfelt gratitude to my seniors, Amilpur Santhosh, and Chandramohan as well as my co-scholars, Rampavan, Shailendra Kumar, Rajesh, Asif and Monika for their invaluable support and selfless cooperation throughout my Ph.D. journey. Their presence and assistance have been truly remarkable, and I am deeply thankful for their unwavering support and readiness to help me at any time. In particular, I would like to express a special thanks to Shailendra for their unconditional support, and constant presence in both the joyful and challenging moments of my personal life during Ph.D Journey. His friendship has been a source of strength and motivation, and I am forever grateful to have such amazing individuals by my side. Without the support and camaraderie of these remarkable individuals, my Ph.D. journey would have been much more challenging. I am truly blessed to have such incredible friends in my life, and I will cherish their friendship and support forever.

Srinivas Arukonda

Dedicated to

Family, Friends & Teachers

ABSTRACT

Disease diagnosis is a fundamental aspect of modern healthcare, where accurate and timely detection can profoundly impact patient outcomes. Leveraging the power of ensemble techniques has emerged as a promising avenue to enhance diagnostic accuracy. However, the intricate landscape of disease data presents challenges that necessitate innovative solutions. Ensemble methods, which combine the predictions of multiple models, offer a means to improve disease diagnosis by leveraging diverse perspectives. Yet, effectively harnessing these techniques remains challenging due to class imbalance within datasets and the intricate task of configuring optimal ensembles.

This thesis embarks on a comprehensive exploration, addressing these challenges through a meticulously designed sequence of objectives. In this thesis first, we introduced a three-level stacking approach that integrates the Adaptive Synthetic Sampling (ADASYN) technique to handle class imbalance, while Particle Swarm Optimization (PSO) fine-tunes Support Vector Machine (SVM) meta-model. The resulting ensemble exhibits exceptional performance across key metrics, including AUC, accuracy, specificity, and precision.

Building on this foundation, we delve into diversity-based ensemble frameworks. In our second objective, address the challenges of diversity based classifier selection. To achieve this proposed a novel fitness function that enhances the diversity of base learners within the ensemble. By combining this with bootstrapped bags and cross-validation, we demonstrate its superiority over existing ensemble models, reinforcing the potential of diversity-driven strategies.

Continuing our exploration, the third objective introduces the Bagging Approach with Teaching-Learning-Based Optimization (BA-TLBO). This dynamic ensemble optimization technique strikes a delicate balance between accuracy and diversity through dynamic weight updation and bag size adjustments. The approach's ability to maintain exploration while optimizing exploitation is validated through rigorous experimentation, positioning it as a robust alternative to traditional ensemble methods.

Our final objective takes on the complex challenge of classifier selection and placement within an ensemble framework. We navigate the intricate landscape of classifier configurations through a dynamic three-level ensemble framework guided by a nested Genetic Algorithm (GA) and an innovative fitness function. The approach's remarkable outcomes further underscore its potential for accurate disease diagnosis.

In summary, this thesis unveils a strategic sequence of ensemble techniques that effectively address challenges in disease diagnosis. By systematically advancing from class imbalance solutions to diversity-driven strategies and sophisticated ensemble optimization, it promises enhanced diagnostic accuracy and ultimately improved patient care.

Contents

List of Figures

List of Tables

Abbreviations

IQR	Inter Quartile Range
CNN	Convolutional Neural Network
SMOTE	Synthetic Minority Over Sampling
BSMOTE	Borderline Synthetic Minority Over Sampling
ADASYN	Adaptive Synthetic Minority Over Sampling
ROS	Random Over Sampling
KNN	K-Nearest Neighbors
DT	Decision Tree
LR	Logistic Regression
SVM	Support Vector Machine
MLP	Multilayer Perceptron (a type of neural network)
GB	Gradient Boosting
AB	Ada Boost
BC	Bagging Classifier (Bootstrap Aggregating)
CB	CatBoost (a gradient-boosting algorithm)
XGB	Extreme Gradient Boosting

MCS	Multiple Classifier System
CE	Classifier Ensemble
TLBO	Teaching-Learning-Based Optimization
GA	Genetic Algorithm
PSO	Particle Swarm Optimization
5FCV	Five-Fold Cross Validation
BA-TLBO	Bagging Approach with Teaching-Learning-Based Optimization
WSR	Wilcoxon Signed Rank
Acc	Accuracy
AUC	Area Under the ROC Curve
Sen	Sensitivity
Spe	Specificity
Pre	Precision
GM	G-Measure
TPR	True Positive Rate
FPR	False Positive Rate
SOTA	State-Of-The-Art
AW	Accuracy Weight
DW	Diversity Weight
Ham	Hamming Distance
Ent	Entropy
BC	Bhattacharya Distance

Qstat	Q statistics
DM	Diversity measure
Acc	Accuracy
Sen	Sensitivity
Spe	Specificity
Pre	Precision
OB	Optimal Bag

Chapter 1

Introduction

Disease diagnosis is a process by which a doctor determines whether a patient has a disease based on the patient's health condition and the type of disease the patient has. In an actual disease diagnosis environment, especially when there are a huge number of patients and the amount of data to be processed is too large, it may be troublesome for doctors to handle in a short period of time. In disease diagnosis, to improve predictive performance, we ensure that data should be preprocessed and processed with outliers, missing values, and data scaling. Various preprocessing techniques, such as the Inter Quartile Range (IQR), are used to assess the variability where most of your values lie. Most of the disease datasets are class-imbalanced, and classification results are biased towards the majority class. There is much attention to dealing with class-imbalanced data for effective disease diagnosis. In the literature, there are various oversampling techniques already used in disease diagnosis, such as the Synthetic Minority Over-Sampling Technique (SMOTE), Borderline Synthetic Minority Over-Sampling Technique (BSMOTE), Adaptive Synthetic Minority Over-Sampling Technique (ADASYN), and Random Over-Sampling Technique (ROS) [1]. SMOTE creates new artificial instances utilizing knowledge about the neighbors that surround each sample of the minority class [2]. Whereas in other approaches oversampled instances are arbitrarily chosen through duplication. To determine the k closest neighbors of a given minority data instance from the neighborhood, SMOTE uses the K-Nearest Neighbour (K-NN) technique. BSMOTE steps are similar to SMOTE to produce artificial data [3]. To solve the issue of minority instance misclassification (and to improve the detection rate of minority instances), it also reinforces the border by taking borderline minority class instances into account while producing synthetic data.ADASYN creates more minority samples near the decision border, helping to develop the classification boundary [4]. Based on the percentage of majority samples in the KNN sets of the minority class, this technique will calculate the number of synthesized minority samples.ROS replicates minority class instances and inserts them into the same class to provide a balanced training dataset, which is the oldest oversampling technique [5].

Various ensembled-based approaches have already been used to improve the predictive performance of models such as bagging [6], boosting, and stacking [7]. In bagging [6] bootstrapped approach is used for homogeneous classifiers to maintain diversity and reduce bias. Boosting [8] is an ensemble modeling technique that attempts to build a strong classifier from the pool of weak classifiers. It is done by building a model by using weak models in series. While bagging and boosting used homogeneous weak learners for ensemble, stacking often considers heterogeneous weak learners, learns them in parallel, and combines them by training a meta-learner to output a prediction based on the different weak learner's predictions [9].

Hyper-parameter optimization will improve the predictive performance of the individual classifier [10]. In the literature, various search techniques are used for hyperparameter optimization such as grid search, random search, etc [11]. In the stacked ensemble parameter optimization of the base model as well as the meta-model is also important otherwise, it may impact the performance of the ensemble model [10]. Various meta-heuristic algorithms such as evolutionary-based, nature-inspired algorithms are used for hyperparameter optimization [12]. Particle swarm optimization (PSO) is an algorithm for swarm intelligence based on stochastic and population-based adaptive optimization inspired by the social behavior of bird flocks and fish swarms [13].

The best configuration of the stacking model will give an effective predictive performance. so selecting optimal base models and meta-models are important in the stacking approach [14].

In disease diagnosis procedures, a physician has to make accurate decisions after analyzing the patient's data. These decisions are very crucial for early diagnosis and sometimes lead to erroneous diagnosis [15]. Hence life-threatening disease diagnoses such as diabetes, chronic kidney disease, heart disease, breast cancer, etc. are challenging. In recent years, most researchers interested in using a combination of optimization techniques and ML algorithms for the early detection of diseases and to improve classification accuracy. Several machine learning algorithms, including LR, KNN, SVM, and DT, have been used to diagnose diseases[16]. Similarly, various meta-heuristic optimization algorithm approaches have been used to detect diseases accurately. However, no researchers concluded that any single classifier is effective in predicting various diseases. As a result, the research community has focused on the use of ensemble learning approaches.

It is also well recognized that individual classifiers' performance is a poor comparison with ensemble approaches. Ensemble-based learners are based on the assumption that different types of classification errors are generated by different base classifiers, and they integrate all of these individual learners in terms of robustness and accuracy. Most of the ensemble models are boosting accuracy while overlooking ensemble complexity and diversity.

Most of the researchers attempted to improve the diagnosis, prediction, classification, therapy, etc using various machine learning algorithms. They have resulted in improvements in profound conventional methods. The current research focuses on Multiple Classifier Systems (MCS) or Classifier Ensemble (CE). By integrating these systems, we may overcome the drawbacks of the traditional approach based on single classifiers and make effective decisions at many levels. The varied biases and variances of each classifier model are exploited by ensembles containing diverse individual classifiers. The model's generalization error can be broken down into bias, variance, and noise.

MCS demonstrates significant complexity in terms of difficulty in separating classes. Three processes are typically involved in the formation of an MCS: (i) pool generation, (ii) classifier selection, and (iii) classifier aggregation [17]. A pool of classifiers is provided in the first step. The MCS training or testing phases, which correspond to a static or dynamic selection, are when the optional classifier selection step can be carried out. Numerous selection options for ensemble selection or single classifier utilizing dynamic or static algorithms may be found in the literature [18].

Despite the fact that there is no proper correlation between accuracy and diversity [19] pool generation is commonly carried out by investigating the concept of developing diverse classifiers in the sense that they each generate different prediction errors and are thus expected to complement one another. Some well-known approaches in the literature investigate diversity to produce homogeneous ensembles given a base inducer. All of these effective approaches involve manipulating data horizontally, vertically, or both. While the latter trains classifiers on samples representing only a portion of the original feature space, such as Random Subspace [20], the former trains classifiers in separate subsets of instances, such as Bagging and Boosting.

The "no free lunch" principle [21] serves as the driving force behind ensemble learning techniques [22], which combine predictions from various machine learning models using various techniques. Compared to individual models, the performance is frequently noticeably improved [23], [24].

The most common ensemble methods right now are bagging and boosting, with the most well-known implementations being Random Forest and Ada Boost, respectively. These techniques can be used in a variety of fields, such as face recognition [25, 26], anomaly detection [27], and medicine [28, 29]. A popular ensemble technique for training individual learners using randomly selected portions of the initial training dataset is bagging. Given the bias-variance breakdown of error for machine learning models, the aggregation of several learners reduces the variance of the model while keeping its bias constant. The bias of the machine learning algorithm is the resemblance between the average prediction of the models and the actual data, and its variance is the difference between the predictions, given many models of the same machine learning algorithm are trained on various training datasets [30].

A popular application of bagging that makes use of decision trees and adds new characteristics to the sampling process is random forest [31]. In contrast to bagging, which takes into account all features, a random forest only takes into account a random subset of the characteristics during each split. To further reduce the computational expense of figuring out how to split the data, decision trees include an addition called Extra-tree [32] that uses random splits. Fewer correlated decision trees are possible with random forest and extra trees than with bagging, which is a desirable trait because it allows alternative features to be represented rather than being dominated by strong predictors [33].

As greater performance is the only benefit of lower variance, bagging has the drawback that the bias of a single machine learner is typically the same as the bias of the combined model [34]. For instance, decision trees will likely underfit the data and yield high bias errors if the data is not appropriate for them. An ensemble of decision trees will then have the same bias error as a single decision tree in that scenario. As a result, errors made by the individual learners that result in high bias (model under-fitting) are carried over to the predictions that are made collectively. To tackle this issue, it becomes sense to focus optimization on bags that are more representative than bootstrapped bags. According to several researchers who used bootstrapping to choose the best size for each bag, employing bags the same size as the entire training set is inefficient [35, 36].

These, however, operate with the best-sized bags without actually taking into account the information contained in each bag. However, it has also been suggested that we should optimize the data contained in each bag and concentrate on the specific issue of imbalanced data with either over or under-sampling for the data labels [37, 38, 39]. The notion of evolution serves as the inspiration for a subset of optimization algorithms known as evolutionary algorithms (EAs) [40]. In EAs, a population of candidate solutions—cooperate and compete using an error-based metric called fitness.

Due to their adaptability, EAs are frequently used for machine learning models for a variety of tasks, including training recurrent neural networks (neuroevolution) [41], decision tree induction [42], image segmentation [43], and multi-task learning [44, 45]. In the past, efforts have been undertaken to improve bagging, with a primary focus on the study of the contents of the bags in the ensemble [46, 47]. We observe that approaches, such as the set of weights for individual learners or the subsets of features [48, 49], that did not change the training samples in each bag had the advantage of optimizing over a smaller search space.

However, because the set of training samples in each bag stayed the same throughout the evolution process, these methods were unable to optimize them. This may be a significant drawback if an informative representation of the data is needed, such as when an imbalanced situation calls for either an oversampling of the majority class or an undersampling of the minority class. Additionally, it is crucial to change the content of the bags during model learning from them (individual learners), as doing so would otherwise result in highly linked learners. By maximizing accuracy and minimizing the amount of data needed, Garcia and Herrera [50] proposed a series of methods for under-sampling for classification with imbalanced datasets.

With large datasets, the absence of a focused optimization method cannot provide a steady learning process. Proposed population-based bagged ensemble learning framework in which the TLBO-based algorithm updates and shuffles the data across the bags to iteratively increase the ensemble's diversity. Dynamic weight updation allows us to maintain the balance between accuracy and diversity.

Ensemble methods combine the predictions of multiple classifiers to produce more accurate and robust predictions compared to individual classifiers. The selection of an appropriate ensemble is crucial in harnessing the benefits of ensemble methods. This work is inspired by the traditional Teaching-Learning-Based Optimization [51] (TLBO) approach for the selection of an optimized ensemble of classifiers. The TLBO algorithm, inspired by the teaching and learning processes in a classroom, facilitates the exploration and exploitation of the solution space to identify superior ensemble configurations.

The objective of this study is to find an ensemble that not only achieves high prediction accuracy but also exhibits diversity among its constituent classifiers. High accuracy ensures reliable predictions, while diversity promotes the ensemble's ability to capture different aspects of the underlying data distribution, leading to improved generalization performance. The BA-TLBO algorithm provides an effective framework to simultaneously optimize accuracy and diversity by iteratively updating the ensemble.

The proposed approach leverages a set of diverse classifiers, including LR [52], KNN [53], DT [54], and SVM [55]. Each classifier is associated with a hyperparameter grid for tuning its configuration, allowing for fine-grained optimization. The BA-TLBO algorithm initializes a set of bags randomly and evaluates their performance based on accuracy and diversity metrics. The worst-performing bag is replaced with a new bag randomly sampled from the dataset, encouraging exploration and convergence toward better ensemble configurations.

To evaluate the performance of the optimized ensemble, extensive experimentation is conducted on real-world datasets. The dataset is divided into training and test sets, with the training set used for optimization and the test set used for evaluation. The ensemble predictions are obtained using majority voting [56], where each classifier's prediction contributes to the final decision. The evaluation metrics include accuracy, AUC, precision, recall, and F1-measure, providing a comprehensive assessment of the ensemble's predictive capabilities.

The contributions of this work lie in the application of TLBO optimization to ensemble classifier selection, combining accuracy and diversity metrics for ensemble evaluation, and the comprehensive evaluation of the optimized ensemble using multiple performance metrics. The experimental results demonstrate the effectiveness of the proposed approach in constructing an optimized ensemble that outperforms individual classifiers and traditional ensemble methods.

A doctor must make a precise conclusion in disease diagnosis procedures after reviewing the patient's data. These choices can sometimes result in incorrect diagnoses but are very important for early diagnosis. Therefore, it can be difficult to diagnose life-threatening diseases such as breast cancer, lung cancer, heart, diabetes, chronic kidney disease, etc [57] [58].

In the literature, several studies have demonstrated that ensemble learning performs better than individual base models. creating a systematic approach to merge base models is a bit challenging task. The performance of models can be evaluated on two factors i.e. bias and variance. The interactions between the data and the selected model affect both. Variance is a model's sensitivity to changes in training data, whereas bias is a model's knowledge of the underlying correlation between features and target output. When trying to predict the output, the main task is to find the most appropriate function with a low-bias and low-variance prediction model it can reduce the variance using techniques like bagging (which helps reduce variance by averaging the outcomes of various models and thus reducing the chance of misclassification) and cross-validation (as most of the data is used for validation). Similarly, bias can be reduced with the help of boosting.

It is also well recognized that as compared to ensemble methods, the performance of an individual classifier is subpar [59]. Based on the premise that different base classifiers produce different types of classification errors, ensemble-based learners combine the resilience and accuracy of all of these diverse learners. The majority of ensemble models increase accuracy while ignoring the complexity and variety of the ensemble.

Majority of researchers used various machine learning methods to try and enhance diagnosis, prediction, categorization, therapy, etc [60]. They have led to significant advancements in conventional techniques. Multiple Classifier Systems (MCS) or Classifier Ensembles (CE) are the focus of the present research. By integrating various systems, one can make decisions that are effective on multiple levels. The various biases and variances of each classifier model assist ensembles composed of diverse classifiers. The generalization error of the model is made up of bias, variance, and noise.

There are various meta-heuristic evolutionary algorithms are already used in disease diagnoses [61] Such as GA, PSO, etc.In the multi-level ensemble approach, classifier placement should be optimal to get maximum fitness value here possible combinations are more so GA is used to search for optimal classifier placement.

Ensemble-based approaches are used with meta-heuristic optimization algorithms to generate the best set of classifier combinations which will improve the predictive performance. Multi-level ensemble learning with novel fitness functions will significantly improve predictive model performance.

When trying to predict the output, the main task is to find the most appropriate function with a low-bias and low-variance prediction model it can reduce the variance using techniques like bagging (which helps reduce variance by averaging the outcomes of various models and thus reducing the chance of misclassification) and cross-validation (as most of the data is used for validation). Similarly, bias can be reduced with the help of boosting.

Based on the premise that different base classifiers produce different types of classification errors, ensemble-based learners combine the resilience and accuracy of all of these diverse learners. The majority of ensemble models increase accuracy while ignoring the complexity and variety of the ensemble.

Ensemble-based approaches are used with meta-heuristic optimization algorithms to generate the best set of classifiers combinations which will improve the predictive performance. Multi-level ensemble learning with novel fitness functions will significantly improve predictive model performance.

1.1 Motivation & Objectives

1.1.1 Motivation

In disease diagnosis, most of the datasets are class-imbalanced. ML models are biased toward the majority of samples in class imbalanced data. To address this problem various oversampling approaches are used. Directly applying oversampling techniques does not guarantee the improvement of performance due to noise while generating synthetic data. To overcome this we combined oversampling and ensemble learning to improve the predictive performance. However, in the ensemble approach, most of the researchers attempted the optimization of base classifiers with limited research on the optimization of meta classifiers. In the stacking approach if we are increasing the number of layers there should be an effective meta-model that can combine the predictions of the previous layers. we have optimized base classifiers with grid search and the last level meta-model with PSO with a novel fitness function. Research questions are to be addressed with the proposed approach.

RQ 1. Can we improve predictive performance with an oversampling and ensemble approach?

RQ 2. Extended stacking approach (Multi-level) is better in prediction than the basic stacking approach?

RQ 3. Does final Meta-model parameter optimization make any improvement in overall performance?

RQ 4. How does the proposed model have more significance than other base-level models

The motivation behind this is to address the critical gap in effective disease diagnosis on a global scale. Despite the advancements in medical science and technology, the challenges posed by the intricate mechanisms and varied symptoms of diseases persistently hinder the development of models for early diagnosis and efficient treatment. While ensemble-based machine learning models have shown promise in aiding doctors with early diagnosis, a major obstacle remains: selecting diverse classifiers to enhance the overall performance of these models.

Past attempts to improve classification accuracy using ensemble learning approaches have faced limitations and yielded less-than-optimal outcomes. Therefore, the motivation of this research is to introduce a novel solution – a diversity-based evolutionary ensemble framework powered by GA. This framework aims to enhance the predictive performance of ensemble models for disease diagnosis by incorporating a variety of classifiers. These classifiers are employed using a bootstrapped approach to create 20 diverse base learners, each generated from five bootstrapped bags and employing 5-FCV.

The motivation extends to proposing a novel fitness function to further improve predictive performance. Robustness testing, involving running the model 20 times and assessing average performance and ensemble complexity, adds credibility to the research's approach.

In essence, the motivation of this study is to tackle the pressing need for enhanced disease diagnosis by introducing an innovative ensemble framework that combines diversity-based classifiers with robust optimization techniques, ultimately leading to superior diagnostic accuracy across a range of benchmark disease datasets.

Balancing accuracy and diversity in ensemble learning is essential to harness the full potential of multiple models while mitigating their limitations. The motivation behind achieving this balance stems from several key reasons:

- **Improved Robustness**: Ensemble models with diverse predictions are less likely to be influenced by outliers or noise in the data. By combining models with varying strengths and weaknesses, the ensemble becomes more robust and capable of handling complex and uncertain situations.

- **Reduced Overfitting**: Highly accurate individual models might overfit specific patterns in the training data, leading to poor generalization. Diversity introduces variability in the predictions, reducing the risk of overfitting and ensuring better performance on unseen data.

- **Enhanced Generalization**: A balanced ensemble strikes a harmony between models that excel in different aspects of prediction. This leads to improved generalization performance as the ensemble collectively leverages the specialized knowledge of each base model.

- **Bias Reduction**: Models might inherently carry biases due to their training data or algorithms. Diversity can help reduce these biases and produce more unbiased and fair predictions, especially in sensitive applications like healthcare.

- **Adaptability**: Ensembles with both accuracy and diversity are more adaptable to changing data distributions, concept drift, or evolving patterns, ensuring that the ensemble's performance remains reliable over time.

- **Handling Uncertainty**: Diverse models provide different perspectives on uncertainty estimation, enabling the ensemble to better quantify and manage uncertainty in predictions, which is particularly important in critical applications like disease diagnosis.

- **Ensemble Paradox**: It has been observed that ensembles composed of less accurate but diverse models can outperform ensembles of more accurate but less diverse models. This paradox underscores the importance of achieving the right balance.

In the ensemble learning approach for effective disease, diagnosis needs an optimal classifier pool to improve predictive model accuracy and effective sensitivity and specificity. Various ML classifiers are used in disease diagnosis, however, classifier selection and also placement in the ensemble framework will impact the ensemble model performance. The effective approach should select optimal classifiers from a pool of classifiers and place them in the ensemble framework. For example, a pool of n classifiers has to choose m classifiers for the proposed model, possible $\binom{n}{m}$ ways. Again, these $\binom{n}{m}$ possibilities can be arranged in $m!$ ways. Finding the best m classifiers and their positions from total $\binom{n}{m}m!$ ways is challenging. There are various meta-heuristic optimization algorithms that exist in the literature. GA is used to solve optimization problems in machine learning and it helps solve complex problems that would take a long time to solve. Our proposed model attempted to solve this problem using a 3-level ensemble framework using a GA with an ensemble-based novel fitness function.

1.1.2 Objectives

The following objectives formulated in this thesis are:

Objective 1: Proposed 3-level Stacked-ADASYN-PSO model to address the challenges of class imbalance and optimal ensemble in the multi-level framework. In this study, a novel 3-level stacking approach with ADASYN oversampling technique with PSO Optimized SVM meta-model (Stacked-ADASYN-PSO) is proposed. Our proposed Stacked-ADASYN-PSO model uses base models such as LR, KNN, SVM, DT, and MLP in layer 0. In layer-1 three meta classifiers namely LR, KNN, and Bagging DT are used. In layer-2 PSO optimized SVM is used as the final meta-model to combine the previous layer predictions.

Objective 2: Proposed ensemble based model using GA to address challenges of the diversity based classifier selections to improve disease diagnosis performance. To improve the disease diagnosis performance in this study a novel diversity-based evolutionary ensemble framework with a GA is proposed. To improve the predictive performance used five diversity-based classifiers such as KNN, SVM, LR, and DT using the bootstrapped approach to generate 20 diverse base learners with five bootstrapped bags using 5-FCV.

Objective 3: Ensemble learning has emerged as a powerful approach to disease diagnosis by combining multiple classifiers to improve predictive accuracy and robustness. However, selecting an optimal ensemble configuration and balancing accuracy and diversity remains a challenge. This study proposed a Bagging Approach with Teaching-Learning-Based Optimization (BA-TLBO) algorithm for ensemble optimization in disease diagnosis. To create a compromise between accuracy and diversity, a novel fitness function was proposed that incorporated ensemble mean accuracy and mean diversity and used hamming distance as a diversity measure. In addition, dynamic weight updation is proposed to optimize the weights over the iterations in the BA-TLBO optimization process to balance exploration and exploitation. And also used dynamic bag size over the iterations to balance the bias and variance and it improves the generalization. By iteratively selecting and replacing bags in the ensemble, the BA-TLBO explores different classifier combinations to achieve high accuracy while maintaining diversity. .

Objective 4: Effective disease diagnosis is a critical unmet need on a global scale. The intricacies of the numerous disease mechanisms and underlying symptoms make developing a model for early diagnosis and effective treatment extremely difficult. ML can help to solve some of these issues. Recently, various ensemble-based ML models have benefited clinicians in early diagnosis. However, one of the most difficult challenges in multi-level ensemble approaches is the classifier selection and their placement in the ensemble framework as it improves the overall performance. Let m classifiers have to select from n classifiers there are $\binom{n}{m}$ ways. Again, these $\binom{n}{m}$ possibilities can be arranged in $m!$ ways. Finding the best m classifiers and their positions from total $\binom{n}{m}m!$ ways is a challenging and hard problem. To address this challenge, a dynamic three-level ensemble framework is proposed. A nested GA and novel fitness function are employed to optimize the classifier selection and their placement in a three-level ensemble framework. Our approach used eleven classifiers and chose seven classifiers by maximizing the novel fitness function.

1.2 Summary of the contributions

In this section, an overview of chapter-wise contributions to this thesis has been presented. Each subsection presents a summary of the contributions of the corresponding chapter.

1.2.1 A Novel Stacking Framework with PSO Optimized SVM for Effective Disease Classification

In this work, in order to address the class imbalance and optimal ensemble to improve the disease diagnosis performance, a three-level stacking framework is proposed. The proposed model is fed with the pre-processed dataset. During the pre-processing step, IQR was used for outlier removal and ADASYN for class imbalance. This pre-processed dataset is used to train the proposed 3-level stacking framework. In this stacking framework, level 0 learners (LR, SVM, DT, KNN, and MLP) and level 1 learners (Bagged DT, KNN, and LR) are optimized using grid search. The level 2 learner i.e., SVM is optimized with PSO. For a better optimization process, a novel fitness function is proposed. The proposed model experimented on PID, SHD, CHD, CKD, and WBC datasets. The proposed model is compared with different combinations of base learners and outperformed in terms of all the performance measures. Further, the proposed model is compared with SOTA ensemble and non-ensemble methods in terms of accuracy, AUC, specificity, and precision and it outperformed all the models in terms of AUC and accuracy on all the datasets. Finally, to prove the robustness of the proposed model a paired statistical t-test is performed. The statistical test proved that the proposed model significantly differs from all the base-level models.

1.2.2 A Novel Diversity-based Ensemble Approach with Genetic Algorithm for Effective Disease Diagnosis

This work aims to improve the diversity and reduce training time and varience for that proposed bagging approach with an evolutionary algorithm, and evaluated the performance of the individual classifiers on various types of disease datasets. To improve the performance of the models performed bootstrapped aggregation of the training set and evaluated the performance of individual classifiers w.r.t to data bag to further improve the performance an ensemble approach using GA and computed fitness using the proposed novel fitness function. In our proposed approach we have used four classifiers such as LR, KNN, SVM, and DT with fine-tuned hyperparameters using grid search. Further, 5-FCV was applied on the training part and divided into 5 folds and applied GA as an evolutionary search for optimal ensemble candidates of 20 learners trained on the bootstrapped data. Using 5-FCV the validation set is used to evaluate the fitness of each chromosome and to evaluated the robustness of the proposed diversity based ensemble model undergone 20 runs and considered the mean of 20 runs as proposed model performance and also considered mean of diversity based selected classifiers.

1.2.3 Enhancing Disease Diagnosis Accuracy and Diversity through BA-TLBO Optimized Ensemble Learning.

The proposed BA-TLBO approach gives promising results in constructing optimized ensembles across the PID, SHD, SLC, and WBC datasets. The experimental analysis demonstrates that the optimized ensemble exhibits improved performance compared to individual classifiers and potentially other baseline ensemble methods. In the proposed study we have introduced a novel fitness function that balances accuracy and diversity and gives good exploration and exploitation in the BA-TLBO optimization process. And also dynamically updated the accuracy and diversity weight, making the proposed model adaptable and robust. And also analyzed various diversity measures such as Hamming, Bhattacharya, Entropy, and Q statistics out of these Hamming distance-based diversity measure performance is superior compared to others.

And also analyzed the worst bag optimization and compared it with the best bag optimization and observed that with the best bag possibility of overfitting risk and not focused weak component in the ensemble. so finally hamming distance-based diversity and worst bag-based optimization will give effective results in disease diagnosis. so it is recommendable to use hamming distance-based diversity with BA-TLBO-based worst bag optimization.

Overall, the proposed BA-TLBO ensemble approach demonstrates its potential to improve predictive performance and generate robust predictions across different datasets. Further research could focus on exploring additional classifiers, enhancing the optimization algorithm, or considering other optimization techniques to improve ensemble performance.

1.2.4 Nested Genetic Algorithm-based Classifier Selection and Placement in Multi-Level Ensemble Framework for Effective Disease Diagnosis

In this study, a dynamic three-level ensemble framework is proposed. It is experimented with using four benchmark disease datasets from UCI and Kaggle repositories. All these disease datasets have undergone the pre-processing stage. After the pre-processing stage, nested GA is employed to optimize the classifiers and their positions in the proposed three-level ensemble framework. Outer GA selects the best classifiers and inner GA is used to optimize the selected classifiers' positions in the framework. Further, proposed a novel fitness function for a better solution. Our approach used eleven classifiers and chose seven classifiers by maximizing the ensemble based fitness function.

The performance of the proposed model is compared with SOTA ensemble and non-ensemble models, and the proposed approach gave better results in terms of accuracy, AUC, precision, recall, specificity, and G-measure. Next, ROC-AUC analysis is carried out and the proposed model achieved superior performance than other ensemble models such as RF, BC, GBC, and XGB. Next, our proposed framework performance is evaluated level-wise, and the proposed 3-level ensemble approach gives superior performance when compared to others.

Further, sensitivity and specificity analysis of the proposed model on the top 5 performed disease datasets is carried out. In terms of sensitivity and specificity, our proposed model performance is superior when compared with SOTA ensemble and non-ensemble models.

1.3 Organization of the Thesis

The main focus of the thesis is to address the challenges of ensemble based models to enhance the disease diagnosis by introducing ensemble-based machine-learning approaches. These approaches are designed to improve the accuracy and effectiveness of early disease diagnosis, particularly in the context of complex disease mechanisms and varied patient symptoms. The thesis aims to enhance the performance of ensemble models by strategically selecting diverse classifiers, optimizing their configurations, and applying innovative techniques such as GA and novel fitness functions.

The thesis centers on the development and evaluation of these ensemble frameworks, which are tailored to overcome the limitations of traditional methods in disease diagnosis. The core emphasis is on achieving robust predictive performance by incorporating a range of diverse classifiers, including KNN, SVM, LR, and DT. Additionally, the thesis focuses on addressing the challenges of class imbalance and optimizing ensemble configurations to strike the right balance between diversity and accuracy.

In summary, the main focus of the thesis revolves around introducing innovative ensemble-based solutions to enhance disease diagnosis accuracy. Through the strategic integration of diverse classifiers, optimization techniques, and robust evaluations on benchmark datasets, the research aims to contribute to the advancement of early disease diagnosis and improve patient care on a global scale.

The thesis consists of seven chapters as follows. The content of each of these chapters is described briefly below:

Chapter 1: This chapter provides a comprehensive introduction to the thesis topic and briefly outlines the objectives that are pursued throughout the research.

Chapter 2: In this chapter, an overview of state-of-the-art works in the field of disease diagnosis using optimization of ensemble approaches is provided, with a specific focus on class imbalance, classifier selection, diversity, and optimal placement of classifiers in the ensemble framework.

Chapter 3: In this chapter, a novel 3-level stacking approach with ADASYN oversampling technique with PSO Optimized SVM meta-model (Stacked-ADASYN-PSO) is proposed. Our proposed Stacked-ADASYN-PSO model uses base models such as LR, KNN, SVM, DT, and MLP in layer 0. In layer-1 three meta classifiers namely LR, KNN, and Bagging DT are used. In layer-2 PSO optimized SVM is used as the final meta-model to combine the previous layer predictions.

Chapter 4: This chapter proposes a novel diversity-based evolutionary ensemble framework with a GA is proposed. To improve the predictive performance of four diversity-based classifiers such as KNN, SVM, LR, and DT using the bootstrapped approach to generate 20 diverse base learners with five bootstrapped bags using 5-FCV. Also to improve the predictive performance proposed a novel fitness function. To test the robustness, the model was run 20 times and the average performance and average ensemble complexity of the proposed model were computed.

Chapter 5: This chapter proposed a BA-TLBO algorithm for ensemble optimization in disease diagnosis. To create a compromise between accuracy and diversity, a novel fitness function was proposed that incorporated ensemble mean accuracy and mean diversity and used hamming distance as a diversity measure. In addition, dynamic weight updation is proposed to optimize the weights over the iterations in the BA-TLBO optimization process to balance exploration and exploitation. And also used dynamic bag size over the iterations to balance the bias and variance and it improves the generalization. By iteratively selecting and replacing bags in the ensemble, the BA-TLBO explores different classifier combinations to achieve high accuracy while maintaining diversity.

Chapter 6: This chapter proposed a dynamic three-level ensemble framework using a nested GA and novel fitness functions are employed to optimize the classifier selection and their placement in a three-level ensemble framework. Our approach used eleven classifiers and chose seven classifiers by maximizing the novel fitness function. The proposed model experiments on four disease datasets.

Chapter 7: This chapter provides conclusions of the thesis, essential outcomes of the contributions, and the scope for future expansion of the research conducted in this thesis.

Chapter 2

Related Work

A comprehensive literature review of different tasks is presented in this chapter. The literature related to optimized ensemble approaches in disease diagnosis is discussed in Section 2.1, while Section 2.2 covers the literature on various bagging approaches used and optimization of bagging approaches. Furthermore, Section 2.3 focuses on the literature related to various diversity measures and optimized configurations. Finally, in the last section classifier placements in the ensemble framework followed by a summary is provided in Section 2.4.

2.1 Ensemble based models and optimization

Kalagotla et al. [62] proposed a novel stacking technique on PID and compared the AdaBoost and stacking revealing that the accuracy of stacking a heterogeneous ensemble 78.2% outperforms the AdaBoost a homogeneous ensemble 76.54%. D. Joshi et al. [63] used the R tool on the PID dataset to predict T2DM. They applied DT and LR classifiers on PID and reported 78.26% and 74.48% accuracies respectively. S. Arukonda et al. proposed a disease diagnosis ensemble model. This study used four diversity-based classifiers on five data bags and optimized classifiers from a pool of 20 diverse learners using GA. This study used PID, SHD, CKD, and WBC disease datasets to test the robustness of the models. Accuracies are 90.91%, 96.05%, 97.56%, and 98.08% respective to PID, CKD, SHD, and WBC datasets.

Singh et al. [64] proposed a stacking approach on PID and evaluated the predictive performance of various ensemble approaches such as Bagging (L-SVM), Bagging (RBF-SVM), Bagging (Poly-SVM), Bagging (REP), Bagging (4.5), Ada boost (DS), Ada boost (C4.5), Random Subspace Method (RSM), Random Forest, Majority Voting (MV), Stacking, Stacking (LR), Stacking (NSGA-II) the proposed system achieve the highest accuracy of 83.8%, the sensitivity of 96.1%, specificity of 79.9%, f-measure of 88.5% and area under ROC curve of 85.9%. S.Arkonda et al. proposed a model for Lung cancer is one of the most common cancer-related disorders with a high mortality rate, which is mostly owing to the late detection of malignancy.Mohapatra et al. [65] proposed a two-level stacking approach for detecting heart irregularities and predicting Cardiovascular disease and pre-processed with outlier detection and the stacking of classifiers for predicting heart diseases . In this study, various classifiers were used to take advantage of their differences in strengths. Using MLP as the meta-learner, Obtained results with 92% accuracy. The proposed stacked classifier outperformed the traditional machine learning classifiers better in terms of overall parameter comparison with a precision of 92.6%, a sensitivity of 92.6%, and a specificity of 91%.

Sampath et al. proposed a model for cancer disease. Cancer is still a fatal illness with numerous subtypes, posing numerous hurdles in biomedical research. Tiwari et al. proposed an Ensemble framework for cardiovascular disease prediction proposed framework consist of Stacking Ensemble learning which adds diversity to the classifier experimented on IEEE Data Port proposed stacked ensemble attained an accuracy of 92.34% [9]. Obaidat et al. proposed a stacking ensemble model for predicting heart attacks and combined a group of three base-level classifiers such as Naïve Bayes, Random Forest, and Extreme Gradient Boosting (XGBoost) in the predictive model. Kolukisa et al. proposed a classification method with ensemble feature selection for coronary artery disease diagnosis and achieved 85.55% and 85.47% accuracy for the Statlog and Cleveland data sets respectively [66]. D. Joshi and K. Dakhal have used the R tool on the PID dataset to predict T2DM [63]. They applied DT and LR classifiers on PID and reported 78.26% and 74.48% accuracies respectively. Kumari et al. have proposed an ensemble method with a soft voting classifier on the PID dataset [67]. They combined three classifiers RF, LR, and Naive Bayes (NB) with a soft voting classifier. They also compared this model with Adaptive Boosting (Ada Boost), RF, Bagging, Gradient Boost, Extreme Gradient Boosting (XGBoost), and CatBoost (CAT) algorithms. Their proposed method obtained an accuracy of 79.04% which is the best among other state-of-the-art ensemble methods. Bashir et al. combined three decision tree-based classifiers namely CART, ID3, and C4.5 on the PIMA dataset and got an accuracy of 76.5% [68].

Kumar Kalagotla et al. have proposed a stacking ensemble approach on the PID dataset [62]. They pre-processed the PID dataset with IQR and min-max scaling. They used MLP and SVM as base learners and LR as meta-learners. The proposed staking approach achieved an accuracy of 78.2%. Rajendra et al. have proposed a stacking framework with KNN, DT, and NB as base learners and LR as meta learner [69]. It reported an accuracy of 77.83% on the PID dataset. In this work, SVM, KNN, DT, MLP, Bagging DT, and Bagging KNN are used in the proposed stacking framework. While DT benefits from flexibility, unambiguity, robustness, and other factors, K-NN benefits from simplicity and non-parametric behavior [70]. SVM is renowned for its excellence in classification tasks, and one of its groundbreaking features is the ability to generalize high-dimensional data [71] and MLP can be applied to resolve challenging nonlinear issues. Additionally, ensemble approach stacking offers diversity, stability, and exceptional performance. Chittora, Pankaj, et al. [72] used seven classification models to predict chronic kidney disease and the highest accuracy reported was 90.73% with random tree classifier . Tiwari et al. have used a stacking-based ensemble model for the prediction of cardiovascular disease and it obtained an accuracy of 92.34% . Al-Azzam et al. proposed a model for breast cancer prediction. In this study, various classification algorithms for supervised and semi-supervised learning were proposed. These models were evaluated using Random Forest, Xgboost, and Gradient Boosting on the Wisconsin cancer dataset and reported 96.00%, 97.00 %, and 93.00 % accuracies respectively.

2.2 Bagging approaches for disease diagnosis

Kolukisa et al. proposed a classification method with ensemble feature selection for coronary artery disease diagnosis and achieved 85.55% and 85.47% accuracy for the Statlog and Cleveland data sets respectively [66]. D. Joshi and K. Dakhal have used the R tool on the PID dataset to predict T2DM [63]. They applied DT and LR classifiers on PID and reported 78.26% and 74.48% accuracies respectively. Kumari et al. have proposed an ensemble method with a soft voting classifier on the PID dataset [67]. They combined three classifiers RF, LR, and NB with a soft voting classifier. They also compared this model with AB, RF, BC, GB, XGBoost, and CAT algorithms. Their proposed method obtained an accuracy of 79.04% which is the best among other state-of-the-art ensemble methods. Bashir et al. combined three decision tree-based classifiers namely CART, ID3, and C4.5 on the PIMA dataset and got an accuracy of 76.5% [68].

Kumar Kalagotla et al. has proposed a stacking ensemble approach on the PID dataset [62]. They pre-processed the PID dataset with IQR and min-max scaling. They used MLP and SVM as base learners and LR as meta-learners. The proposed staking approach achieved an accuracy of 78.2%. Rajendra et al. have proposed a stacking framework with KNN, DT, and NB as base learners and LR as meta learner [69]. It reported an accuracy of 77.83% on the PID dataset. In this work, SVM, KNN, DT, MLP, Bagging DT, and Bagging KNN are used in the proposed stacking framework. While DT benefits from flexibility, unambiguity, robustness, and other factors, K-NN benefits from simplicity and non-parametric behavior [70]. SVM is renowned for its excellence in classification tasks, and one of its groundbreaking features is the ability to generalize high-dimensional data [71] and MLP can be applied to resolve challenging nonlinear issues. Additionally, ensemble approach stacking offers diversity, stability, and exceptional performance.

Chittora, Pankaj, et al. used seven classification models to predict chronic kidney disease and the highest accuracy reported was 90.73% with random tree classifier [72]. Tiwari et al. have used a stacking-based ensemble model for the prediction of cardiovascular disease and it obtained an accuracy of 92.34% [9]. Al-Azzam et al. proposed a model for breast cancer prediction. In this study, various classification algorithms for supervised and semi-supervised learning were proposed. These models were evaluated using Random Forest, Xgboost, and Gradient Boosting on the Wisconsin cancer dataset and reported 96.00%, 97.00 %, and 93.00 % accuracies respectively.

2.3 Diversity based optimization

The most common ensemble methods right now are bagging and boosting, with the most well-known implementations being Random Forest and Ada Boost, respectively. These techniques can be used in a variety of fields, such as face recognition [25, 26], anomaly detection [27], and medicine [28, 29]. A popular ensemble technique for training individual learners using randomly selected portions of the initial training dataset is bagging. Given the bias-variance breakdown of error for machine learning models, the aggregation of several learners reduces the variance of the model while keeping its bias constant. The bias of the machine learning algorithm is the resemblance between the average prediction of the models and the actual data, and its variance is the difference between the predictions, given many models of the same machine learning algorithm are trained on various training datasets [30].

A popular application of bagging that makes use of decision trees and adds new characteristics to the sampling process is random forest [31]. In contrast to bagging, which takes into account all features, a random forest only takes into account a random subset of the characteristics during each split. To further reduce the computational expense of figuring out how to split the data, decision trees include an addition called extra-tree [32] that uses random splits. Fewer correlated decision trees are possible with random forest and extra trees than with bagging, which is a desirable trait because it allows alternative features to be represented rather than being dominated by strong predictors [33].

As greater performance is the only benefit of lower variance, bagging has the drawback that the bias of a single machine learner is typically the same as the bias of the combined model [34]. For instance, decision trees will likely underfit the data and yield high bias errors if the data is not appropriate for them. An ensemble of decision trees will then have the same bias error as a single decision tree in that scenario. As a result, errors made by the individual learners that result in high bias (model under-fitting) are carried over to the predictions that are made collectively. To tackle this issue, it becomes sense to focus optimization on bags that are more representative than bootstrapped bags. According to several researchers who used bootstrapping to choose the best size for each bag, employing bags the same size as the entire training set is inefficient [35] and [36].

These, however, operate with the best-sized bags without actually taking into account the information contained in each bag. However, it has also been suggested that we should optimize the data contained in each bag and concentrate on the specific issue of imbalanced data with either over or under-sampling for the data labels [37, 38, 39]. The notion of evolution serves as the inspiration for a subset of optimization algorithms known as EAs [40]. In EAs, a population of candidate solutions cooperate and compete using an error-based metric called fitness.

Due to their adaptability, EAs are frequently used for machine learning models for a variety of tasks, including training recurrent neural networks (neuroevolution) [41], decision tree induction [42], image segmentation [43], and multi-task learning [44, 45]. In the past, efforts have been undertaken to improve bagging, with a primary focus on the study of the contents of the bags in the ensemble [46, 47]. We observe that approaches, such as the set of weights for individual learners or the subsets of features [48], [49], [46], that did not change the training samples in each bag had the advantage of optimizing over a smaller search space.

However, because the set of training samples in each bag stayed the same throughout the evolution process, these methods were unable to optimize them. This may be a significant drawback if an informative representation of the data is needed, such as when an imbalanced situation calls for either an oversampling of the majority class or an under sampling of the minority class. Additionally, it is crucial to change the content of the bags during model learning from them (individual learners), as doing so would otherwise result in highly linked learners. By maximizing accuracy and minimizing the amount of data needed, Garcia and Herrera [50] proposed a series of methods for under-sampling for classification with imbalanced datasets. Using a multi-objective optimization technique, Roshan and Asadi [47] enforced these properties together with good classification performance.

With large datasets, the absence of a focused optimization method cannot provide a steady learning process. These studies [46] and [47] are likewise restricted to unbalanced classification problems with under-sampling, which is vulnerable to biased selection of the majority class and also a potential loss of crucial data if caution is not exercised. The population-based bagged ensemble learning framework in which the TLBO-based algorithm updates and shuffles the data across the bags to iteratively increase the ensemble's diversity. Dynamic weight updation allows us to maintain the balance between accuracy and diversity.

Ensemble methods combine the predictions of multiple classifiers to produce more accurate and robust predictions compared to individual classifiers. The selection of an appropriate ensemble is crucial in harnessing the benefits of ensemble methods. This work is inspired by the traditional TLBO [51] approach for the selection of an optimized ensemble of classifiers. The TLBO algorithm, inspired by the teaching and learning processes in a classroom, facilitates the exploration and exploitation of the solution space to identify superior ensemble configurations.

The objective of this study is to find an ensemble that not only achieves high prediction accuracy but also exhibits diversity among its constituent classifiers. High accuracy ensures reliable predictions, while diversity promotes the ensemble's ability to capture different aspects of the underlying data distribution, leading to improved generalization performance. The BA-TLBO algorithm provides an effective framework to simultaneously optimize accuracy and diversity by iteratively updating the ensemble.

The proposed approach leverages a set of diverse classifiers, including LR [52], KNN [53], DT [54], and SVM [55]. Each classifier is associated with a hyperparameter grid for tuning its configuration, allowing for fine-grained optimization. The BA-TLBO algorithm initializes a set of bags randomly and evaluates their performance based on accuracy and diversity metrics. The worst-performing bag is replaced with a new bag randomly sampled from the dataset, encouraging exploration and convergence toward better ensemble configurations.

To evaluate the performance of the optimized ensemble, extensive experimentation is conducted on real-world datasets. The dataset is divided into training and test sets, with the training set used for optimization and the test set used for evaluation. The ensemble predictions are obtained using majority voting [56], where each classifier's prediction contributes to the final decision. The evaluation metrics include accuracy, AUC, precision, recall, and F1-measure, providing a comprehensive assessment of the ensemble's predictive capabilities.

The contributions of this work lie in the application of TLBO optimization to ensemble classifier selection, combining accuracy and diversity metrics for ensemble evaluation, and the comprehensive evaluation of the optimized ensemble using multiple performance metrics. The experimental results demonstrate the effectiveness of the proposed approach in constructing an optimized ensemble that outperforms individual classifiers and traditional ensemble methods.

In the literature that falls under ensemble learning approaches such as bagging, boosting, and stacking for disease diagnosis and evolutionary approaches such as GA, PSO, and so on various disease datasets such as PID, SHD, CKD, WBC, and so on and reported our findings as follows.

Ensemble-based proposed model and used six algorithms for classification tasks using Machine Learning and Deep Learning the best disease prediction accuracy achieved 88.70% [73].

Srinu et al. [74] proposed a disease diagnosis ensemble model. This study used four diversity-based classifiers on five data bags and optimized classifiers from a pool of 20 diverse learners using GA . This study used PID, SHD, CKD, and WBC disease datasets to test the robustness of the models. Accuracies are 90.91%, 96.05%, 97.56%, and 98.08% respectively, for the PID, CKD, SHD, and WBC datasets.

Singh et al. [64] proposed the prediction of diabetes using a stacking approach that used diversity-based classifiers and achieved 83.8% highest accuracy.

Kumari et al. [67] developed a model for diabetes mellitus classification and prediction using an ensemble technique, with the highest accuracy achieved on the PID dataset being 79.04%.

Ubeyli et al. [75] proposed a model and used neural networks such as recurrent neural networks, probabilistic neural networks, combined neural networks, multi-layer perceptron, and SVM, achieving 98.15%, 98.61%, 97.4%, 91.92%, and 99.54%, respectively . Saifudin et al. [76] proposed a model to reduce the misclassification rate and bagging technique used based on random forest, and bagging achieved 77.10% and Random Forest + Bagging is 84.07%, respectively .Miene et al. [77] proposed a model for predicting heart disease. and accuracy of the models such as 93% and 91%, respectively, to the Cleveland and Framingham datasets.

Kolukisa et al. [66] proposed a model using an ensemble approach of 85.55% and 85.47% accuracies achieved, respectively, to the SHD and Cleveland datasets . D. Joshi and K. Dakhal et al. [63] proposed a model to predict diabetes and achieved 78.26% and 74.48% accuracy with DT and LR classifiers, respectively. Kalgotla et al [62] proposed an ensembled model using RF, LR, NB, and soft voting classifiers.

Kalagotla et al. [62] used an ensemble technique to propose a model on the PID dataset. The proposed model was accurate to 78.2% . Rajendra et al. [69] proposed a model on PID with DT, NB, and KNN as base learners and logistic regression as a meta learner, and the proposed model achieved 77.83% . Chittora et al. [72] proposed a model to predict the CKD disease with a random tree, and the accuracy reported was 90.73% . Tiwari et al. proposed a model for heart disease, and it achieved an accuracy of 92.34%. On the Wisconsin Breast Cancer (WBC) dataset, Al-Azzam et al. offered a model on breast cancer employing ensemble models such as RF, XGB, and GBC. They reported 96.00%, 97.00%, and 93.00% accuracy, respectively.

In the literature that falls under ensemble learning approaches such as bagging, boosting, and stacking for disease diagnosis and evolutionary approaches such as GA, PSO, and so on, various disease datasets such as PID, SHD, CKD, WBC, and so on were reported, as follows:.

Alqahtani et al. [73] proposed an ensemble-based model that used six algorithms for classification tasks using machine learning and deep learning. The best disease prediction accuracy achieved was 88.70% .

Rani et al. proposed an ML-based model to predict heart diseases and also used the RF model for accurate prediction with 86.60% accuracy [78, 79] proposed a model and used RF, NB, SVM, Hoeffding Decision Tree, and Logistic Model Tree (LMT), with Cleveland and RF giving the best accuracy. Nasser et al. [80] proposed a model for lung cancer prediction using the KNN algorithm and achieved an accuracy of 90.7%.

Arukonda et al. proposed a disease diagnosis ensemble model. This study used four diversity-based classifiers on five data bags and optimized classifiers from a pool of 20 diverse learners using GA .This study used PID, SHD, CKD, and WBC disease datasets to test the robustness of the models. Accuracies are 90.91%, 96.05%, 97.56%, and 98.08

Singh et al. proposed the prediction of diabetes using a stacking approach that used diversity-based classifiers and achieved 83.8% highest accuracy [64]. Kumari et al. developed a model for diabetes mellitus classification and prediction using an ensemble technique, with the highest accuracy achieved on the PID dataset being 79.04% [67].

Ubey et al. proposed a model and used neural networks such as recurrent neural networks, probabilistic neural networks, combined neural networks, multi-layer perceptron, and SVM, achieving 98.15%, 98.61%, 97.4%, 91.92%, 99.54% respectively [75]. Saifudinp et al. proposed a model to reduce the misclassification rate and bagging technique used based on random forest, and bagging achieved 77.10% and Random Forest + Bagging is 84.07% respectively [76, 77] proposed a model for predicting heart disease. and accuracy of the models such as 93% and 91%, respectively, to the Cleveland and Framingham datasets.

.

2.4 Summary

The presented literature review extensively covers various aspects of ensemble-based approaches to disease diagnosis. The review focuses on different types of ensemble models, their optimization, bagging techniques, diversity measures, and classifier placements within ensemble frameworks. The studies discussed pertain to a diverse range of diseases, including diabetes, heart disease, lung cancer, chronic kidney disease, and more. Key points from the literature review include:

- **Ensemble Model Types and Optimization**: The review delves into the usage of ensemble techniques like stacking, bagging, and other optimization algorithms to enhance the accuracy of disease prediction models. These models often combine the strengths of different base learners, such as decision trees, support vector machines, k-nearest neighbors, and neural networks, to create more robust predictions.

- **Diversity Measures and Importance**: The literature underscores the significance of diversity among the classifiers within an ensemble. Diversity ensures that various aspects of the data distribution are captured, leading to better generalization performance. Techniques for achieving diversity involve optimizing the composition of training bags and dynamically updating the weights of individual classifiers.

- **Evolutionary Algorithms and TLBO**: The review highlights the use of evolutionary algorithms like GA and PSO, along with TLBO, for selecting and evolving ensemble configurations. These optimization strategies help strike a balance between accuracy and diversity, resulting in improved ensemble performance.

- **Application to Different Diseases**: The studies cover a wide range of diseases, showcasing the versatility of ensemble-based approaches in medical diagnosis. The discussed diseases include diabetes, heart disease, lung cancer, chronic kidney disease, and more.

- **Accuracy Enhancement and Robustness**: Ensemble techniques are employed to enhance the accuracy and robustness of disease prediction models. By combining multiple models, the ensembles can better handle the complexities and variations present in medical datasets.

- **Comparison of Different Approaches**: The literature review compares the performance of various ensemble-based approaches, showcasing the strengths of different methods for different datasets and diseases. This comparison helps in understanding the relative efficacy of each approach.

In summary, the literature review provides a comprehensive overview of the use of ensemble techniques in disease diagnosis. It demonstrates the evolution of these techniques, from traditional methods to more advanced optimization algorithms, and highlights the importance of diversity and optimization in constructing effective ensemble models for accurate and robust disease diagnosis models. In this regard, the following chapters (chapters 3, 4, 5, and 6) delve into more detailed solutions for ensemble-based optimization approaches for disease diagnosis.

Chapter 3

A Novel Stacking Framework with PSO Optimized SVM for Effective Disease Classification

In this chapter three-level stacking framework is proposed and the final meta-model is optimized SVM with PSO for effective disease diagnosis. The novel 3-level stacking approach, Stacked-ADASYN-PSO, proposed in this chapter presents a compelling solution for enhancing predictive performance in disease diagnosis. By integrating ADASYN oversampling and PSO-optimized SVM as a meta-model, this ensemble framework operates by leveraging diverse and complementary insights from various base models. The multi-level aggregation strategy employed in the approach is particularly effective as it enables the exploitation of intricate patterns and relationships within the data. Through the stacking mechanism, lower-level base models contribute their strengths to capture different aspects of the underlying complexities in disease data. The subsequent layers capitalize on the collective intelligence of these base models to generate more refined and accurate predictions.

The introduction of ADASYN oversampling ensures that the ensemble is well-equipped to address class imbalance, a common challenge in medical datasets, thereby enhancing the framework's robustness. Moreover, the utilization of PSO-optimized SVM as the final meta-model further refines the ensemble's predictive capabilities by fine-tuning the model's parameters for optimal performance.

Chapter Organization: Section 3.1 provides the background knowledge related to the proposed model. Section 3.2 presents the proposed methodology. Section 3.3 discusses the experimental settings. Section 3.4 discusses the experimental results and analysis. Section 3.5 provides the discussion and summary of the work.

3.1 Background

This section provides background knowledge relevant to classifier combinations for ensembles and hyperparameter optimization for ensembles, covering topics such as PSO, oversampling techniques, stacking frameworks, and various ML-based classifiers such as LR, DT, KNN, SVM, and MLP. Particle Swarm Optimization is a popular optimization algorithm commonly used for global optimization problems.

3.1.1 Classifier Combination for Ensembles

The selection of classifiers and a combination of those for the best ensemble is a very tedious task [17]. Researchers and data analysts use various machine learning algorithms and choose the best algorithm according to the performance measures [81]. To make the best predictions, a single algorithm may be unable to capture the entire underlying structure of the data [82]. This is where the successful integration of numerous models gathered into a single meta-model has been discovered [82]. Bagging creates numerous versions of predictors and aggregates them by voting on each version and taking the average of them [6]. Bagging meta-estimator and random forest are two algorithms that use the bagging approach [6]. Boosting works similarly to bagging in that it adaptively combines numerous low-performing base learners [8]. Bagging is beneficial for data sets with noisy values, according to experimental results. Stacking is the third approach. It uses the output of selected classifiers on the training data to predict response values using another learning algorithm. The stacking generalization architecture typically consists of two layers. First, in layer 1, there is base classification, which uses basic classifiers to build the ensemble by training the dataset. It generates the second layer's input. Second, in layer 2, the meta-classification integrates the outputs of layer 1 using a meta-classifier to build the final predictive model.

3.1.2 Hyper parameter optimization of classifiers

Best hyperparameters will give a better performance so optimization of hyperparameters is a very crucial step in machine learning [83]. Hyper parameter optimization is the process of selecting the right parameter values for classifiers to build the best prediction model. For optimizing hyperparameters, there are numerous methods available [10], including (1) grid search, (2) random search, (3) simulated annealing algorithm, (4) Bayesian optimization, (5) genetic algorithm, and (6) particle swarm optimization. Grid search, random research, and Bayesian optimization are the most prevalent hyperparameter optimization methodologies [83].

The grid search is the most basic way. For each possible combination of all hyperparameter settings, a prediction model will be built, and each model will be assessed to see which architecture produces the best results. Random search provides better models than grid search because it searches a larger, less promising configuration space. The following method, also known as the surrogate method, keeps track of previous assessment outcomes that are utilized to form a probabilistic model and converts the hyperparameters to a probability of a score on the objective function that it employs. Because they investigate the best set of hyperparameters to evaluate based on previous trials, it may be able to find a better set of hyperparameters in less time. citesun2021improved.

GA is a meta-heuristic algorithm that is based on the evolutionary concept [84]. It looks for individuals that have the best chance of survival. The abilities of one generation are passed on to the next. The next generation inherits that trait from their parents and matures into better people as a result. The worst of humanity will gradually fade away. This concept will be utilized to optimize classifier hyperparameters. The population, chromosomes, and genes will be programmed to look for space, hyperparameters, and values. The fitness value will calculate and evaluate performance. On chromosomes, selection, cross-over, and mutation will be utilized to create a new generation and assess performance. These steps will be repeated until the best hyperparameters are found. Particle swarm optimization is another evolutionary optimization technique. Particle swarm optimization is less difficult to implement than the Genetic approach. It works by allowing a group of particles to move semi-randomly around the search space [13].

3.1.3 Outlier Removal using IQR

IQR is a data processing method used to remove outliers. By dividing a rank-ordered dataset into four equal portions, or quartiles, it calculates dispersion [85]. The middle values in the first and second halves of the rank-ordered dataset, respectively, are designated by the letters Q1, Q2, and Q3, while the median value for the entire set is denoted by Q2. Then, Q3$-$Q1 is equal to IQR. Here, data instances outside of the normal range (Q1$-$(1.5$*$IQR) or Q3+(1.5$*$IQR) are considered outliers.

3.1.4 Particle Swarm Optimization (PSO)

PSO was developed from the study of bird migration and foraging behavior by Eberhart and Kennedy near the end of the twentieth century [86]. Each member of the group has a unique perceptual capacity, which allows them to recognize the best local and global individual locations and change their next behavior accordingly. Individuals are treated as particles in a multi-dimensional search space in the method, with each particle representing a potential solution to the optimization issue. The particle characteristics are described using three factors: location, velocity, and fitness value. The fitness function determines the fitness value. The particle modifies its traveling direction and distance independently based on the ideal global fitness value, iterative arriving at the best option. we are using velocity and position updates for every iteration based on that it computes the personal best and global best and up to termination condition met or no of iterations. It takes a group of candidate solutions and uses a position-velocity updating approach to try to select the optimal one. Uses a star topology in which each particle is drawn to the best-performing particle. The position update can be defined as:

$$y_i(t+1) = y_i(t) + v_i(t+1) \tag{3.1}$$

where $y_i(t)$ is position value at time t $v_i(t+1$ is velocity at time t+1 The velocity update rule

$$\begin{aligned} v_{ij}(t+1) = w * v_{ij}(t) + c_1 r_{1j}(t) * [y_{ij}(t) - x_{ij}(t)] \\ + c_2 r_{2j}(t) * [\hat{y}_j(t) - x_{ij}(t)] \end{aligned} \tag{3.2}$$

Here, c_1 and c_2 are the cognitive and social parameters respectively. They choose between two options for particle behavior: (1) pursue its own best or (2) follow the swarm's global best position. Overall, this determines whether the swarm is explorative or exploitative. In addition, the swarm's inertia is controlled by the parameter w.

3.1.5 Support Vector Machine (SVM)

SVM is a popular statistical-based supervised machine learning technique. It is used for regression and classification tasks [87]. It was developed in 1995 by Cortes and Vapnik to improve class separation and reduce prediction error. SVM is well known for working with both linear and non-linear data and is highly good at overcoming dimensionality-related problems [88]. It works well with short datasets and high-dimensional feature spaces in particular. SVM divides training samples into distinct classes when dealing with linear data by locating a hyperplane with the greatest margin. Additionally, it establishes the maximum separation between the support vectors or nearest points to the margin edge, and the hyperplane with n-1 dimensions [89]. The mathematical formula for maximizing the margin is represented by equation (1), which signifies the weight vector, the input vector, and the bias [90]. Using some kernel functions and the kernel trick, SVM uses a kernel-based approach to cope with non-linear data, locating the optimum hyperplane to linearly segregate data [91]. The list of Kernel functions that were looked through in this study to identify the best is shown below [89]. The linear kernel function is shown in equation (2), where c is a constant.

3.1.6 K-Nearest Neighbor (K-NN)

KNN is a non-parametric supervised machine learning technique. It was created in the early 1950s and later expanded by Thomas Cover [92]. As it uses the entire dataset to categorize the unlabeled data points by assigning them to the closest class based on the distance measurement, K-NN is regarded as a lazy learner technique. The distances that were looked for in this study to determine the best outcomes are listed below. The formulas for computing the Euclidean distance, Minkowski distance, and Manhattan distance, respectively, are represented by equations (6), (7), and (8), where k stands for the total number of neighbors and p is any real value [93].Euclidean distance: K-NN begins by scouring the whole training dataset in search of (K) neighbors that have the shortest path between the target point and the data points. The new data is then classified using the neighborhood's data points' majority voting results.

3.1.7 Decision Tree (DT)

DT is a popular supervised machine learning approach for both classification and regression problems. Although the concept of a DT has been around since the late 1950s, it only really gained traction in 1986 when Quinlan put up the idea of trees with numerous responses [94]. It is renowned for having a structure like a tree that is simple to understand when visualized as a tree. Leaf nodes and internal nodes make up DT. The leaf nodes denote the resultant class, but the internal nodes signify a test over an attribute and have numerous branches reflecting the test outcome. The best quality features are selected using a hierarchical or statistical approach, and DT is built using a recursive divide-and-conquer strategy [95].

3.1.8 Multi Layer Perceptron (MLP)

MLP is a feed-forward network with gradient descent as a back propagation algorithm. It reduces loss function and maximizes performance. Unlike perceptron, MLP has more than one layer. The input layer just translates the input, whereas the hidden and output layer compute the weighted sum of inputs and their associated weights, plus the bias of that neuron [96]. Bagging is an ensemble approach that was introduced by Breiman in 1996

[6]. It employs a bootstrapping technique to create a diverse subset of training datasets, as it lessens the variance. Then, these subsets are trained in parallel by multiple weak learners. Afterward, the outcome of each learner is aggregated using soft or hard voting, depending on the task type.

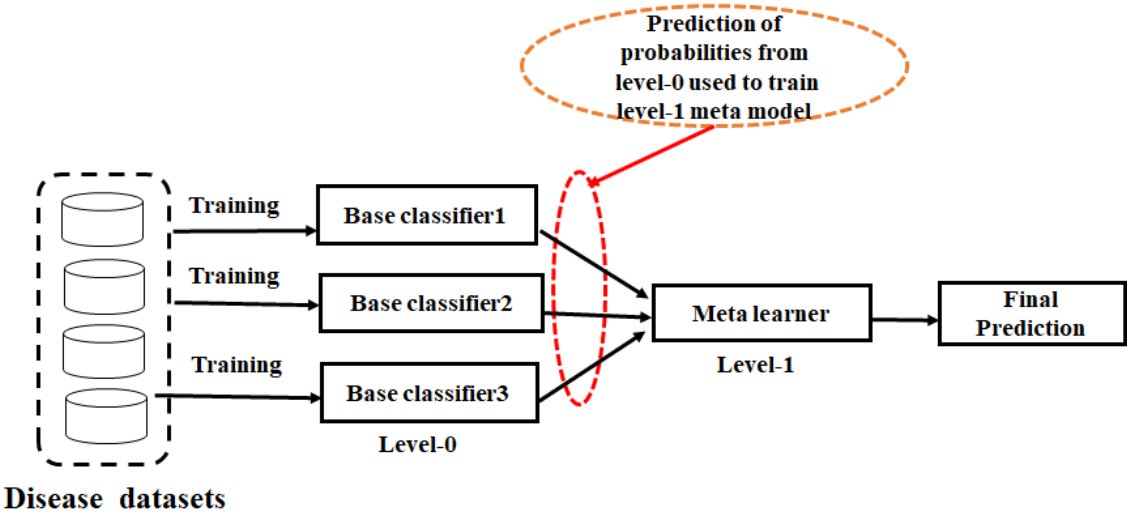

Figure 3.1: Basic stacking approach

3.1.9 Stacking

Stacking is another ensemble framework where a new classifier combines several distinct predictions from base learners to classify the unseen sample. It was first presented by Wolpert in 1992 to completely minimize bias and variance, which increases predictive accuracy [7]. There are two layers in the stacking structure [97]. The first layer consists of many base learners, while the second layer acts like a combiner and meta-learner. Basic stacking in Fig. 3.1 and extended (3-level) stacking in Fig. 3.2 are shown.

3.2 Methodology

In our proposed work, we have

1. performed preprocessing of the dataset, removed outliers and missing values, and scaled the data.
2. model selection using 10-FCV. The proposed hybrid model consists of ADASYN oversampling and a 3-level stacking approach with a PSO-optimized SVM meta-model.
3. designed a three-layer stacking framework with KNN, DT, SVM, MLP, and LR in layer 0, bagged DT, KNN, and LR in layer 1, and optimized SVM in layer 2.
4. optimized SVM hyperparameters using PSO with novel fitness function. Finally, statistical analysis with a paired T-test was performed to test the significance of the proposed model with base-level classifiers.

The proposed model is shown in Fig. 3.3. The 3-level proposed model is described in the section.

Initially, the data set will be pre-processed using IQR. Then, the dataset underwent further experiments to determine the best oversampling technique for class balance.

3.2.1 Architecture of the proposed ensemble

An extended version of the two-layer stacking ensemble has been proposed to investigate whether stacking increases prediction model accuracy. The proposed stacked generalization is made up of three layers: (1) base classification, (2) meta classification 1, and (3) meta classification 2. To obtain the layer 2 meta-models, the proposed stacking classifier

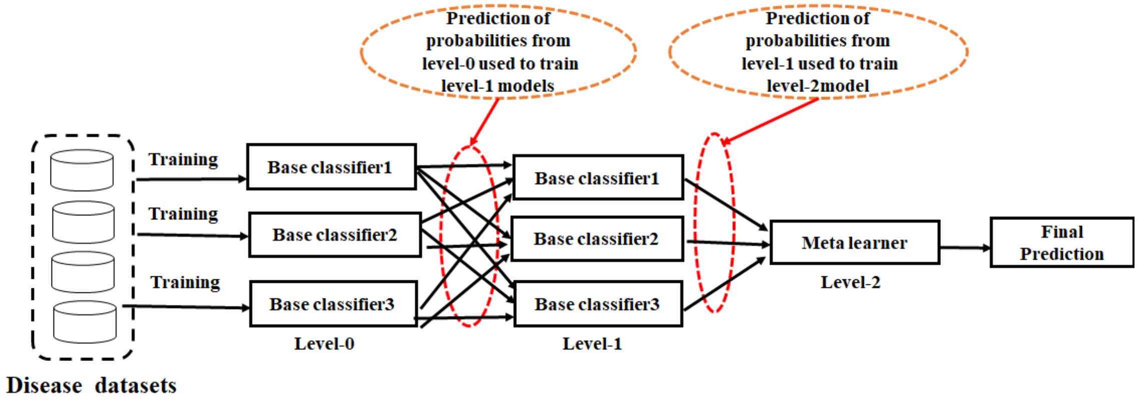

Figure 3.2: 3-level stacking approach

utilized five (5) base classifiers, all of which were trained using three (3) selected meta-classifiers. The three (3) meta-models formed by each meta-classifier were transmitted to the next layer, which produced the final prediction model with a single meta-classifier.

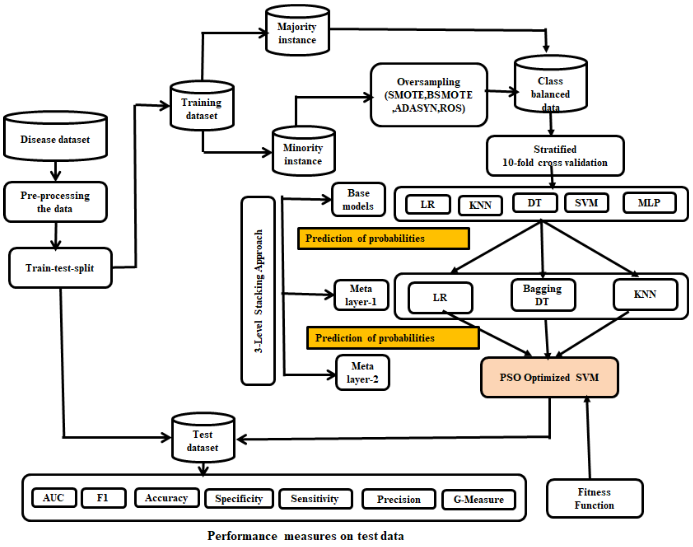

Figure 3.3: Proposed Novel Stacking Approach

In layer 0 base models used extended stacking (3-levels) classifier employs the LR, KNN, DT, SVM, and MLP algorithms. Because these ML models were chosen using

10-FCV. Individual classifiers create prediction models with varying degrees of accuracy. Layer 1's output prediction models were used as layer 2's inputs.

Layer 2 meta-classifiers include LR, KNN, and bagged DT classifiers. The choice of a meta-classifier should be based on the prediction job, and as of this writing, the meta-learners have opted to produce the layer 2 output [98]. SVM was used as the proposed procedure's layer 3 meta-classifier. The selection of distinct algorithms is motivated by the fact that they take fundamentally varied approaches to model generation and focus on data in different ways to make a meaningful contribution to ensemble implementation. On a single dataset S, different learning algorithms L_1, L_2,..., L_N are applied to examples s_k =(x_k, y_k), i.e., pairs of feature vectors (x_k) and their classifications. (y_k). The first layer generates the basis classifiers C_1, C_2,..., and C_N, where C_k = L_k. Meta-level classifiers are trained in the second layer to aggregate the outputs of base-level classifiers.

3.2.2 Stacking Framework

A novel three-level stacking framework is proposed. In the proposed framework there are three levels. A novel stacking framework is introduced, featuring three levels. Within this proposed framework, the terms "layer" and "level" are interchangeably used, conveying a synonymous meaning in the context of this three-level stacking approach.

1. In level 0, LR, KNN, DT, SVM, and MLP classifiers are used.
2. In level 1, bagging DT, KNN, and LR classifiers are used.
3. In level 2, an optimized SVM is used. Here, the SVM parameters are optimized using PSO with a novel fitness function.

The choice of classifiers in each layer was based on a combination of factors, including their individual strengths in capturing different aspects of the data, diversity in learning algorithms, and empirical performance in preliminary experiments. We selected classifiers known for their diverse modeling approaches, such as KNN, LR, MLP, DT, and SVM, to ensure a comprehensive exploration of the feature space. While Naïve Bayes could have been considered, its assumption of feature independence might not align well with the data characteristics. As for the specific numbers (5 in the first layer and 3 in the second), we aimed for a balance between model complexity and computational efficiency, ensuring sufficient diversity without overwhelming computational resources.

3.2.3 Multi-level stacking appraoch

To enhance the performance of the level 2 stacking approach (level-0 base models and level-1 meta models) the number of levels. In our proposed model total of 3 levels (level-0 base models,level-1 meta classifiers,level-2 meta classifiers). In our proposed approach we have selected the best-performing models from a pool of ML algorithms such as LR, KNN, DT, MLP, SVM, NB, and RC. From the pool NB and RC are not selected because the cross-validation score is less. The selected models are considered base models and have undergone for stacking approach. Stacking performance may degrade if we do not do a proper configuration of ensemble classifiers. To avoid overfitting we have used 10-FCV to generate predictions of base models. All base models' probabilistic outcomes and original class labels become auxiliary datasets for training the meta-classifiers of layer 2. In a similar way meta classifiers layer-1 will use 10-FCV and generate probabilistic outcomes here one more auxiliary dataset will be generated and used for the training of the level-2 meta classifier. Here level-1 and level-1 depending on previous layers will predict in similar ways but level-1 and level-2 classifiers are entirely different. Here selected meta classifiers used in level-1 are LR, KNN, and bagging DT. Meta classifiers in level 1 will train with all base classifiers. Meta classifiers in level 2 will train based on the meta classifier's level-1 predictions. so the last level meta classifier is used as SVM.SVM is so efficient non-linear algorithm that can classify samples efficiently. Through evolutionary search, SVM parameters are optimized using particle swarm optimization.PSO is a bio-inspired optimal search algorithm. Unlike other optimization algorithms, it required only an objective function and few hyperparameters compared to GA. It is not dependent on the gradient or any differential form of the objective.

The decision to store the AUC if it's the best during the training phase is based on the distinction between hyper parameter optimization using PSO and best hyper parameters of SVM used for test the model generalization ability. While PSO optimizes hyper parameters to enhance the model's generalization ability on validation data level2, the final assessment of model performance is conducted on completely unseen test data. Therefore, even though the hyper parameters were optimized using PSO, we still need to evaluate the model's performance on the test data to ensure its robustness and generalizability.

3.2.4 SVM hyperparameter tuning using PSO

SVM is used as a binary classifier that is used to determine classes from diseased data. SVM with a kernel function is used to improve classification performance whenever data is not linearly separable. The proposed model uses non-linear SVM with Radial Basis Function (RBF) as kernel function which is given in Eq. 3.3.

$$\begin{aligned} k(y, y_i) &= \frac{exp - ||y - y_i||^2}{2\sigma^2} \\ \gamma &= \frac{1}{2\sigma^2} \end{aligned} \tag{3.3}$$

```
Data: Training dataset $D$, Number of Folds $K$
Result: Optimized three-level stacking ensemble with PSO-optimized SVM
Initialize empty lists $M_0$, $M_1$, $M_2$ to store base models;
Initialize empty list $AUC_{best}$ to store best AUC;
for $i \leftarrow 1$ to $K$ do
    $D_{\text{train}}, D_{\text{val}} \leftarrow$ Split $D$ into training and validation sets for fold $i$;
    Train base models (LR, KNN, SVM, DT, MLP) on $D_{\text{train}}$;
    $P_{\text{base}} \leftarrow$ Predict probabilities on $D_{\text{val}}$ using base models;
    Append $P_{\text{base}}$ to auxiliary dataset $D_{\text{level1}}$;
    Train base models (LR, KNN, Bagging DT) on $D_{\text{level1}}$;
    $P_{\text{level1}} \leftarrow$ Predict probabilities on $D_{\text{val}}$ using level 1 models;
    Append $P_{\text{level1}}$ to auxiliary dataset $D_{\text{level2}}$;
end
Use PSO to optimize hyperparameters $C$ and $\gamma$ for SVM on $D_{\text{level2}}$;
$C_{\text{best}} \leftarrow$ Best optimized $C$ from PSO;
$\gamma_{\text{best}} \leftarrow$ Best optimized $\gamma$ from PSO;
Train SVM with $C_{\text{best}}$ and $\gamma_{\text{best}}$ on $D_{\text{level2}}$;
for $i \leftarrow 1$ to $K$ do
    $D_{\text{test}} \leftarrow$ Test data for fold $i$;
    $P_{\text{base}} \leftarrow$ Predict probabilities on $D_{\text{test}}$ using base models;
    $P_{\text{level1}} \leftarrow$ Predict probabilities on $D_{\text{test}}$ using level 1 models;
    $P_{\text{level2}} \leftarrow$ Predict probabilities on $D_{\text{test}}$ using the level 2 SVM model with $C_{\text{best}}$
      and $\gamma_{\text{best}}$;
    Compute AUC for $P_{\text{level2}}$ and store in $AUC_{best}$ if it's the best;
end
return Best hyperparameters $C_{\text{best}}$ and $\gamma_{\text{best}}$, as well as the trained three-level
  stacking ensemble with PSO-optimized SVM;
```

Algorithm 1: Three-Level Stacking with PSO-Optimized SVM

To get the best hyperplane SVM tries to optimize the objective function which is given in Eq. 3.4.

$$Minimize = J(w, d, \eta) = \frac{1}{2}\|w\|^2 + c\Sigma_{i=1}^{N}\eta_i \tag{3.4}$$

$$subject\ to\ x_i(w^T y_i + d) \geq 1 - \eta_i$$

Where, σ is variance,
$||y - y_i||$ is the L2-norm.

There are two hyperparameters in Eq. 3.3 and Eq. 3.4. To achieve enhanced SVM performance, we have to fine-tune kernel function parameters (γ) as well as a soft margin (c). We are proposing PSO for this purpose as PSO converges very fast and quickly moves from exploration to exploitation than other bio-inspired approaches. The Algorithm 1 will describe how PSO is used for SVM hyperparameters tuning.

The parameters of the Level2 SVM were determined through a systematic approach that involved grid search coupled with cross-validation. In this process, we defined a grid of parameter combinations, including values for hyper parameters such as C and γ. Subsequently, we performed cross-validation on the auxiliary dataset level2 to evaluate the performance of each parameter combination. The parameter combination that yielded the highest performance, as measured by a suitable metric (e.g., AUC), was selected as the optimal choice for training the Level 2 SVM model. While PSO was not utilized in this specific step, the grid search technique allowed us to efficiently explore the parameter space and identify an effective configuration for the SVM model.

3.2.5 Fitness function for PSO

We have proposed a novel fitness function that optimizes SVM hyperparameters. The function is devised for imbalanced data by considering AUC, F1-score, and G-measure. For better SVM performance on imbalanced data, the fitness function needs to be maximized.

$$Fitnessfunction(f) = \arg\max AUC \tag{3.5}$$

3.3 Experiments

This section describes the datasets, noise levels, and evaluation metrics employed to assess the proposed model. The qualitative and quantitative results of the proposed model are then compared with the existing models.

The HP Compaq Intel(R) Core(TM) i7-1065G7 CPU and 8 GB RAM were used in this experiment. All the modules in the proposed methodology and results analysis are carried out using Python and the sklearn library. The HP Compaq Intel(R) Core(TM) i7-1065G7 CPU and 8 GB RAM were used in this experiment. All the modules in the proposed methodology and results analysis are carried out using Python and the sklearn library.

3.3.1 Datasets

Various bench-marked disease data sets are used to evaluate the performance of the proposed model from the UCI repository [99]. Those are

1. Pima Indian Diabetes dataset (PID)

2. Statlog Heart Data (SHD)

3. Chronic Kidney Disease (CKD)

4. Wisconsin Breast cancer (WBC)

and the description of datasets shown in Table. 3.1

S. No	**Data set name**	**#patterns**	**#features**	**#patterns in -ve class**	**#patterns in +ve class**
1	Pima Indian Diabetes (PID)	768	8	500	268
2	Statlog Heart Data (SHD)	270	13	150	120
3	Chronic Kidney Disease(CKD)	400	24	150	250
4	Wisconsin Breast Cancer (WBC)	569	32	357	212

Table 3.1: Datasets used in this study

3.3.2 Data set pre-processing

All the disease datasets are processed before the construction of the proposed ensemble model. In the pre-processing following steps are carried

1. replaced zero values with a median.

2. checked numerical columns, binary columns with 2 values, and columns with more than 2 values.

3. label encoding of binary columns.

4. multi-value columns are duplicated.

5. scaling numerical columns with a standard scalar.

6. dropping original values merging scaled values for numerical columns.

7. outlier removal with IQR

3.3.2.1 Outliers removal with IQR

Finally, IQR is applied to remove the outliers from the disease dataset. IQR is applied with two thresholds namely $Q1 = 0.25$ and $Q3 = 0.90$ where $Q1$ is a threshold used in quartile1 and $Q3$ is a threshold used in quartile3. Data samples whose values are below $Q1$ and above $Q3$ are considered as outliers. Once the outliers are identified these outliers are replaced by $low - limt$ if sample value $< Q1$ else replaced with $up - limt$ if sample value $> Q3$. The $low - limt$ and $up - limt$ are calculated using Eq. 3.6.

$$\begin{aligned} up - limit &= Q3 + 1.5 * IQR \\ low - limit &= Q1 - 1.5 * IQR \end{aligned} \tag{3.6}$$

Where, IQR = Q3-Q1.

3.3.3 Performance Measures

To evaluate the performance of the proposed model various performance measures such as accuracy, sensitivity, specificity, G-measure, Precision, Recall, and F-measure are chosen. These measures are obtained from the confusion matrix which is given in Table 3.2. These measures are defined as follows:

		Predicted	
		Diseased	**Healthy**
Actual	**Diseased**	TP	FN
	Healthy	FP	TN

Table 3.2: Confusion matrix

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.7}$$

$$Specificity = \frac{TN}{TN + FP} \tag{3.8}$$

$$G - measure = \sqrt{specificity{*}sensitivity} \tag{3.9}$$

$$Precision = \frac{TP}{TP + FP} \tag{3.10}$$

$$F - measure = 2 * \frac{(Precision * Recall)}{(Precision + Recall)} \tag{3.11}$$

$$False\ Positive\ Rate(FPR) = \frac{FP}{FP + TN} \tag{3.12}$$

$$True\ Positive\ Rate(TPR) = \frac{TP}{TP + FN} \tag{3.13}$$

Where,

- TP represents the disease-positive class that the classifier has classified as disease-positive.
- TN represents the disease-negative class that the classifier has observed as disease-negative.
- FP represents the disease-negative class that the classifier has categorized as disease-positive and
- FN represents the disease-positive class that the classifier has classified as disease-negative.
- The ROC-AUC is a graph that depicts the relationship between the TPR and FPR, indicating the TPR that we can expect for a certain trade-off with the FPR.
- The Area Under the ROC curve (AUC) score, which means that the resulting score measures the model's ability to properly predict the disease classes.

Dataset	Classifier	Mean AUC	Standard Deviation
PID	LR	0.907	0.015
	KNN	0.925	0.012
	DT	0.852	0.018
	MLP	0.900	0.014
	NB	0.841	0.021
	RC	0.820	0.017
SHD	LR	0.890	0.013
	KNN	0.910	0.016
	DT	0.842	0.017
	MLP	0.882	0.011
	NB	0.834	0.019
	RC	0.821	0.015
CKD	LR	0.921	0.014
	KNN	0.901	0.017
	DT	0.852	0.016
	MLP	0.891	0.013
	NB	0.845	0.020
	RC	0.842	0.015
WBC	LR	0.884	0.012
	KNN	0.879	0.011
	DT	0.868	0.014
	MLP	0.866	0.013
	NB	0.851	0.016
	RC	0.849	0.015

Table 3.3: Model selection with 10-FCV including mean and standard deviation of AUC

S.no	**Hyper parameter search space**
1	’hidden_layer_sizes’: [(10,30,10),(20,)], ’activation’: [’tanh’, ’relu’], ’solver’: [’sgd’, ’adam’], ’alpha’: [0.0001, 0.05], ’learning_rate’: [’constant’,’adaptive’]

Table 3.4: Hyper parameter search space used for fine tune MLP using grid search

Dataset	LR	KNN	SVM	DT	MLP	Bagging DT
PID	C=0.01	#neighbours = 11	C =50 gamma =0.001 kernel = RBF	SC = Gini Depth =6	activation=tanh, Alpha = 0.005 Hidden layer sizes: 10, 30, 10	#est =500
SHD	C=0.01	#neighbours = 5	C =100 gamma =0.001 kernel = RBF	SC = Gini Depth =5	activation=tanh, Alpha = 0.005 Hidden layer sizes: 20,	#est =1000
CKD	C=0.001	#neighbours = 13	C =100 gamma =0.001 kernel = RBF	SC = Gini Depth =5	activation=tanh, Alpha = 0.005 Hidden layer sizes: 10, 30, 10	#est =100
WBC	C=0.001	#neighbours = 11	C =50 gamma =0.01 kernel = RBF	SC = Gini Depth =6	activation=tanh, Alpha = 0.005 Hidden layer sizes: 20,	#est =500
C	Regularization Parameter	gamma	RBF kernel coefficient	Alpha	Learning rate	
SC	Splitting Criteria	Depth	Maximum depth of DT			
RBF	Radial Basis Function	est	no of estimators			

Table 3.5: Optimized hyperparameters values of selected classifiers

Dataset	**Classifier**	**Without sampling**	**With sampling**			
			SMOTE	**BSMOTE**	**ADASYN**	**ROS**
PID	KNN	81.23	84.56	85.23	**85.88**	82.96
	SVM	83.21	85.96	83.28	**86.59**	82.20
	LR	78.60	81.23	**82.59**	82.23	79.10
	DT	84.60	85.23	84.50	**85.98**	80.58
	MLP	85.10	**86.10**	82.21	85.23	81.23
SHD	KNN	82.23	83.56	84.23	**84.88**	82.96
	SVM	84.21	86.96	83.28	**87.59**	83.20
	LR	79.60	83.23	**85.59**	83.23	76.10
	DT	83.60	84.23	85.50	**86.98**	84.58
	MLP	84.10	85.10	83.21	**85.23**	81.23
CKD	KNN	85.23	86.56	84.23	**87.88**	81.96
	SVM	84.21	83.90	84.38	**87.70**	84.20
	LR	79.60	83.23	**84.59**	78.23	76.10
	DT	86.60	87.23	85.50	**88.98**	83.58
	MLP	86.10	**87.10**	84.21	84.23	80.23
WBC	KNN	85.23	88.56	**89.23**	84.88	86.96
	SVM	85.21	86.96	84.28	**87.59**	83.20
	LR	79.60	79.23	**84.59**	83.23	81.10
	DT	85.60	82.23	83.50	**86.98**	82.58
	MLP	86.10	87.10	83.21	**88.23**	83.23

Table 3.6: Performance comparison of various oversampling techniques over disease datasets w.r.t AUC

3.4 Experimental Results

In this section, a comprehensive analysis of the results obtained during the evolutionary process is presented.

Dataset	**C1**	**C2**	**w**	γ	**C**
PID	1.4962	1.4962	0.72984	1.8190	3.38
SHD	1.4962	1.4962	0.72984	8.562	3.245
CKD	1.4962	1.4962	0.72984	5.623	1.235
WBC	1.4962	1.4962	0.72984	8.562	4.256
Swarm size(N_p)	20				
iterations(T)	100				
C1	Cognitive constant				
C2	Social constant				
ω	Inertia weight				
γ	kernel parameter				
C	Penalty parameter				

Table 3.7: SVM parameter tuning using PSO

Dataset	Meta Layer	**Classifier**	**Accuracy (%)**	**AUC (%)**	**Sensitivity (%)**	**Specificity (%)**	**F1-Measure (%)**	**Precision (%)**	**G-Measure (%)**
PID	layer-1	LR	79.69	79.65	69.70	92.50	75.89	76.90	75.50
	layer-1	KNN	83.86	84.99	74.50	91.35	76.47	79.40	73.50
	layer-1	bagging DT	**87.80**	**90.68**	79.68	**89.25**	**85.10**	**88.63**	**87.50**
	layer-2	SVM	**89.80**	**93.54**	74.07	**94.00**	**85.10**	**92.73**	**86.06**
SHD	layer-1	LR	85.46	84.80	83.61	87.23	83.65	85.52	83.61
	layer-1	KNN	88.23	85.62	78.51	89.65	**87.23**	**86.32**	**98.31**
	layer-1	bagging DT	**88.24**	**92.54**	**83.65**	90.21	91.62	91.52	89.58
	layer-2	SVM	**91.54**	**92.03**	82.60	**91.68**	**87.65**	**93.57**	**88.58**
CKD	layer-1	LR	83.54	84.67	79.67	89.50	79.89	76.50	85.61
	layer-1	KNN	74.67	78.30	68.72	89.23	76.58	77.32	79.50
	layer-1	bagging DT	**88.65**	**91.52**	79.54	**91.32**	**92.58**	**92.62**	**93.67**
	layer-2	SVM	**94.05**	**95.62**	84.76	**93.54**	**87.65**	**93.54**	**88.98**
WBC	layer-1	LR	86.78	87.60	82.63	89.52	83.65	83.58	84.67
	layer-1	KNN	91.58	89.64	78.67	**92.56**	87.65	**96.78**	87.37
	layer-1	bagging DT	91.65	89.56	76.72	**73.52**	86.54	**95.78**	88.52
	layer-2	SVM	**97.08**	**96.50**	94.29	**98.76**	**96.22**	**96.73**	**89.60**

Table 3.8: Performance of with meta classifiers in layer-1 and layer-2 on various data sets

Further AUC is a proper measure when the dataset is imbalanced. ROC-AUC is a graph showing the performance of a classification model at all classification thresholds. It

Dataset	Classifier	Accuracy	AUC	Sensitivity	F1-score	Precision	time(sec)
PID	LR	74.02	86.40	81.48	68.75	59.25	0.14
	KNN	85.06	88.30	88.88	80.67	73.84	0.18
	DT	85.71	88.90	90.74	81.66	74.24	0.22
	MLP	82.46	87.38	77.77	75.67	73.68	24.36
	SVM	87.66	93.10	**92.59**	84.03	76.92	0.29
	Bagging DT	85.06	90.80	73.84	80.67	73.84	128.37
	Stacking(Level-1 with LR as meta model)	85.71	90.90	83.33	80.35	77.58	436.54
	Stacking(Level-1 with KNN as meta model)	85.71	90.90	83.33	80.35	77.58	523.15
	Stacking(Level-1 with Bagging DT as meta model)	86.36	92.80	85.18	81.41	77.96	456.32
	Stacking(Level-2 SVM)	87.69	92.56	77.77	75.67	73.68	513.25
	Stacking(Level-2 with with PSO Optimized SVM)	**89.80**	**93.54**	74.07	**85.10**	**92.73**	528.45
SHD	LR	76.02	84.40	83.48	78.75	65.25	0.25
	KNN	86.06	86.30	84.88	83.67	78.84	0.14
	DT	84.71	86.90	**89.74**	83.66	79.24	0.16
	MLP	85.46	88.38	78.77	77.67	82.68	22.35
	SVM	86.66	90.10	88.59	83.03	79.92	0.27
	Bagging DT	87.06	91.80	75.84	84.67	76.84	32.56
	Stacking(Level-1 with LR as meta model)	82.71	89.90	86.33	83.35	79.58	412.23
	Stacking(Level-1 with KNN as meta model)	84.71	86.90	88.33	84.35	83.58	524.23
	Stacking(Level-1 with Bagging DT as meta model)	88.36	91.80	86.18	84.41	78.96	465.32
	Stacking(Level-2 SVM)	88.69	90.56	76.77	74.67	75.68	472.363
	Stacking(Level-2 with with PSO Optimized SVM)	**91.54**	**92.03**	82.60	**87.65**	**93.57**	521.36
CKD	LR	76.02	87.40	84.48	76.75	65.25	0.12
	KNN	84.06	86.30	87.88	82.67	78.84	0.16
	DT	84.71	89.90	89.74	85.66	76.24	0.18
	MLP	84.46	86.38	82.77	78.67	77.68	34.62
	SVM	89.66	90.10	**92.59**	85.03	79.92	0.25
	Bagging DT	88.06	92.80	85.84	84.67	84.84	38.24
	Stacking(Level-1 with LR as meta model)	86.71	92.90	85.33	84.35	82.58	421.23
	Stacking(Level-1 with KNN as meta model)	87.71	84.90	86.33	84.35	84.58	435.26
	Stacking(Level-1 with Bagging DT as meta model)	88.36	91.80	84.18	83.41	82.96	475.21
	Stacking(Level-2 SVM)	89.69	91.56	84.77	**88.67**	88.32	485.26
	Stacking(Level-2 with with PSO Optimized SVM)	**94.05**	**95.62**	84.76	87.65	**93.54**	523.21
WBC	LR	79.02	87.40	86.48	78.75	79.25	0.14
	KNN	87.06	86.30	89.88	89.67	78.84	0.18
	DT	87.71	86.90	89.74	85.66	77.24	0.14
	MLP	84.46	88.38	76.77	84.67	85.68	34.65
	SVM	89.66	91.10	94.59	88.03	87.92	0.23
	Bagging DT	89.06	93.80	88.84	86.67	88.84	0.28
	Stacking(Level-1 with LR as meta-model)	87.71	92.90	85.33	86.35	78.58	436.87
	Stacking(Level-1 with KNN as meta-model)	86.71	92.90	86.33	87.35	83.58	485.22
	Stacking(Level-1 with Bagging DT as meta-model)	88.36	90.80	87.18	813.41	86.96	483.32
	Stacking(Level-2 SVM)	88.69	91.56	82.77	84.67	84.68	476.23
	Stacking(Level-2 with with PSO Optimized SVM)	**97.08**	**96.50**	**94.29**	**96.22**	**96.73**	501.66

Table 3.9: Comparison of the proposed model with individual models

Dataset	Accuracy(%)	AUC(%)	Sensitivity(%)	Specificity(%)	F-measure (%)
PID	89.80	93.54	74.07%	94.00	85.10
SHD	91.54	92.03	82.60	91.68	87.65
CKD	94.05	95.62	84.76	93.54	87.65
WBC	97.08	96.50	94.29	98.76	96.22

Table 3.10: Proposed model performance on PID, SHD, CKD, and WBC datasets

Dataset	**Classifiers**	**Accuracy (%)**	**AUC (%)**	**Sensitivity (%)**	**Specificity (%)**	**Yr. Ref.**
PID	Stacking(LR)	76.10	83.80	87.10	55.90	2019 [64]
	Adaboost (DS)	75.00	81.00	84.90	56.60	2019 [64]
	Bagging (4.5)	75.40	82.50	85.50	56.50	2019 [64]
	Adaboost (C4.5)	72.50	78.00	80.40	57.80	2019 [64]
	Bagging (L-SVM)	76.40	81.30	88.90	54.10	2019 [64]
	Bagging (RBF-SVM)	68.10	73.40	86.70	33.30	2019 [64]
	Majority Voting(MV)	76.20	72.10	88.70	53.20	2019 [64]
	Bagging (Poly-SVM)	76.20	81.10	88.20	53.90	2019 [64]
	Stacking(NSGA-II)	83.80	85.90	**96.10**	79.10	2019 [64]
	Bagging (REP)	75.80	83.20	83.70	61.10	2019 [64]
	Random Subspace Method (RSM)	75.30	82.70	86.90	54.20	2019 [64]
	Random Forest	76.30	83.90	84.60	60.30	2019 [64]
	Stacking	68.80	66.50	74.20	58.70	2019 [64]
	Dia-Net	90.87	-	95.74	83.15	2020 [100]
	soft-voting	80.90	79.08	70.69	78.40	2021 [67]
	AdaBoost	74.98	75.32	68.25	60.13	2021 [67]
	Bagging	70.11	74.89	68.75	-	2021 [67]
	GradientBoost	71.89	75.32	48.75	-	2021 [67]
	XGBoost	69.01	75.75	67.50	-	2021 [67]
	CatBoost	74.56	75.32	65.00	-	2021 [67]
	Proposed Approach	**89.80**	**93.54**	74.07	**94.00**	**This study**
SHD	Stacking ensemble	92.34	92.28	93.49	91.07	2022 [9]
	Random Forest	90.21	89.97	**95.12**	84.82	2022 [9]
	Extra Tree Classifier	90.93	90.45	94.30	86.60	2022 [9]
	XGB	91.91	91.79	94.30	89.28	2022 [9]
	Adaboost	83.40	83.14	88.61	77.67	2022 [9]
	GBM	84.25	83.96	90.24	77.67	2022 [9]
	Proposed Approach	**91.54**	**92.03**	82.60	**91.68**	**This study**
CKD	Extra Tree Classifier	94.00	-	**96.00**	91.00	2021 [72]
	Random Tree	91.43	**96.10**	94.00	-	2021 [72]
	Proposed Approach	**94.05**	95.62	84.76	**93.54**	**This study**
WBC	RF	96.00	96.00	**95.00**	96.00	2021 [101]
	Xgboost	97.00	97.00	**95.00**	**99.00**	2021 [101]
	Gradient Boosting	93.00	**98.00**	93.00	94.00	2021 [101]
	Proposed Approach	**97.08**	96.50	94.29	98.76	**This study**

Table 3.11: Comparison between SOTA ensemble models and proposed model on various datasets

Dataset	**Classifier**	**Accuracy (%)**	**AUC (%)**	**Sensitivity (%)**	**Precision (%)**	**F1 Score (%)**	**Yr. Ref.**
PID	SM rule miner	89.87	-	**94.60**	-	-	2017 [22]
	RST-BAT miner	85.33	-	92.6	-	-	2018 [24]
	LR	75.1	-	71.0	68.90	69.90	2021 [102]
	DT	66.80	-	71.1	63.0	75.1	2021 [102]
	MLP	77.20	-	52.50	68.2	59.00	2021 [62]
	NB	72.69	-	66.10	75.90	70.70	2021 [103]
	SVM	74.10	74.08	71.20	75.40	73.20	2021 [103]
	KNN	71.92	66.31	61.25	58.33	59.75	2021 [67]
	DT	85.98	85.11	-	82.12	**90.32**	2022 [104]
	DCN	86.29	91.20	84.2	81.90	-	2022 [105]
	C4.5	75.10	79.26	82.90	71.60	76.80	2022 [103]
	Proposed Approach	**89.90**	**93.54**	88.65	**89.65**	87.51	**This study**
SHD	LR	84.07	90.10	83.58	85.06	83.80	2023 [66]
	LDA	84.07	90.60	83.58	85.04	83.80	2023 [66]
	SVM	83.70	90.30	83.08	84.92	83.40	2023 [66]
	MLP	84.25	84.00	89.43	82.08	85.60	2022 [9]
	KNN	80.85	80.54	86.99	78.67	82.62	2022 [9]
	CART	84.25	84.12	86.99	83.59	85.25	2022 [9]
	Proposed Approach	**94.56**	**92.22**	**93.56**	**90.65**	**94.89**	**This study**
CKD	LR	71.71	78.40	**98.60**	56.48	71.80	2021 [72]
	KNN	64.39	66.50	96.00	59.01	73.09	2021 [72]
	Proposed Approach	**94.05**	**93.86**	95.13	**92.26**	**94.53**	**This study**
WBC	LR	95.62	-	95.84	**97.19**	96.50	2021 [106]
	SVM	**97.18**	-	95.84	97.18	96.50	2021 [106]
	KNN	92.98	-	91.67	97.06	94.29	2021 [106]
	DT	91.00	-	91.00	91.00	91.00	2021 [106]
	DT	91.00	89.00	88.00	91.00	91.00	2021 [101]
	GNB	94.00	94.00	93.00	94.00	94.00	2021 [101]
	SVM Linear	97.00	**97.00**	91.68	97.00	**97.00**	2021 [101]
	SVM RBF	97.00	96.00	93.00	96.00	96.00	2021 [101]
	Proposed Approach	97.08	95.50	96.29	89.96	96.22	**This study**

Table 3.12: Comparison between SOTA non ensemble models and proposed model

plots TPR on the x-axis and FPR on the y-axis at different classification thresholds.

Dataset	LR vs stack	KNN vs stack	SVM vs stack	DT vs stack	MLP vs stack
PID	0.021	0.010	0.0221	0.002	0.003
SHD	0.002	0.002	0.0393	0.021	0.0045
CKD	0.038	0.0032	0.031	0.028	0.026
WBC	0.024	0.003	0.038	0.012	0.025

Table 3.13: Statistical analysis of the performance of base class and proposed stacking model ($p<0.05$)

3.4.1 Results analysis

The above-pre-processed disease datasets are partitioned into training datasets and test datasets. There are plenty of ML-based classifiers, but not all of them may give better predictive performance, so for selecting classifiers, we have used 10-FCV of LR, KNN, DT, SVM, MLP, NB, and RC.Out of those, the NB and RC classifiers have poor predictive performance and are not selected in most of the datasets. So we have removed the NB and RC classifiers for further processing. Model selection is shown in Table 3.3. Class-imbalanced disease datasets will affect the classifier performance as the training dataset undergoes various oversampling techniques such as SMOTE, BSMOTE, ADASYN, and ROS. This over-sampled training dataset is used to fine tune hyperparameters of various classifiers using grid search on PID, CKD,WBC and SHD datasets. The fine tuned hyperparameters are shown in Table 3.5. Hyperparameter search space also provided is shown in Table 3.14.

The results of various oversampling techniques are shown in Table 3.6 and the best results are highlighted. From the table, it is observed that ADASYN outperforms the majority of the classifiers in the majority of disease datasets in terms of the AUC measure, which is the right measure for imbalanced datasets. Hence, we have considered the

S.no	Hyper parameter search space
1	'hidden_layer_sizes': [(10,30,10),(20,)], 'activation': ['tanh', 'relu'], 'solver': ['sgd', 'adam'], 'alpha': [0.0001, 0.05], 'learning_rate': ['constant','adaptive']

Table 3.14: MLP search space using grid search

ADASYN oversampling technique for further processing.

After applying the ADASYN oversampling technique to the disease dataset, class labels are balanced. This balanced data set is partitioned into 10-fold cross-validation (10-FCV). Next, this balanced dataset is used for training the proposed stacking framework.

The proposed stacking framework consists of three layers. Using 10-FCV in each fold level, one learner is trained with nine folds and validated with the remaining one fold This process will repeat for all base models. Probabilistic predictions of the 10-fold cross-validation along with a true class label will form meta-features in the auxiliary dataset. All the base models LR, KNN, SVM, DT, and MLP in level 1, along with three meta-models LR, KNN, and bagged DT in level 2, were trained using generated meta-features from the auxiliary dataset. Similar to the base models, meta-models will also generate probabilistic predictions using 10-fold cross-validation from a new auxiliary dataset generated in the previous layer. All the probabilistic features, along with the original class label, form a new auxiliary dataset for final meta-model training. Using the new auxiliary dataset, the final meta-model will be trained. Once the meta-model is trained, all the base classifiers will undergo training with the entire training data. Final predictions with text data. The last-level meta-model combines the predictions, and it will give the outcome of diseased or not diseased. Here, level-1 and level-2 parameters are optimized with grid search, and level-3 is optimized with SVM.

The PSO itself has some hyperparameters, and these parameters are chosen per the construction coefficient method discussed in PSO Parameter Selection. Next, this fine-tuned PSO is applied to optimize the SVM parameters with a novel fitness function in Eq. 3.5. The fine-tuned hyperparameters of both SVM and PSO are given in Table. 3.7. The optimization of SVM parameters C and γ using PSO is given in Algorithm 1.

3.4.2 PSO parameter selection

The PSO has a cognitive constant (c1), social constant (c2), inertia weight (ω), swarm size, and maximum iterations for termination control parameters. The c1, c2, and w are fine-tuned as per the construction coefficient method [107]. This method helps to prevent explosions and also helps particles converge to an optimal solution. The following formula and inequalities are used to fine-tune c1, c2, and ω values.

$$\chi = \frac{2K}{|2 - \phi - \sqrt{\phi^2 - 4\phi}|} \tag{3.14}$$

such that $0 \leq K \leq 1$

$$\phi = \phi_1 + \phi_2 > 4$$

ω = χ, $c_1 = \chi\phi_1$, $c_2 = \chi\phi_2$

By using this method in our proposed work and from Equation Eq. 3.14 we have fine-tuned K=1, $\Phi_1 = 2.05$, and $\phi_2 = 2.05$. And remaining parameters' swarm size is 20 and the max iteration is 100.

3.4.3 Comparative analysis

Proposed model is compared with meta models in layer 2 and layer 3, and results are shown in Table 3.8, and the best values are highlighted.

Our proposed model, which leverages PSO (Particle Swarm Optimization) to optimize SVM (Support Vector Machine), demonstrates superior performance compared to individual models and meta-models. Specifically, our approach outperforms other models in terms of accuracy, robustness, and overall effectiveness. The integration of PSO optimization with SVM contributes to enhanced predictive capabilities, making our proposed model a compelling choice for addressing complex tasks and yielding superior results when compared to individual classifiers and also other meta models our proposed PSO optimized SVM in level-2 is giving best results comparing with others and results are shown in Table 3.9 and best results are highlighted.

The table shows that the proposed model performs better in terms of accuracy, AUC, F-measure, and precision. The proposed model overall results w.r.t disease datasets is shown and the best results are highlighted in Table 3.10.

Further, the proposed model is compared with the State-Of-The-Art (SOTA) ensemble models in the literature. These results are shown in Table 3.11 and the best results are highlighted. The table shows that the proposed model performs better than other SOTA ensemble models in terms of Accuracy, AUC, and Specificity.

Further, the proposed model is compared with the SOTA non-ensemble models in the literature. These results are shown in Table 3.12 and the best results are highlighted. The table shows that the proposed model performs better than other SOTA non-ensemble models in terms of accuracy, AUC, and precision.

3.4.4 Validating the performance of the proposed ensemble

Using 10-FCV, the statistical significance of the difference between individual base classifiers and the ensemble's final prediction model is evaluated using a paired t-test technique with a significance level of 95%.

By generating a null and alternative hypothesis, the statistical significance of the difference in prediction accuracy between the proposed staking ensemble and the individual algorithms is determined.

- The null hypothesis (H_0) assumed that both models performed equally well.
- Alternative hypothesis (H_1) assumed that the models performed differently.

The following are the hypotheses developed for comparing the proposed stacking ensemble and the LR algorithm: H_0: There is no difference between the proposed stacking ensemble and the LR classifier in terms of performance.

In this manner, the null and alternative hypotheses for all algorithms for whole datasets were created, and they were tested using the Python-supported paired t-test module. Table. 3.13 shows that data sets all had p-values less than 0.05. This suggests that the null hypothesis may be rejected, and statistically convincing evidence has been provided that LR and the proposed stacking ensemble perform differently.

The hypothesis test is repeated for the remaining pairs. The KNN and the proposed stacking model are then selected for the paired t-test. The results reveal that there is a substantial difference between the performance of the KNN algorithm and the novel stack with a 95% confidence level.

When a single dataset does not match the criterion and the other's p-values are less than the significant threshold value, the DT and stack pair work in the same way. As a result, this demonstrates that there is a discernible difference between the selected algorithm pair in terms of prediction accuracy. The p-values for SVM and the suggested stacking ensemble were examined, and all datasets were found to be significant at the 0.05 level. As a result, it is possible to deduce that the SVM and the stacking ensemble perform differently. To begin the t-test, the DT and stacking ensemble are coupled. The null hypothesis was rejected with 95% certainty, implying that these algorithms performed differently in prediction tasks. Finally, the p-value analysis was performed on the last two algorithm pairs. The null hypothesis was rejected with 95% confidence based on the findings of the paired t-test, and the alternative hypothesis was accepted by demonstrating that there is a substantial difference between their performances.

The primary goal of this study is to determine whether there is any utility in adding layers to the proposed stacking ensemble. The significance level of accuracy between the layer 1 output and the layer 2 stack is obviously below the threshold (0.05) for all datasets. This indicates that there is a discernible difference between them, and hence the null hypothesis was rejected.

As a result, it may be stated that there is a difference in their forecast accuracies. The null hypothesis was rejected again, whereas the alternative hypothesis was accepted. The paired t-test significant values were less than the cutoff (0.05). As a result, the null hypothesis was rejected and the alternative hypothesis was accepted due to a significant difference between them.

Finally, the last two pairs were applied to the paired t-test, and the null hypothesis was rejected while the alternative hypothesis was accepted because the significant values for all of the test datasets were less than 0.05. These statistical numbers demonstrate that dividing the stack generalization into three layers can result in significant and obvious accurate prediction results for any machine learning application.

Statistical test is performed on proposed method and other SOTA models with p is less than 0.05. From this test we can say that proposed model significantly differ with other SOTA models.

3.5 Discussion

The following research questions are addressed with the proposed stacking approach.

RQ1. Can we improve predictive performance with an oversampling and ensemble approach?

In the proposed approach used hybrid model with ADASYN oversampling and a stacked ensemble. It gives a significant performance concerning various performance measures such as AUC, F-measure, sensitivity, and specificity balancing with ADASYN, and improving the model performance with 3-level stacking will significantly improve the overall performance of the model.

RQ2. Extended stacking approach(Multi-level) is better in prediction than the basic stacking approach?

In the basic stacking approach base models and one meta-model. In stacking choosing the best configuration of base models as well as meta-models is very crucial otherwise the model will degrade the performance of the individual classifier. The extended stacking approach will always improve performance than basic stacking unless the best configuration and hyperparameters of the classifiers used.

RQ 3. Does the final Meta-model parameter?

Does optimization make any improvement in overall performance?

In 3-level stacking, final meta-model selection and parameter optimization are very important. Many parameter optimization techniques exist, but meta-heuristic optimization such as PSO will optimize efficiently.

RQ 4. How does the proposed model have more significance than other base-level models?

We can evaluate our proposed model performance with the statistical analysis we have done in the statistically paired T-test majority of the classifiers on various datasets significantly differ with p-value (<0.05) with a 95% confidence level.

Carefully choosing base classifiers and parameter optimization with evolutionary algorithms will significantly improve stacking model performance. Large datasets will take a lot of computation time so we need high-power computing resources to deal with multi-level stacking.

Oversampling sample techniques may reduce performance due to noise while generating synthetic data we can be cautious about borderline samples to improve the predictive model performance.

3.5.1 Summary

The summary of this chapter is enhancing disease diagnosis performance through the introduction of a three-level stacking framework. This framework is applied to a pre-processed dataset, which underwent outlier removal using the IQR method and addressed class imbalance using ADASYN. The resultant pre-processed dataset becomes the basis for training the proposed three-level stacking model.

The stacking framework involves three levels: Level 0 learners (including LR, KNN, SVM, DT, KNN, and MLP), Level 1 learners (comprising Bagged DT, KNN, and LR), and a Level 2 learner (SVM). The optimization of these learners is achieved using techniques like grid search for Levels 0 and 1, and PSO for Level 2.

To validate the effectiveness of the proposed model, experiments are conducted on various datasets such as PID, SHD, CHD, CKD, and WBC. The proposed model is compared against different combinations of base learners and consistently shows superior performance across various performance metrics.

Furthermore, a comparison is made between the proposed model and State-of-The-Art (SOTA) ensemble and non-ensemble methods. The proposed model outperforms these models in terms of AUC and accuracy across all datasets. This demonstrates its superiority in diagnostic accuracy.

To establish the robustness of the proposed model, a paired statistical t-test is performed. The results of this test confirm that the proposed model significantly outperforms all base-level models, providing additional evidence of its effectiveness in disease diagnosis.

In summary, the chapter introduces a novel three-level stacking framework for disease diagnosis. It employs optimized Level 0 and Level 1 learners, along with a Level 2 SVM optimized using a fitness function. Experimental results across various datasets show that the proposed model consistently outperforms other models, including SOTA methods, in terms of diagnostic accuracy and AUC. The statistical t-test further validates the significant improvement offered by the proposed model over individual base-level models.

Chapter 4

A Novel Diversity-based Ensemble Approach with Genetic Algorithm for Effective Disease Diagnosis

This chapter introduced a novel diversity-based evolutionary ensemble framework. This framework aims to address the limitations of conventional ensemble methods by focusing on the selection of diverse base classifiers. To achieve this, a GA strategically chooses base classifiers that offer complementary insights into the data. This approach inherently extends the multi-level stacking concept introduced in (Chapter 3), by expanding the notion of model diversity to enhance the ensemble's overall performance.

Chapter Organization: The proposed methodology is presented in section 4.1. The experimental results are provided in section 4.2. Lastly, section 4.3 presents the summary of the work.

4.1 Proposed Method for diversity based ensemble approach

In this section, a novel diversity-based evolutionary ensemble framework with a GA is proposed.

To improve the disease diagnosis performance in this work a novel framework is proposed. The proposed model is shown in Fig. 4.1. In the proposed framework

1. Datasets are pre-processed.
2. Grid search has been used for hyperparameter tuning of individual classifiers.
3. Twenty base learns are created by performing bootstrapping over LR, DT, SVM, and KNN classifiers.
4. GA is applied to finding the optimal ensemble.
5. A novel fitness function is proposed for GA.

After prepossessing of dataset performed hyperparameter tuning of LR, SVM, KNN, and DT using grid search.

The selection of 20 base learners in our proposed diversity-based ensemble approach using GA was not arbitrary but rather based on a deliberate strategy aimed at maximizing diversity and exploring a wide range of classifier combinations. By training each of the five bootstrapped bags on four base learners (LR, DT, SVM, and KNN), we aimed to introduce diversity in the ensemble through the incorporation of classifiers with different modeling approaches and characteristics. The choice of LR, DT, SVM, and KNN as base learners was motivated by their popularity, diverse modeling techniques, and complementary strengths in capturing different aspects of the data. Moreover, these classifiers have been widely used in the literature for disease diagnosis tasks, making them suitable candidates for our ensemble approach. The resulting ensemble of 20 base learners allows for a rich diversity of classifier combinations, which can potentially enhance the ensemble's robustness and generalization ability. By applying GA for the selection of the best ensemble from these base learners, we aim to exploit the diversity inherent in the ensemble candidates and identify the optimal combination of classifiers for disease diagnosis.

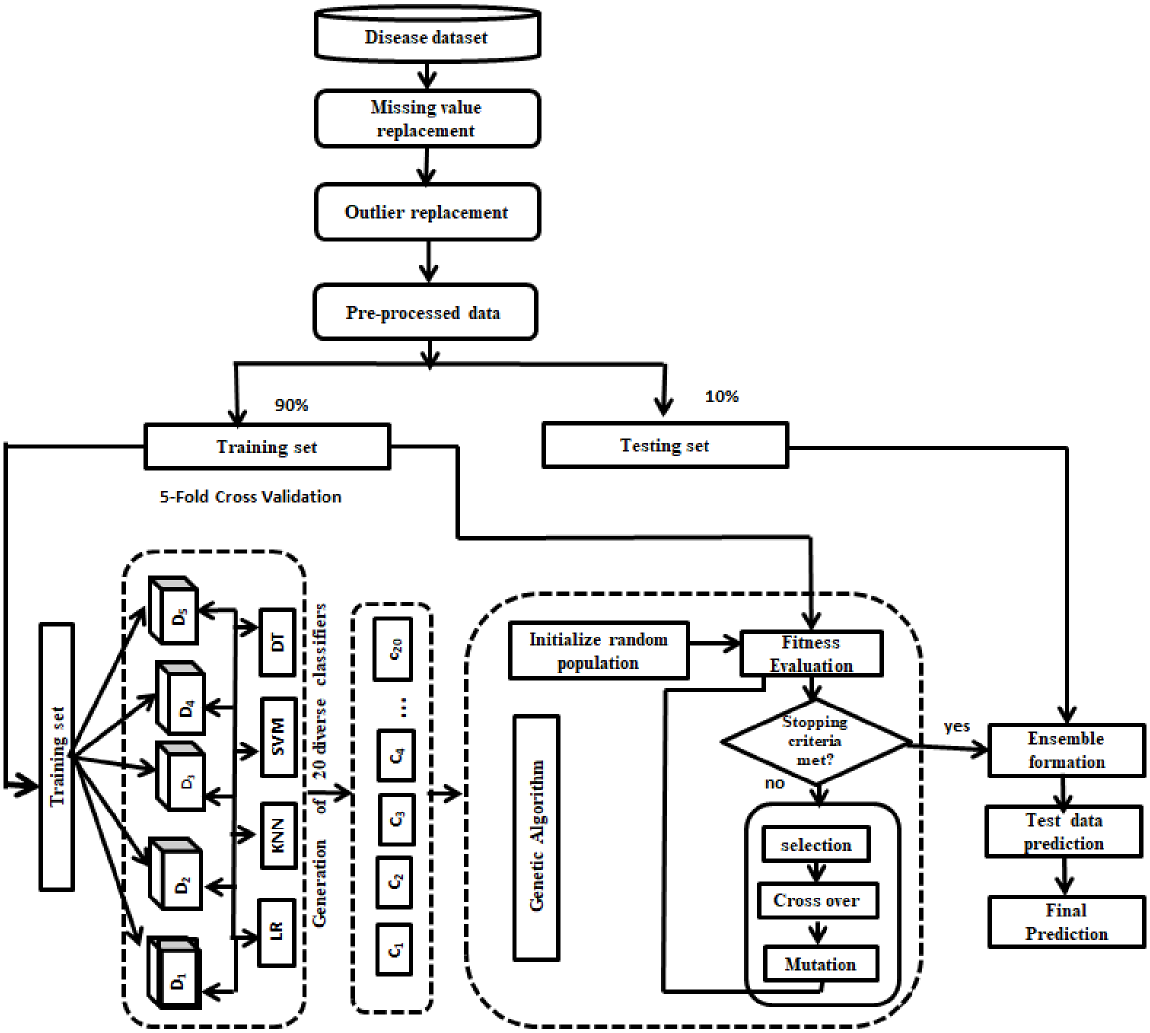

Figure 4.1: Proposed diversity-based ensemble approach using GA

4.1.1 Proposed diversity-based approach using GA

Initially, the data set is pre-processed by handling missing values and outliers. Then such a prepossessed dataset is randomly divided into two disjoint sets in a 90:10 ratio. The 90% data makes a training dataset and is used for hyperparameter tuning of individual classifiers. The remaining 10% of data constitutes a test dataset and is used for testing purposes to estimate the generalization capability of the selected model.

```
Data: Initialize population of candidate solutions
Result: The best chromosome with a maximum fitness value
$p_s$ population size;
$n_f$ number of features;
$c_p$ cross over probability;
$m_p$ mutation probability;
$n_g$ number of generations;
Function_GA($p_s$,$n_f$,$c_p$,$m_p$,$n_g$)
while stop condition is false do
    Compute the fitness of population using Algorithm 4;
    Selection of parents;
    With a crossover probability $p_c$, perform crossover;
    With a mutation probability $p_m$, perform mutation;
    Using crossover rate and mutation rate, generate new solutions;
    If their fitness increases, then accept the new solutions;
    Select the current best for new generations;
    Update new solutions;
end
return the best chromosome with maximum fitness value (used for testing the
proposed model)
```

Algorithm 3: GA function definition

Next, the training data set is partitioned as per 5-fold cross-validation (5-FCV). The 5-FCV generates the five bootstrapped bags, which represent five diverse training datasets. Each of these training datasets is trained on four base learners namely LR, DT, SVM, and KNN. This leads to the creation of 20 base learners. Further, GA is applied for the selection of the best ensemble out of 20 base learners. Moreover, a novel fitness function is proposed for the selection of the best ensemble. How GA is applied for the selection of the best ensemble is discussed in Algorithm 2. The encoding schema of the GA chromosome is shown in Fig. 4.2. Fig 4.1 depicts the proposed model in which the learning process is shown for ensemble candidate selection using the proposed GA.

In algorithm 4, with a chromosome size of 20 and i denoting the chromosome index, the fitness of a proposed ensemble model is computed. Optimal and diverse classifiers are then chosen based on this fitness.The process involves two distinct genetic algorithm (GA) operations. Initially, there is a function call for data bag selection with GA, picking from a set of five available data bags. Subsequently, another GA-based function is invoked for classifier selection, making a choice from a pool of 20 classifiers. Following the sequential selection of data bags and classifiers, the algorithm proceeds to estimate the performance, measured in terms of accuracy, of the chosen classifiers with respect to the selected data bags.

4.1.2 Diversity based classifier selection

5-FCV is used to divide the training set into 5 equal parts. In this case, 5-FCV produces five bootstrapped bags, which stand for five different training sets. On each of these training sets, four base learners are used with hyperparameters that are optimized in respective bootstrapped bags that lead to 20 different models. Here, the validation set is used to assess each chromosome's fitness to respective bootstrapped bags to determine the best ensemble, and test data is utilized to assess the effectiveness of the proposed ensemble model.

Evaluation of an ensemble's diversity, which reveals the disparities between the learners, is not a deciding factor in how effective it is; rather, taught learners should be as

```
Function Function_fitness_score (population) {
    $i \leftarrow 0$;
    while $i < 20$ do
        Function_Classifier_Selection (i);
        Function_Data_Bag_Selection (i);
        Find the ensemble score using Eq.4.1 and Eq.4.2
        Function call for ensemble score calculation
        $i \leftarrow i + 1$;
    end
}
Function Function_Classifier_Selection (i) {
    if $(i \mod 4) == 0$ then
        Select LR;
    end
    if $(i \mod 4) == 1$ then
        Select KNN;
    end
    if $(i \mod 4) == 2$ then
        Select SVM;
    end
    if $(i \mod 4) == 3$ then
        Select DT;
    end
}
Function Function_Data_Bag_Selection (i) {
    if $i \leq 3$ then
        Select bag $D_1$;
    end
    if $4 \leq i \leq 7$ then
        Select bag $D_2$;
    end
    if $8 \leq i \leq 11$ then
        Select bag $D_3$;
    end
    if $12 \leq i \leq 15$ then
        Select bag $D_4$;
    end
    if $16 \leq i \leq 19$ then
        Select bag $D_5$;
    end
}
```

Algorithm 4: Estimation of fitness value

diverse as possible. The various training subsets produced by the 5-FCV result in a variety of base learners. Since diversity is achieved by taking advantage of different learners' biases, selecting the best learners is an important factor in the effectiveness of the ensemble strategy.

4.1.3 Chromosome representation

In this GA is used to select the best candidate ensemble that maximizes the overall performance. In the proposed approach GA chromosome is encoded as a bit string with five groups(bag) with four bits each where each bit is associated with a binary value of 0 or 1. Where binary values 0 and 1 in chromosome represent whether the respective learner is selected or not w.r.t. the learner bag. Therefore, to represent 20 base learners a chromosome of size 20 is used in our proposed approach. A random chromosome is shown in Fig. 4.2 which consists of nine 1's. According to this figure, LR is selected from bag-1, LR, KNN from bag-2, LR from bag-3, LR, DT from bag-4, and LR, SVM, and DT from bag-5 are selected for the ensemble.

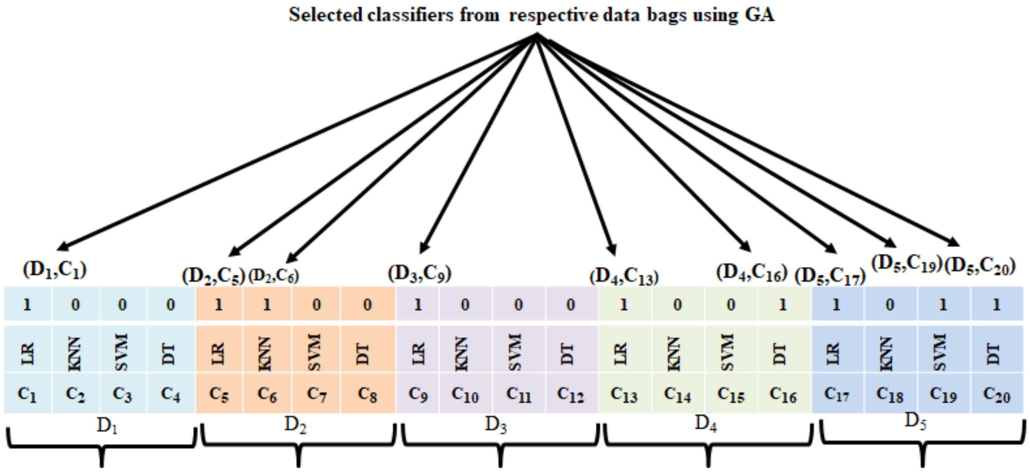
Selected classifiers from respective data bags using GA
(D_1,C_1)
(D_2,C_5) (D_2,C_6)
(D_3,C_9)
(D_4,C_{13})
(D_4,C_{16}) (D_5,C_{17})
(D_5,C_{19}) (D_5,C_{20})
1 0 0 0 1 1 0 0 1 0 0 0 1 0 0 1 1 0 1 1
LR KNN SVM DT LR KNN SVM DT LR KNN SVM DT LR KNN SVM DT LR KNN SVM DT
C_1 C_2 C_3 C_4 C_5 C_6 C_7 C_8 C_9 C_{10} C_{11} C_{12} C_{13} C_{14} C_{15} C_{16} C_{17} C_{18} C_{19} C_{20}
D_1 D_2 D_3 D_4 D_5

Figure 4.2: Structure of the chromosome in the proposed method

The random chromosome of GA From Fig. 4.2 consists of nine classifiers selected w.r.t. various data bags as pairs of data bags and classifiers such as (D_1, C_1), (D_2, C_5), (D_2, C_6), etc are LR Classifier selected concerning data bag-1 (D_1), LR Classifier selected concerning data bag-2 (D_2) and KNN Classifier selected concerning data bag-3(D_3), etc upto 11 random classifiers selected as shown in Fig. 4.2.

4.1.4 GA-based model selection

Diversity is a key concept in ensemble learning, and it refers to the idea that individual model predictions in an ensemble should be as dissimilar as possible. This is because different models are more likely to make different types of errors. To improve the predictive performance of disease diagnosis, it is essential to select an efficient ensemble model from diverse learners. Hence, GA based technique is used to select the best ensemble using the evolutionary search process. Natural selection and genetics are the foundations of GA. It is commonly used to find ideal or nearly ideal solutions to difficult problems that would otherwise take a lifetime to solve. In the GA process, each chromosome acts as an encoded solution in the search space. To search optimal candidate ensemble solution from 20 learners GA is applied with a novel fitness function. This problem is formulated with chromosome size 20 followed by population initialization and evaluating fitness function along with the genetic operators for exploration of search space. The same is explained in the Algorithm 2 and Algorithm 4.

4.1.5 Novel fitness function

In our proposed approach a novel fitness function is used. It is given in Eq. 4.1 and Eq. 4.2

$$Fitness function(f) = \frac{\sum_{j=0}^{b} ES(B_j)}{N_b} \tag{4.1}$$

$$ES(B_i) = MVC\left(\sum_{i=4j}^{4j+3}(B_j, C_i)\right) \tag{4.2}$$

Where,

N_b is the number of active bags,

B_j is the j^{th} selected bag,

C_i is the i^{th} selected classifier,

$ES(B_j)$ is the ensemble score of j^{th} selected data bag,

MVC is the majority voting on all selected C_is in B_j.

Here the fitness function will be computed mean ensemble score of optimal chromosomes in the evolutionary search process. Fitness is evaluated based on the mean of individual data bag ensemble score. The ensemble score of the individual data bag is obtained by applying majority voting on selected classifiers in the respective data bag. For better ensemble selection fitness function needs to be maximized.

4.2 Experimental Results

4.2.1 Experimental setup

The HP Compaq Intel(R) Core(TM) i7-1065G7 CPU and 8 GB RAM were used in this experiment. All the modules in the proposed methodology and results analysis are carried out using Python and the sklearn library. All the datasets used in this study is already described in chapter 3. shown in Table 3.1.

4.2.2 Performance Measures

To evaluate the performance of the proposed model various performance measures such as accuracy, sensitivity, specificity, G-measure, Precision, Recall, and F1-score are chosen. These measures are already explained in chapter 3.

Dataset	Classifier	Accuracy (%)	AUC (%)	Sensitivity (%)	Specificity (%)	F1-Measure (%)	Precision (%)	G-Measure (%)
PID	LR	76.62	70.07	48.14	92.00	60.86	73.68	68.31
	KNN	81.81	78.33	66.66	90.00	72.00	78.26	77.45
	SVM	84.41	82.88	**77.77**	88.00	77.77	77.77	82.73
	DT	**90.90**	**87.03**	74.07	**94.00**	**85.10**	**96.73**	**86.06**
CKD	LR	82.41	82.88	77.77	86.00	77.77	77.77	82.73
	KNN	73.75	77.00	64.00	90.00	75.29	96.07	75.89
	SVM	84.41	83.74	**81.48**	86.00	87.37	84.90	81.24
	DT	**87.01**	**84.48**	77.77	**92.00**	**99.00**	**98.03**	**98.31**
SHD	LR	84.41	83.74	81.48	86.00	87.37	84.90	81.24
	KNN	87.01	84.48	77.77	92.00	**99.00**	**98.03**	**98.31**
	SVM	90.90	87.03	74.07	**94.00**	85.10	96.73	86.06
	DT	**98.00**	**95.66**	**98.00**	93.33	97.02	96.07	95.63
WBC	LR	84.41	82.88	77.77	88.00	77.77	77.77	82.73
	KNN	90.90	87.03	74.07	**94.00**	85.10	**96.73**	86.06
	SVM	**98.00**	**95.66**	**98.00**	93.33	**97.02**	96.07	**95.63**
	DT	90.90	87.03	74.07	**94.00**	85.10	**96.73**	86.06

Table 4.1: Performance of various classifiers on various data sets before applying the proposed model

4.2.3 Results analysis

Initially, the four disease datasets are pre-processed and split into training sets and testing sets with 90% and 10% of data samples respectively. Then we have evaluated the performance of the individual classifier performance before applying in the proposed ensemble model the results are tabulated and it is hown in Table 4.1. To further enhance the individual classifier performance fine-tuned hyperparameters of LR, KNN, SVM, and DT classifiers using grid search. The results are tabulated and shown in Table 4.2.

The further training set is partitioned into 5 equal parts using 5-FCV. The 5-FCV generates 5 bootstrapped bags namely D1, D2, D3, D4, and D5 which represent five diverse data sets. However, diversity is taken into consideration at the time of classifier pool gen-

Dataset	LR	KNN	SVM	DT

PID	C=0.01	#neighbours = 11	C =50
			gamma =0.001
			kernel = RBF
				SC = Gini
				Depth =6

CKD	C=0.001	#neighbours = 13	C =100
			gamma =0.001
			kernel = RBF
				SC = Gini
				Depth =5

SHD	C=0.01	#neighbours = 5	C =100
			gamma =0.001
			kernel = RBF
				SC = Gini
				Depth =5

WBC	C=0.001	#neighbours = 11	C =50
			gamma =0.01
			kernel = RBF
				SC = Gini
				Depth =6

C	Regularization Parameter	gamma	RBF kernel coefficient	
SC	Splitting Criteria	Depth	Maximum depth of DT	
RBF	Radial Basis Function			

Table 4.2: Hyper parameters of various classifiers over various datasets

Dataset	Classifier	D1	D2	D3	D4	D5
PID	LR	C = 1	C = 1	C = 0.1	C = 0.1	C = 0.01
	KNN	# Neighbours = 19	# Neighbours = 7	# Neighbours = 15	# Neighbours = 13	# Neighbours = 11
	SVM	C = 100 Kernel = RBF Gamma = 0.01	C = 10 Kernel = RBF Gamma = 0.01	C = 1 Kernel = RBF Gamma = 0.01	C = 100 Kernel = RBF Gamma = 0.01	C = 50 Kernel = RBF Gamma = 0.001
	DT	SC = GINI Index Depth = 2 Min # samples in leaf = 5	SC = GINI Index Depth = 2 Min # samples in leaf = 5	SC = GINI Index Depth = 2 Min # samples in leaf = 5	SC = GINI Index Depth = 2 Min # samples in leaf = 5	SC = GINI Index Depth = 2 Min # samples in leaf = 5
CKD	LR	C = 10	C = 0.01	C = 100	C = 0.1	C = 100
	KNN	# Neighbours = 13	# Neighbours = 11	# Neighbours = 3	# Neighbours = 5	# Neighbours = 5
	SVM	C = 100 Kernel = RBF Gamma = 0.001	C = 100 Kernel = RBF Gamma = 0.001	C = 100 Kernel = RBF Gamma = 0.001	C = 100 Kernel = RBF Gamma = 0.001	C = 100 Kernel = RBF Gamma = 0.01
	DT	SC = GINI Index Depth = 2 Min # samples in leaf = 5	SC = GINI Index Depth = 2 Min # samples in leaf = 5	SC = GINI Index Depth = 2 Min # samples in leaf = 5	SC = GINI Index Depth = 2 Min # samples in leaf = 10	SC = GINI Index Depth = 10 Min # samples in leaf = 20
SHD	LR	C = 100	C = 100	C = 1.0	C = 100	C = 100
	KNN	# Neighbours = 3	# Neighbours = 13	# Neighbours = 3	# Neighbours = 1	# Neighbours = 3
	SVM	C = 100 Kernel = RBF Gamma = 0.001	C = 100 Kernel = RBF Gamma = 0.01	C = 10 Kernel = RBF Gamma = 0.001	C = 100 Kernel = RBF Gamma = 0.01	C = 100 Kernel = RBF Gamma = 0.01
	DT	SC = GINI Index Depth = 2 Min # samples in leaf = 5	SC = GINI Index Depth = 2 Min # samples in leaf = 5	SC = GINI Index Depth = 2 Min # samples in leaf = 5	SC = GINI Index Depth = 2 Min # samples in leaf = 5	SC = GINI Index Depth = 2 Min # samples in leaf = 5
WBC	LR	C = 0.1	C = 100	C = 0.1	C = 10	C=100
	KNN	# Neighbours = 1	# Neighbours = 9	# Neighbours = 1	# Neighbours = 5	# Neighbours = 7
	SVM	C = 100 Kernel = RBF Gamma = 0.001	C = 10 Kernel = RBF Gamma = 0.001	C = 100 Kernel = RBF Gamma = 0.01	C = 100 Kernel = RBF Gamma = 0.001	C = 100 Kernel = RBF Gamma = 0.01
	DT	SC = GINI Index Depth = 2 Min # samples in leaf = 5	SC = GINI Index Depth = 2 Min # samples in leaf = 5	SC = GINI Index Depth = 2 Min # samples in leaf = 5	SC = GINI Index Depth = 2 Min # samples in leaf = 5	SC = GINI Index Depth = 2 Min # samples in leaf = 5
		C	Regularization Parameter		gamma	RBF kernel coefficient
		SC	Splitting Criteria		Depth	Max depth of DT
		RBF	Radial Basis Function			

Table 4.3: Hyper parameters of various classifiers over data bags of various datasets

eration. Then each of these bootstrapped bags is split into training and validation sets in a 90:10 ratio. These five training datasets are applied to four base learners such as LR, KNN, SVM, and DT, and created twenty diverse base learners. These four base learners are fine-tuned on five diversed data bags using grid search. These fine-tuned hyperparameter values are shown in Table 4.3. These fine-tuned models are tested on the respective validation dataset. These results on four datasets namely PID, CKD, SHD, and WBC are shown in Table 4.4 and and also computed the average performance of all the classifiers with respect to the data bags and compared the performance of individual classifiers without a bootstrapped approach. Performance with respect to data bags is better compared to individual classifier performance. Further, we have applied an ensemble approach using a GA for up to 20 runs(20 experiments) and we have considered the average performance of 20 runs (20 experiments). we have evaluated the same for all four bench-marked datasets.

The selection of 20 runs for the genetic algorithm (GA) in our study was based on several considerations. Firstly, it aligns with common practices in the literature regarding the application of genetic algorithms for optimization tasks. Prior research, such as the work by Singh et al. has demonstrated the effectiveness of using multiple runs of GA to enhance the performance of optimization algorithms for disease diagnosis.

Additionally, we conducted preliminary experiments to assess the convergence behavior and stability of the GA across different numbers of runs. Through these experiments, we observed that 20 runs provided a balance between computational efficiency and result stability. Further increasing the number of runs did not significantly improve the performance metrics while substantially increasing computational costs.

Regarding the sensitivity of the results to this choice, we performed sensitivity analysis by varying the number of GA runs and evaluating the resulting performance metrics. Our findings indicated that while there were minor fluctuations in performance metrics with variations in the number of runs, the overall trends and conclusions remained consistent.

Dataset	**Classifier**	**Data**	**Accuracy (%)**	**AUC (%)**	**Sensitivity (%)**	**Specificity (%)**	**F1 Score (%)**	Precision (%)	G-measure (%)
PID	LR	D1	76.62	73.48	62.96	84.00	65.38	68.00	72.72
		D2	76.62	72.62	59.25	86.00	63.99	69.56	71.38
		D3	66.66	74.33	66.66	82.00	66.66	66.66	73.93
		D4	72.72	67.07	48.14	86.00	55.31	65.00	64.34
		D5	79.22	75.48	62.96	88.00	68.00	73.91	74.43
		Avg	**74.36**	**72.59**	**59.99**	**85.20**	**63.86**	**68.62**	**71.36**
	KNN	D1	92.20	89.74	81.48	98.00	88.00	95.65	89.35
		D2	85.71	83.03	74.07	92.00	78.43	83.33	82.55
		D3	88.31	88.31	81.48	92.00	83.01	84.61	86.58
		D4	80.51	73.07	48.14	98.00	63.41	92.65	68.69
		D5	88.31	84.18	70.37	98.00	80.85	95.00	70.37
		Avg	**87.08**	**83.66**	**71.10**	**95.60**	**78.74**	**90.24**	**79.50**
	SVM	D1	85.71	85.55	85.18	86.00	80.70	76.66	85.59
		D2	80.51	76.48	62.96	90.00	69.38	77.27	75.27
		D3	77.92	72.77	55.55	90.00	63.82	75.00	70.71
		D4	80.51	77.33	66.66	88.00	70.58	75.00	76.59
		D5	79.22	73.77	55.55	92.00	65.21	78.94	71.49
		Avg	**80.77**	**77.18**	**65.18**	**89.20**	**69.93**	**76.57**	**75.93**
	DT	D1	90.90	90.44	88.88	92.00	87.27	85.71	90.43
		D2	89.61	89.44	88.88}	90.00	85.71	82.75	89.44
		D3	92.20	89.74	81.48	98.00	88.00	95.65	89.35
		D4	90.90	88.74	81.48	96.00	86.27	91.66	88.44
		D5	90.90	90.44	88.88	92.00	87.27	85.71	90.43
		Avg	**90.90**	**89.76**	**85.92**	**93.60**	**86.90**	**88.29**	**89.61**
CKD	LR	D1	92.50	90.66	98.00	83.33	94.23	90.74	90.36
		D2	92.50	92.66	92.00	93.33	93.87	9583	92.66
		D3	96.25	95.66	98.00	93.33	97.02	96.07	95.63
		D4	91.25	89.66	96.00	83.33	93.20	90.56	89.44
		D5	72.72	67.07	48.14	86.00	55.31	65.00	64.34
		Avg	**89.04**	**87.14**	**86.42**	**87.86**	**86.72**	**87.64**	**86.48**
	KNN	D1	65.00	62.00	74.00	50.00	72.54	71.15	60.82
		D2	65.00	58.66	84.00	33.33	75.00	67.74	52.91
		D3	71.25	69.00	78.00	60.00	77.22	76.47	68.41
		D4	66.25	64.33	72.00	56.66	72.72	73.46	63.87
		D5	66.25	69.00	58.00	80.00	68.23	82.65	68.11
		Avg	**66.75**	**64.59**	**73.20**	**55.99**	**73.14**	**74.29**	**62.82**
	SVM	D1	71.25	62.33	98.00	26.66	80.99	69.04	51.12
		D2	67.50	58.00	96.00	20.00	78.68	66.66	43.81
		D3	70.00	62.00	94.00	30.00	79.66	69.11	53.10
		D4	63.75	53.00	96.00	10.00	76.79	64.00	30.98
		D5	66.25	59.00	88.00	30.00	76.52	67.89	51.38
		Avg	**67.75**	**58.86**	**94.40**	**23.33**	**78.52**	**67.34**	**46.07**
	DT	D1	91.25	89.66	96.00	83.33	93.20	90.56	89.44
		D2	91.25	89.66	96.00	83.33	93.20	90.56	89.44
		D3	91.25	91.00	92.00	90.00	92.92	93.87	90.99
		D4	92.50	92.66	92.00	93.33	93.87	95.83	92.66
		D5	92.50	92.00	94.00	90.00	95.83	94.00	91.97
		Avg	91.75	90.99	94.00	87.99	93.80	92.96	90.90

Table 4.4: Performance of classifiers over various datasets w.r.t. data bags (Continued)

Dataset	**Classifier**	**Data**	**Accuracy (%)**	**AUC (%)**	**Sensitivity (%)**	**Specificity (%)**	**F1 Score (%)**	Precision (%)	G-measure (%)
SHD	LR	D1	96.29	95.83	91.66	98.99	95.65	96.28	95.74
		D2	92.50	92.66	92.00	93.33	93.87	95.83	92.66
		D3	92.59	92.08	87.50	96.66	91.30	95.45	91.96
		D4	87.03	87.91	95.83	80.00	86.79	79.31	87.55
		D5	94.44	94.58	95.83	93.33	93.87	92.00	94.57
		Avg	**92.57**	**92.61**	**92.56**	**92.46**	**92.29**	**91.77**	**92.49**
	KNN	D1	88.88	95.83	91.66	96.66	86.36	95.00	87.48
		D2	87.03	86.24	79.16	93.33	84.44	90.47	85.95
		D3	90.74	90.00	83.33	96.66	88.88	95.23	89.75
		D4	94.44	94.16	91.66	96.66	93.61	95.65	94.13
		D5	85.18	85.41	87.50	83.33	84.00	80.76	85.39
		Avg	**89.25**	**90.32**	**86.66**	**93.32**	**87.45**	**91.42**	**88.54**
	SVM	D1	94.44	94.16	91.66	96.66	93.61	95.65	94.13
		D2	94.44	94.16	91.66	96.66	93.61	95.65	94.13
		D3	87.03	86.24	79.16	93.33	84.44	90.47	85.95
		D4	90.74	91.25	95.85	86.66	90.19	85.18	91.13
		D5	98.14	98.33	79.16	96.66	97.95	96.00	98.31
		Avg	**92.95**	**92.82**	**87.49**	**93.99**	**91.96**	**92.59**	**92.73**
	DT	D1	92.50	90.66	98.00	83.33	94.23	90.74	90.36
		D2	92.50	92.66	92.00	93.33	93.87	95.83	92.66
		D3	96.25	95.66	98.00	93.33	97.02	96.07	95.63
		D4	91.25	89.66	96.00	83.33	93.20	90.56	89.44
		D5	72.72	67.07	48.14	86.00	55.31	65.00	64.34
		Avg	**89.04**	**87.14**	**86.42**	**87.86**	**86.72**	**87.64**	**86.48**
WBC	LR	D1	94.73	94.84	95.23	94.44	93.02	90.90	94.84
		D2	98.24	98.61	85.71	97.22	92.30	89.67	93.43
		D3	88.88	95.83	91.66	96.66	86.36	95.00	87.48
		D4	87.03	86.24	79.16	93.33	84.44	90.47	85.95
		D5	90.74	90.00	83.33	96.66	88.88	95.23	89.75
		Avg	**91.92**	**93.10**	**87.01**	**95.66**	**89.00**	**92.25**	**90.29**
	KNN	D1	92.98	93.45	95.23	91.66	90.90	86.95	93.43
		D2	94.73	92.85	85.71	97.22	92.30	90.90	92.58
		D3	89.43	87.69	80.95	94.44	85.00	89.47	87.43
		D4	98.24	98.61	85.95	97.22	97.67	95.45	98.60
		D5	94.73	95.23	90.47	97.28	95.00	90.90	94.84
		Avg	**94.02**	**93.56**	**87.66**	**95.56**	**92.17**	**90.73**	**93.37**
	SVM	D1	96.49	96.23	95.23	97.22	95.23	95.23	96.22
		D2	98.24	97.61	95.23	96.20	97.56	96.00	97.59
		D3	96.49	95.23	90.47	93.30	95.00	92.00	95.11
		D4	94.73	92.85	80.95	94.44	85.00	89.47	87.43
		D5	92.48	93.45	95.23	91.66	90.90	86.95	93.43
		Avg	**95.68**	**95.07**	**91.42**	**94.56**	**92.73**	**91.93**	**93.95**
	DT	D1	88.88	95.83	91.66	94.44	93.02	90.00	94.37
		D2	87.03	86.24	79.16	97.22	92.30	89.67	93.43
		D3	90.74	90.00	96.20	94.44	85.00	95.00	87.47
		D4	94.44	94.16	93.30	97.22	97.67	94.36	88.60
		D5	85.18	85.41	90.47	97.28	95.00	96.87	89.23
		Avg	**89.25**	**90.32**	90.15	96.12	92.59	93.18	90.62

Table 4.4: Performance of classifiers over various datasets w.r.t. data bags

Thus, while the choice of 20 runs was somewhat arbitrary, our results demonstrate that the conclusions drawn from the study are robust and not overly sensitive to this specific parameter.

Parameter	Value
Maximum no of iterations	100
Cross over rate	0.90
Population size	100
Mutation rate	0.002
Number of runs	20

Table 4.5: Fine-tuned parameters used in genetic algorithm

Figures 4.3-4.5 correspond to PID dataset but the analysis in the thesis was conducted using a total of four datasets. However, the conclusions drawn from the analysis are applicable to all datasets used in the study.

4.2.4 GA parameter optimization

GA parameters such as cross-over rate, mutation rate, population size, and the number of generations will impact the model's performance. GA parameters will vary from problem to problem. So tuning of parameters is required to improve the performance of the model. In the literature, the range of GA parameters used crossover rate, mutation rate, and population size are (0.6-0.9),(0.001-0.005), and (50-100) respectively. We have fine-tuned the crossover rate of various values 0.6,0.7,0.8 and 0.9 with different values of mutation rate. we have selected the crossover rate of 0.9 and the mutation rate is 0.002 with the best fitness value as shown in Fig.4.3.

we have evaluated the fitness of the model with various values of population size within the range of (50-100) and we are getting the best fitness value with a population size of 100. It is shown in Fig.4.4. We have to run the algorithm up to a certain number of generations for convergence. In our proposed approach we have used up to 100 generations for a single run(single experiments)(experiment). It converged at 80 generations. It is shown in Fig. 4.5.

Next, GA is applied for ensemble selection in which the fitness of the candidate chromosome is evaluated. The size of the chromosome is 20 which represents 20 base learners. This GA searches fittest candidate chromosomes over all generations until convergence of population is reached. Here training dataset is utilized to evaluate the fitness of each chromosome for identifying the optimal ensemble.

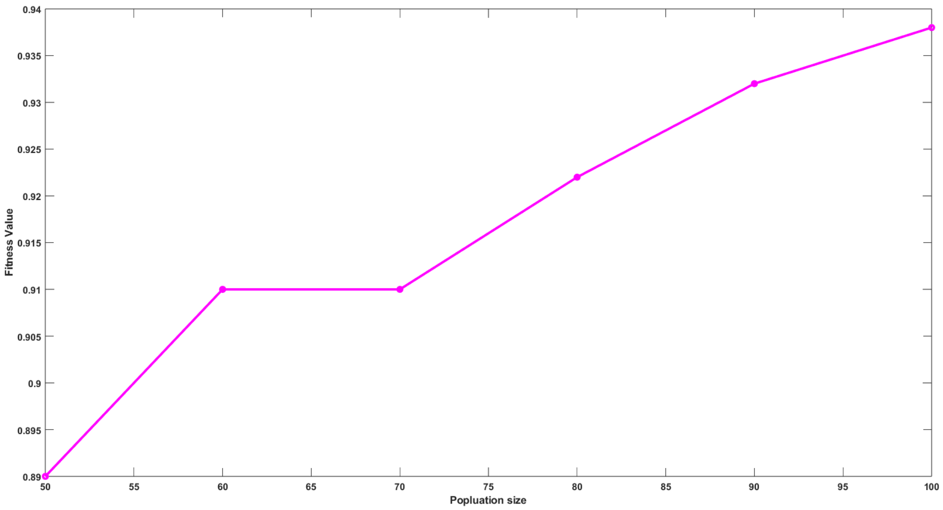

Figure 4.3: Fine tuning of mutation and cross-over rate on PID dataset

For better ensemble selection fitness function needs to be maximized. To find the optimal ensemble the GA parameters are fine-tuned using grid search. These values are shown in Table 4.5. After fine-tuning GA parameters, It experiments for twenty runs on a given dataset.

The further improvement we have applied an ensemble approach using a genetic algorithm for up to 20 runs considered the average performance of 20 runs and evaluated the

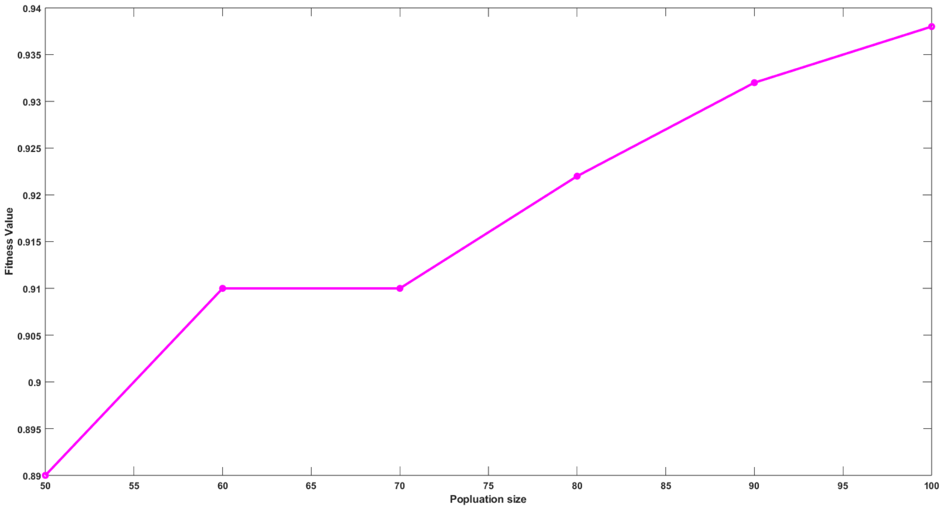

Figure 4.4: Fine tuning of population size on PID dataset

same for all four bench-marked datasets. Each run of GA outputs the best ensemble that maximizes the fitness function and that best chromosome which maximizes the fitness. Summary of all the datasets with performance measures shown in Table**??**

This chromosome it signifies selected diversed classifiers from the five diversed datasets. This fittest chromosome will undergone for testing to evaluate the proposed model performance. These test results for each run of GA on PID, CKD, SHD, and WBC are shown in Table 4.6, Table 4.7, Table 4.8 and Table 4.9 respectively. These tables also present the average performance over 20 runs of GA in terms of ensemble complexity, accuracy, AUC, sensitivity, specificity, and F-measure. Where the ensemble complexity represents the number of classifiers selected out of 20 base learners. The average testing performance of the proposed model on PID, CKD, SHD, and WBC datasets is shown in Table. 4.10

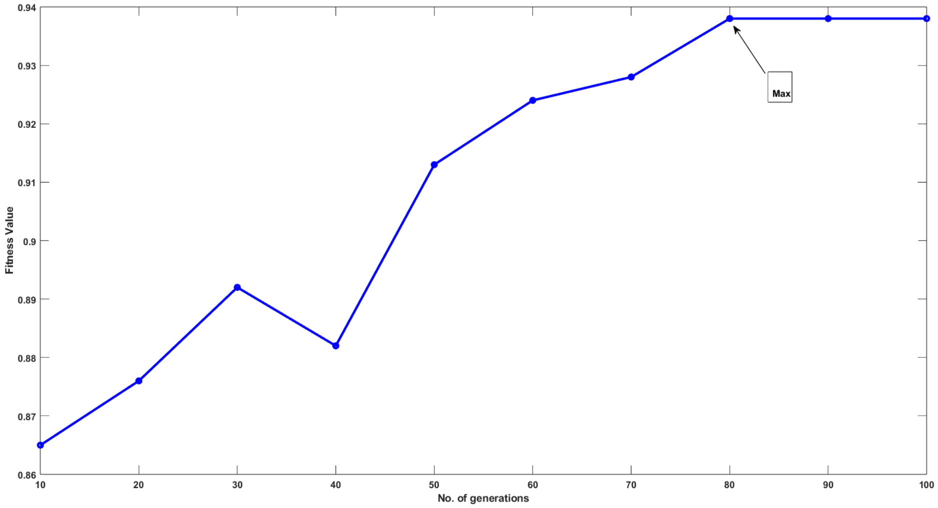

Figure 4.5: Fine tuning of number of generations on PID dataset

Number of runs	**Complexity of ensemble**	**Test results**				
		Accuracy (%)	**AUC (%)**	**Sensitivity (%)**	**Specificity (%)**	**F1 Score (%)**
1	5	90.90	89.51	82.18	94.00	86.97
2	9	92.53	90.41	83.33	97.50	88.48
3	10	91.88	89.04	79.62	98.50	86.68
4	11	93.18	91.12	84.25	98.00	89.5
5	13	92.20	89.74	81.48	98.00	87.65
6	7	91.55	88.81	79.62	98.00	86.65
7	9	87.66	83.25	68.51	98.00	78.84
8	5	88.63	84.64	71.29	98.00	80.48
9	8	90.25	88.24	81.48	95.00	85.17
10	8	91.77	91.11	88.88	93.33	88.75
11	6	89.61	87.52	80.55	94.50	84.01
12	8	88.96	86.60	78.70	94.50	82.89
13	7	89.93	87.77	80.55	95.00	84.42
14	8	92.20	91.23	87.96	84.50	89.11
15	9	93.83	91.81	86.11	98.00	90.62
16	10	91.68	89.00	80.00	98.00	86.45
17	11	90.90	87.71	77.03	98.4	84.08
18	10	90.25	86.74	75.00	98.5	82.55
19	11	90.90	87.71	77.03	98.4	84.08
20	9	92.53	90.41	83.33	97.50	88.48
Min	5	87.66	83.25	68.51	84.5	78.84
Median	9	91.225	88.905	80.55	98	86.55
Max	13	93.83	91.81	88.88	98.5	90.62
Average	**8**	**90.91**	**88.38**	**79.81**	**95.82**	**85.51**

Table 4.6: Test results of proposed model with 20 runs on PID dataset

Number of runs	**Complexity of ensemble**	**Test results**				
		Accuracy (%)	**AUC (%)**	**Sensitivity (%)**	**Specificity (%)**	**F1 Score (%)**
1	14	99.37	99.33	99.50	99.16	99.49
2	13	98.75	98.50	99.50	97.50	99.00
3	12	99.06	99.08	99.00	99.16	99.24
4	11	98.83	98.74	98.41	99.07	98.41
5	10	99.37	99.33	99.50	99.16	99.49
6	10	99.37	99.33	99.50	99.16	99.49
7	7	94.75	93.93	97.19	90.66	95.88
8	5	97.54	97.26	96.19	98.33	96.63
9	8	94.50	93.46	97.60	89.33	95.69
10	9	96.00	95.46	97.60	93.33	96.82
11	13	99.56	99.40	98.80	99.16	99.39
12	10	96.50	95.59	99.20	91.99	97.28
13	9	96.00	95.46	97.60	93.33	96.82
14	8	96.00	95.46	97.60	93.33	96.82
15	7	91.50	91.86	90.39	93.33	92.45
16	6	91.99	91.99	91.99	91.99	92.91
17	7	92.50	92.66	91.99	93.33	93.30
18	9	93.00	93.33	91.99	94.66	93.70
19	10	93.24	93.53	92.40	94.66	93.90
20	9	93.24	93.53	92.40	94.66	93.90
Min	5	91.50	91.86	90.39	89.33	92.45
Median	9	96.00	95.46	97.60	94.66	96.82
Max	14	99.56	99.40	99.50	99.16	99.49
Average	**9.35**	**96.05**	**95.86**	**96.13**	**95.26**	**96.53**

Table 4.7: Test results of proposed model with 20 runs on CKD data set.

Number of runs	**Complexity of ensemble**	**Test results**				
		Accuracy (%)	**AUC (%)**	**Sensitivity (%)**	**Specificity (%)**	**F1 Score (%)**
1	5	98.83	98.74	98.41	99.07	98.41
2	9	99.07	98.95	97.91	100.00	98.91
3	10	97.22	96.87	93.75	100.00	96.64
4	9	96.29	96.00	93.33	98.66	95.49
5	12	97.89	97.14	94.28	100.00	96.80
6	11	99.25	99.16	98.33	100.00	99.13
7	11	97.89	97.14	94.28	100.00	96.80
8	12	97.77	97.50	95.00	100.00	97.31
9	11	95.55	95.00	90.00	100.00	94.45
10	15	99.29	99.04	98.09	100.00	99.02
11	10	96.29	95.83	91.66	100.00	95.23
12	9	95.37	94.79	89.58	100.00	94.15
13	10	96.29	95.83	91.66	100.00	95.32
14	11	97.77	97.50	95.00	100.00	97.14
15	12	97.77	97.50	95.00	100.00	97.14
16	13	97.77	97.50	95.00	100.00	97.14
17	12	97.77	97.50	95.00	100.00	97.14
18	10	97.03	96.66	93.33	100.00	96.27
19	11	99.25	99.16	98.33	100.00	99.13
20	10	97.03	96.66	93.33	100.00	96.27
Min	5	95.37	94.79	89.58	98.66	94.15
Median	11	97.77	97.32	94.64	100	96.97
Max	15	99.29	99.16	98.41	100	99.13
Average	**10.65**	**97.56**	**97.22**	**94.56**	**99.88**	**96.89**

Table 4.8: Test results of the proposed model with 20 runs on SHD data set

Number of runs	**Complexity of ensemble**	**Test results**				
		Accuracy (%)	**AUC (%)**	**Sensitivity (%)**	**Specificity (%)**	**F1 Score (%)**
1	5	98.20	98.16	98.00	99.16	98.52
2	8	99.12	99.05	98.80	99.30	98.80
3	10	97.54	96.86	94.28	99.44	96.34
4	6	98.83	98.74	98.41	99.07	98.41
5	12	97.89	97.14	94.28	100.00	96.80
6	11	97.89	97.14	94.28	100.00	96.80
7	14	97.89	97.14	94.28	100.00	96.80
8	5	97.54	97.26	96.19	98.33	96.63
9	10	93.33	90.95	81.90	100.00	89.63
10	15	99.29	99.04	98.09	100.00	99.02
11	13	99.56	99.40	98.80	100.00	99.39
12	12	98.24	97.61	95.23	100.00	97.40
13	14	99.29	99.04	98.09	100.00	99.02
14	12	96.49	95.23	90.47	100.00	94.70
15	14	99.29	99.04	98.09	100.00	99.02
16	10	99.64	99.52	99.04	100.00	99.51
17	7	99.12	98.80	97.61	100.00	98.75
18	9	97.54	96.66	93.33	100.00	96.29
19	10	97.54	96.66	93.33	100.00	96.29
20	11	97.54	96.66	93.33	100.00	96.29
Min	5	93.33	90.95	81.9	98.33	89.63
Median	10.5	98.045	97.435	95.71	100	97.1
Max	15	99.64	99.52	99.04	100	99.51
Average	**10.4**	**98.08**	**97.50**	**95.29**	**99.76**	**97.22**

Table 4.9: Test results of proposed model with 20 runs on WBC data set

Dataset	Accuracy(%)	AUC(%)	Sensitivity(%)	Specificity(%)	F1 Score(%)
PID	90.91	88.38	79.81%	95.82	85.51
CKD	96.05	95.86	96.13	95.26	96.53
SHD	97.56	97.22	94.56	99.88	96.89
WBC	98.08	97.50	95.29	99.76	97.22

Table 4.10: Proposed model performance on PID, CKD, SHD and WBC datasets over 20 runs

Dataset	**Classifiers**	**Accuracy (%)**	**AUC (%)**	**Sensitivity (%)**	**Specificity (%)**	**Reference**
PID	Stacking(LR)	76.10	83.80	87.10	55.90	2019 [64]
	Adaboost (DS)	75.00	81.00	84.90	56.60	2019 [64]
	Bagging (4.5)	75.40	82.50	85.50	56.50	2019 [64]
	Adaboost (C4.5)	72.50	78.00	80.40	57.80	2019 [64]
	Bagging (L-SVM)	76.40	81.30	88.90	54.10	2019 [64]
	Bagging (RBF-SVM)	68.10	73.40	86.70	33.30	2019 [64]
	Majority Voting(MV)	76.20	72.10	88.70	53.20	2019 [64]
	Bagging (Poly-SVM)	76.20	81.10	88.20	53.90	2019 [64]
	Stacking(NSGA-II)	83.80	85.90	**96.10**	79.10	2019 [64]
	Bagging (REP)	75.80	83.20	83.70	61.10	2019 [64]
	Random Subspace Method (RSM)	75.30	82.70	86.90	54.20	2019 [64]
	Random Forest	76.30	83.90	84.60	60.30	2019 [64]
	Stacking	68.80	66.50	74.20	58.70	2019 [64]
	Dia-Net	90.87	-	95.74	83.15	2020 [100]
	soft-voting	80.90	79.08	70.69	78.40	2021 [67]
	AdaBoost	74.98	75.32	68.25	60.13	2021 [67]
	Bagging	70.11	74.89	68.75	-	2021 [67]
	GradientBoost	71.89	75.32	48.75	-	2021 [67]
	XGBoost	69.01	75.75	67.50	-	2021 [67]
	CatBoost	74.56	75.32	65.00	-	2021 [67]
	Proposed Approach	**90.91**	**88.38**	79.81	**95.82**	**This study**
CKD	Extra Tree Classifier	94.00	-	96.00	91.00	2021 [72]
	Random Tree	91.43	**96.10**	94.00	-	2021 [72]
	Proposed Approach	**96.05**	95.86	**96.13**	**95.26**	**This study**
SHD	Stacking ensemble	92.34	92.28	93.49	91.07	2022 [9]
	Random Forest	90.21	89.97	**95.12**	84.82	2022 [9]
	Extra Tree Classifier	90.93	90.45	94.30	86.60	2022 [9]
	XGB	91.91	91.79	94.30	89.28	2022 [9]
	Adaboost	83.40	83.14	88.61	77.67	2022 [9]
	GBM	84.25	83.96	90.24	77.67	2022 [9]
	Proposed Approach	**97.56**	**97.22**	94.56	**99.88**	**This study**
WBC	RF	96.00	96.00	**95.00**	96.00	2021 [101]
	Xgboost	97.00	97.00	**95.00**	99.00	2021 [101]
	Gradient Boosting	93.00	98.00	93.00	94.00	2021 [101]
	Proposed Approach	**98.08**	**97.50**	95.29	**99.76**	**This study**

Table 4.11: Comparison between SOTA ensemble models and proposed model on various datasets

Dataset	**Classifier**	**Accuracy (%)**	**AUC (%)**	**Sensitivity (%)**	**Precision (%)**	**F1 Score (%)**	**Ref.**
PID	SM rule miner	89.87	-	**94.60**	-	-	2017 [22]
	RST-BAT miner	85.33	-	92.6	-	-	2018 [24]
	LR	75.1	-	71.0	68.90	69.90	2021 [102]
	DT	66.80	-	71.1	63.0	75.1	2021 [102]
	MLP	77.20	-	52.50	68.2	59.00	2021 [62]
	NB	72.69	-	66.10	75.90	70.70	2021 [103]
	SVM	74.10	74.08	71.20	75.40	73.20	2021 [103]
	KNN	71.92	66.31	61.25	58.33	59.75	2021 [67]
	DT	85.98	85.11	-	82.12	**90.32**	2022 [104]
	DCN	86.29	91.20	84.2	81.90	-	2022 [105]
	C4.5	75.10	79.26	82.90	71.60	76.80	2022 [103]
	Proposed Approach	**90.91**	**88.38**	79.81	**95.82**	**85.51**	**This study**
CKD	LR	71.71	78.40	**98.60**	56.48	71.80	2021 [72]
	KNN	64.39	66.50	96.00	59.01	73.09	2021 [72]
	Proposed Approach	**96.05**	**95.86**	96.13	**95.26**	**96.53**	**This study**
SHD	LR	84.07	90.10	83.58	85.06	83.80	2023 [66]
	LDA	84.07	90.60	83.58	85.04	83.80	2023 [66]
	SVM	83.70	90.30	83.08	84.92	83.40	2023 [66]
	MLP	84.25	84.00	89.43	82.08	85.60	2022 [9]
	KNN	80.85	80.54	86.99	78.67	82.62	2022 [9]
	CART	84.25	84.12	86.99	83.59	85.25	2022 [9]
	Proposed Approach	**97.56**	**97.22**	**94.56**	**96.30**	**96.89**	**This study**
WBC	LR	95.62	-	95.84	**97.19**	96.50	2021 [106]
	SVM	97.18	-	95.84	97.18	96.50	2021 [106]
	KNN	92.98	-	91.67	97.06	94.29	2021 [106]
	DT	91.00	-	91.00	91.00	91.00	2021 [106]
	DT	91.00	89.00	88.00	91.00	91.00	2021 [101]
	GNB	94.00	94.00	93.00	94.00	94.00	2021 [101]
	SVM Linear	97.00	97.00	\textbf{98.00}	97.00	**97.00**	2021 [101]
	SVM RBF	97.00	96.00	93.00	96.00	96.00	2021 [101]
	Proposed Approach	**98.08**	**97.50**	95.29	96.30	96.89	**This study**

Table 4.12: Comparison between SOTA models and proposed model on various datasets

4.2.5 Comparative analysis

Our proposed model is compared with SOTA ensemble methods. These results are shown in Table 4.11 and the best results are highlighted. It is observed from table results that the proposed method achieved the best compared to SOTA models in terms of various performance measures such as accuracy, AUC, sensitivity, specificity, F measure and G measure. Finally, the proposed model is compared with SOTA non-ensemble models. These experimental results are shown in Table 4.12 and the best results are highlighted. From the Table the results it is inferred that the proposed model outperformed comparing with non-ensemble models.

4.3 Summary

The study aimed to enhance the performance of disease classification models through a multi-step approach. Initially, various individual classifiers were evaluated across different disease datasets. To improve model performance, bootstrapped aggregation was applied to the training set. Furthermore, individual classifier performance was assessed concerning data bags, which led to the adoption of an ensemble approach using a GA. A novel fitness function was introduced to compute the fitness of ensemble candidates. Four classifiers, namely LR, KNN, SVM, and DT, were utilized with optimized hyperparameters determined by grid search.

In the experimentation phase, a 5-FCV technique was employed on the training dataset, with GA employed as an evolutionary search to identify optimal ensemble configurations comprising 20 learners trained on bootstrapped data. The fitness of each chromosome in GA was evaluated using the validation set within the 5-FCV framework. Model performance was assessed using metrics such as Precision, Recall, Accuracy, AUC, Specificity, and G-measure.

The robustness of the proposed model was assessed across various disease datasets, including PID, CKD, SHD, and WBC. The performance of the proposed model was compared against state-of-the-art ensemble and non-ensemble models. Impressively, the proposed model exhibited superior Precision, Accuracy, and Specificity, suggesting its effectiveness in disease diagnosis. The model's performance was consistently promising across all tested datasets, suggesting its generalizability. In conclusion, the study recommends the adoption of the proposed model for disease diagnosis due to its notable performance improvements over existing approaches.

Chapter 5

Enhancing Disease Diagnosis Accuracy and Diversity through BA-TLBO Optimized Ensemble Learning

This chapter proposed a Bagging Approach with a Teaching-Learning-Based Optimization (BA-TLBO) for ensemble optimization in disease diagnosis. While ensemble learning shows promising results for disease diagnosis, the task of optimizing ensemble configuration to achieve a delicate balance between accuracy and diversity remains challenging. The diversity-based ensemble framework proposed in chapter 4 marks an important step towards addressing this challenge by selecting diverse base classifiers. However, the ensemble optimization process involves further intricacies.

To extend the groundwork laid in chapter 4, we introduced the Bagging Approach with Teaching-Learning-Based Optimization (BA-TLBO) it aims to optimize the ensemble configuration by incorporating a novel fitness function that considers both mean accuracy and mean diversity. By doing so, we not only address the challenge of base classifier selection as in chapter 4 but also introduce a mechanism for evaluating the performance of different ensemble combinations.

Furthermore, the BA-TLBO algorithm introduces dynamic weight updation and bag size adjustment over iterations. This dynamic adaptation helps strike a balance between exploration and exploitation, allowing the optimization process to navigate the search space more effectively. The iterative process of selecting and replacing bags in the ensemble extends the concept of diversification introduced in 4 to achieve a more balanced and optimized ensemble configuration.

Chapter Organization: Section 5.1 provides the Preliminaries. The proposed methodology is presented in Section 5.2. The experimental results are provided in Section 5.3. A summary of the work is described in Section 5.4.

5.1 Preliminaries

In this section, the overview of the preliminaries, including various diversity measures, such as Hamming distance, Entropy, Bhattacharya Distance, and Q -statistics which are relevant to the proposed method is discussed.

5.1.1 Various diversity measures used in the evaluation of proposed model

Classifier combination is thought to be complicated by the diversity of the classifiers on a team. Nevertheless, because there isn't a widely agreed-upon formal definition, assessing diversity is not simple [108]. Kuncheva et al. [109] described the diversity measures and relation with accuracy [110], [111]. Hamming distance is used to measure the dissimilarity between two binary strings and is commonly employed in ensemble methods to assess diversity among the model's predictions. Entropy quantifies the uncertainty in a probability distribution and is used as a diversity measure in ensembles to evaluate the variability of the model's predictions. Bhattacharyya distance calculates the similarity between two probability distributions and is utilized in ensemble learning to gauge the agreement or complementary among individual models. The Q statistic measures the level of agreement among the model's predictions in an ensemble, helping to select diverse models and improve the ensemble's performance by reducing overfitting and capturing a wider range of patterns.

5.1.1.1 Hamming distance

The Hamming distance is a valuable diversity measure widely used in ensemble methods to assess the variability and complementary among individual models [112]. In the context of ensembles, where multiple models are combined to make predictions, diversity is a crucial factor that affects the overall performance. The Hamming distance is calculated by comparing the predictions of each model in the ensemble for a given set of instances. It quantifies the number of positions at which two predictions differ, resulting in a single numerical value that represents the dissimilarity between the two models. High Hamming distance indicates significant diversity, suggesting that the models in the ensemble are making distinct predictions, enhancing the ensemble's ability to capture a wide range of patterns and making it more robust and accurate. By promoting diverse model behaviors, the Hamming distance contributes to reducing overfitting and increasing the generalization power of the ensemble, ultimately leading to improved performance and reliable predictions across different scenarios.

Let M_1 and M_2 be two classifiers with predictions $\mathbf{y}_1$ and $\mathbf{y}_2$, respectively. The Hamming distance between the two classifiers can be computed as follows:

$$\text{Hamming distance} = \frac{1}{N}\sum_{i=1}^{N}\delta(\mathbf{y}_1[i], \mathbf{y}_2[i]) \tag{5.1}$$

where N is the number of instances in the dataset, and $\delta(\mathbf{y}_1[i], \mathbf{y}_2[i])$ is the Kronecker delta function, which equals 1 if $\mathbf{y}_1[i]$ is not equal to $\mathbf{y}_2[i]$ (i.e., there is a difference in predictions for instance i) and 0 otherwise.

5.1.1.2 Entropy

Entropy is a popular diversity measure used in ensemble learning to quantify the uncertainty or variability among the predictions of individual models [113]. In the context of ensembles, which combine multiple models to make collective predictions, entropy serves as an information-theoretic metric that reflects the unpredictability of the ensemble's output. It is calculated by considering the probability distribution of class labels across the ensemble's predictions [114]. A higher entropy value indicates greater diversity, implying that the individual models within the ensemble have dissimilar behaviors, covering a broader range of patterns and making the ensemble more robust. Conversely, lower entropy suggests more agreement among the models, potentially leading to overfitting or reduced generalization. By utilizing entropy as a diversity measure, ensemble algorithms can dynamically adjust the combination of models to strike a balance between exploration and exploitation, effectively enhancing the ensemble's performance and adaptability to various scenarios. Moreover, entropy-based diversity measures provide insights into the ensemble's collective decision-making process, aiding model selection and refinement strategies, and further reinforcing the reliability of the ensemble's predictions in real-world applications [115].

Let $M_1, M_2, \ldots, M_N$ be N classifiers in an ensemble, and let p_{ij} be the probability of classifier M_i predicting class j for a given instance. The entropy of the ensemble can be calculated as:

$$\text{Entropy} = -\sum_{i=1}^{N}\sum_{j=1}^{C} p_{ij} \log p_{ij} \tag{5.2}$$

where C is the number of classes. The entropy measures the uncertainty or information content in the ensemble's predictions. A higher entropy value indicates greater diversity among the classifiers, suggesting that they have different preferences for classifying instances. This diversity can lead to improved ensemble performance by capturing a wider range of patterns and reducing overfitting. On the other hand, a lower entropy value indicates more agreement among the classifiers, potentially leading to reduced ensemble performance. Therefore, entropy serves as a valuable metric for assessing the diversity and adaptability of ensembles in machine-learning applications.

5.1.1.3 Bhattacharya Distance

The Bhattacharyya distance [116] is a meaningful diversity measure frequently employed in ensemble learning to assess the dissimilarity and complementary among individual models within an ensemble. In the context of ensembles, where multiple models are combined to make collective predictions, the Bhattacharyya distance [117] is calculated by measuring the similarity of the probability distributions of class labels across the predictions of each model. A lower Bhattacharyya distance implies greater diversity, suggesting that the individual models in the ensemble have distinct decision boundaries and are specialized in different regions of the feature space. This diversity enhances the ensemble's ability to capture a wide array of patterns, leading to improved generalization and robustness. By promoting diverse model behaviors, the Bhattacharyya distance [118] contributes to minimizing the risk of overfitting and mitigating the impact of individual model weaknesses, thus increasing the ensemble's overall performance. Its incorporation into ensemble approaches allows for effective model selection and combination, enabling the ensemble to produce more accurate and reliable predictions across diverse and challenging real-world scenarios.

Let $M_1, M_2, \ldots, M_N$ be N classifiers in an ensemble, and let p_{ij} and q_{ij} be the probability of classifier M_i and M_j predicting class j for a given instance, respectively. The Bhattacharyya distance between classifiers M_i and M_j can be calculated as:

$$\text{Bhattacharyya distance}(M_i, M_j) = -\log\left(\sum_{j=1}^{C}\sqrt{p_{ij}q_{ij}}\right) \tag{5.3}$$

where C is the number of classes. The Bhattacharyya distance measures the similarity of the probability distributions of class labels across the predictions of two individual classifiers. A lower Bhattacharyya distance value indicates greater diversity between the classifiers, meaning they have distinct decision boundaries and offer complementary information. In ensemble learning, by selecting classifiers with significant Bhattacharyya distance values, we can form a diverse and well-performing ensemble that captures a broader range of patterns and achieves improved generalization compared to using similar models.

5.1.1.4 Q Statistic

The Q statistic [119], also known as the Q index or the inter-rater agreement index, is a valuable diversity measure widely utilized in ensemble learning to assess the level of agreement or disagreement among the predictions of individual models within an ensemble [120] [121]. In the context of ensembles, where multiple models are combined to make collective predictions, the Q statistic is calculated by comparing the predicted class labels of each model for a given set of instances. It represents the proportion of instances for which all models in the ensemble agree on the same prediction. A higher Q statistic indicates greater diversity, implying that the individual models within the ensemble have different perspectives and decision boundaries, leading to a diverse range of predictions. This diversity enhances the ensemble's capacity to capture complex patterns and reduces the risk of overfitting. On the other hand, a lower Q statistic suggests more agreement among the models, which might lead to decreased ensemble diversity and potential performance degradation. By incorporating the Q statistic as a diversity measure, ensemble methods can effectively balance model consensus and variability, resulting in improved ensemble performance and robust predictions across various real-world scenarios. Furthermore, the Q statistic aids in identifying and selecting complementary models, enhancing the ensemble's predictive accuracy and overall reliability. The Q statistic is a diversity measure that quantifies the level of agreement among the predictions of individual models within an ensemble. It is defined as the proportion of instances for which all models in the ensemble agree on the same prediction. The equation for calculating the Q statistic for an ensemble of N classifiers is as follows:

$$Q = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{L} \sum_{j=1}^{L} \delta(\mathbf{y}_i[j], \text{mode}(\mathbf{y}_1[j], \mathbf{y}_2[j], \ldots, \mathbf{y}_N[j])) \tag{5.4}$$

where L is the number of instances in the dataset, $\mathbf{y}_i[j]$ represents the predicted class label of classifier M_i for instance j, and $\delta(a, b)$ is the Kronecker delta function, which equals 1 if a is equal to b, and 0 otherwise. The mode function returns the most frequent class label among the predictions of all classifiers for a given instance. The Q statistic ranges from 0 to 1, with 0 indicating no agreement and 1 indicating complete agreement among the models' predictions. A higher Q value suggests greater diversity in the ensemble, reflecting that the individual models have different decision boundaries and offer complementary information, which can lead to improved ensemble performance and generalization.

5.2 Proposed Methodology

proposed approach using BA-TLBO Optimization is inspired by traditional TLBO proposed by Rao et al. [51]. In this study, TLBO integrates an bagging approach and optimizes the bags for effective disease diagnosis and also strikes the balance between accuracy and diversity. As per best knowledge, it is the first time using the TLBO-based bag optimization and ensemble integration for effective disease diagnosis and it balances the accuracy and diversity simultaneously. BA-TLBO is used for bagging optimization and the proposed model is shown in Fig. 5.1. In previous studies, most bag's content and size are static and it creates a fixed number of bags with fixed instances, often randomly sampled from the training data. This lack of adaptability means that the ensemble's composition remains constant regardless of changes in the data distribution or the performance of individual models. In dynamic bagging, bags are updated iteratively, allowing the ensemble to adapt to changes and potentially improve over time. In static bagging, certain instances in the dataset may be left out or underrepresented in the bags, resulting in the underutilization of potentially informative data points. Dynamic bagging can actively incorporate different instances into bags during the optimization process, ensuring a more comprehensive coverage of the data. Static bagging might lead to an ensemble composed of similar or redundant base models. This limits the diversity of predictions and might result in sub-optimal performance. Dynamic bagging, on the other hand, can optimize the composition of the ensemble by replacing under performing models with more effective ones. The above-mentioned reasons introduced a dynamic bagging approach and also it introduced adaptivity.

The proposed approach followed the steps

1. **Data Splitting**: The input dataset is divided into training and test sets using a standard approach. The training set will be used for ensemble optimization, while the

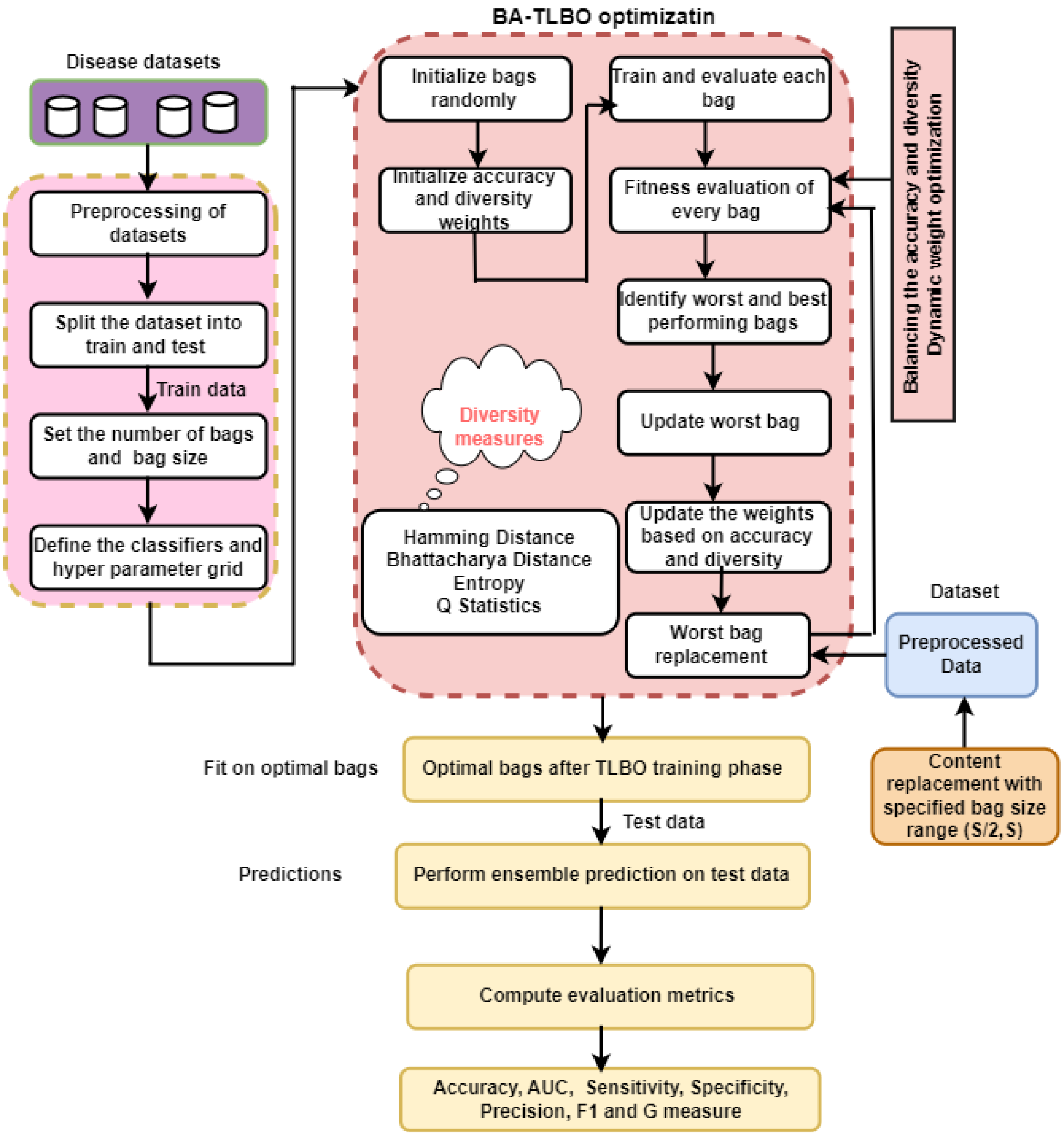
BA-TLBO optimizatin
Disease datasets
Preprocessing of datasets
Split the dataset into train and test
Train data
Set the number of bags and bag size
Define the classifiers and hyper parameter grid
Initialize bags randomly
Initialize accuracy and diversity weights
Train and evaluate each bag
Fitness evaluation of every bag
Identify worst and best performing bags
Update worst bag
Update the weights based on accuracy and diversity
Worst bag replacement
Diversity measures
Hamming Distance
Bhattacharya Distance
Entropy
Q Statistics
Balancing the accuracy and diversity
Dynamic weight optimization
Dataset
Preprocessed Data
Content replacement with specified bag size range (S/2, S)
Fit on optimal bags
Optimal bags after TLBO training phase
Test data
Predictions
Perform ensemble prediction on test data
Compute evaluation metrics
Accuracy, AUC, Sensitivity, Specificity, Precision, F1 and G measure

Figure 5.1: Proposed Model

test set will be used for evaluating the performance of the optimized ensemble.

2. **Ensemble Construction**: The proposed methodology aims to construct an optimized ensemble of classifiers by leveraging the BA-TLBO algorithm that facilitates the selection and updating of bags of classifiers to create an ensemble with high accuracy and diversity.

3. **Bag Initialization**: The optimization process starts by initializing a set of bags randomly. Each bag represents a potential ensemble configuration, and the number of bags is predefined in our approach using 5 bags and 10 bags. The size of each bag is randomly determined within a specified range, ensuring variability in the ensemble.

4. **BA-TLBO Optimization**: The TLBO algorithm is employed to iteratively optimize the ensemble by selecting and updating bags of classifiers. The optimization process consists of the following steps:

   (a) **Training and Evaluation**: For each bag in the optimization process, the classifiers within the bag are trained using the instances in the training set. The performance of each bag is evaluated using 5-fold cross-validation with classifiers such as LR, SVM, DT, and KNN using accuracy and diversity metrics.

   (b) **Disagreement Calculation**: The predicted labels of the classifiers within each bag are used to calculate the disagreement matrix. This matrix represents the diversity among the classifiers in a bag and is typically based on a distance metric such as Hamming distance.

   (c) **Fitness Calculation**: The fitness of each bag is computed by considering both accuracy and diversity. A higher fitness value indicates a better-performing bag. The fitness is typically calculated with the fitness function which is dynamic weights of accuracy and diversity.

(d) **Replacement of the Worst Bag**: The bag with the worst performance (lowest fitness) is identified, and it is replaced with a new bag randomly sampled from the dataset. The size of the new bag is randomly determined within the specified range. This replacement operation promotes exploration and convergence towards better ensemble configurations.

(e) **Iterative Optimization**: The above steps are repeated for a predefined number of iterations, allowing the TLBO algorithm to explore and refine the ensemble configuration. The iterative optimization process aims to identify bags with high accuracy and diversity, leading to an optimized ensemble.

5. **Optimal Ensemble**: After the TLBO optimization process, the algorithm returns the set of optimized bags, representing the final ensemble configuration. Each bag contains the indices of instances in the training set corresponding to the selected classifiers for that bag.

6. **Ensemble Predictions**: The optimized bags are ensemble configurations used for prediction on test data. The predictions of each classifier for the test dataset are stored in an ensemble prediction matrix.

7. **Majority Voting**: The ensemble predictions are obtained by performing majority voting. For each instance in the test dataset, the class label with the most votes among the classifiers in the ensemble is selected as the final prediction.

8. **Evaluation Metrics**: The performance of the optimized ensemble is evaluated using various evaluation metrics, including accuracy, area under the ROC curve (AUC), precision, recall (sensitivity), and F1-measure. These metrics provide a comprehensive assessment of the ensemble's predictive capabilities

The proposed methodology leverages the BA-TLBO algorithm to construct an optimized ensemble of classifiers with high accuracy and diversity. By combining the strengths of individual classifiers, the ensemble can improve prediction performance and capture different aspects of the underlying data distribution. The experimental results and evaluation metrics demonstrate the effectiveness of the proposed approach in constructing a high-performing ensemble.

The term "bags" in the context of our methodology may have been inadvertently conflated with the concept of ensemble classifiers, leading to ambiguity. To clarify, in the context of our algorithm, "bags" refer specifically to collections of randomly selected samples from the dataset, rather than ensembles of classifiers. Each bag represents a subset of the training data, and the optimization process aims to iteratively select and update these bags to construct an optimized ensemble of classifiers. The provided algorithms, particularly Algorithm 5, outline the process of TLBO optimization for bagging classifiers, where the focus is on selecting and updating bags of data samples rather than classifiers themselves. This optimization process involves iteratively evaluating the fitness of each bag based on its accuracy and diversity, as calculated using Algorithm 6. Subsequently, in the teaching and learning phases of the TLBO algorithm (as depicted in Algorithm 5), bags are replaced or updated based on their performance relative to others in the ensemble. To further emphasize this distinction and clarify any confusion, we will ensure that the terminology used throughout the methodology accurately reflects the role of bags as collections of data samples rather than classifiers themselves.

5.2.1 Novel Fitness Function

A novel fitness function is applied in our proposed technique. It is given in Eq. 5.5

$$F = \alpha * B_{ma} + \beta * B_{md} \tag{5.5}$$

Where,

B_{ma} represents the mean accuracies of bag B_{md} represents the mean diversities of bag computed pairwise dissimilarity measure using the hamming distance between pairwise classifiers. α and β controlling parameters of accuracies and diversity. Here α + β =1. F is the fitness function it will compute based on mean accuracies and diversities of bags.

The fitness function calculates the overall fitness value F by summing the mean accuracies and diversities with suitable α and β. By maximizing this fitness function, aim to maximize the accuracy and diversity of every individual bag. The fitness function is designed to evaluate the performance of bags and identify the best and worst bags in every iteration of TLBO. To calculate the ensemble score, majority voting is applied to the diverse classifiers within each data bag. The goal is to maximize the fitness value, indicating optimal bag selection.

5.2.2 Dynamic weight updation

Initially, accuracy weight and diversity weights are assigned to 0.5 to give equal preference in computing fitness over the iterations the weights will update dynamically. The dynamic weight update aims to find a balance between accuracy and diversity that is conducive to the optimal performance of the ensemble in disease diagnosis. The procedure of dynamic weight updation is explained in Algorithm 7.

When accuracy is deemed more important the algorithm increases the weight assigned

```
Input  : Classifiers,$num\_of\_bags$, $min\_bag\_size$, $max\_bag\_size$
Output: optimal_bags: Optimized bags (ensembles)

Initialization: Generate random bags $B$ with size $N$ and bag size within the
  range $[min\_bag\_size, max\_bag\_size]$;
TLBO Optimization: for iteration in range($T$) do
    Evaluate the fitness of each bag in $B$ using algorithm 2;
    Sort bags in ascending order of fitness;
    Teaching Phase: for each bag $b$ in $B$ do
        if $b$ is the worst bag then
            Generate a new bag $b'$ by randomly sampling a subset from the dataset
              with size within the range $[min\_bag\_size, max\_bag\_size]$;
            Replace the worst bag $b$ with the new bag $b'$;
        end
    end
    Learning Phase: for each bag $b$ in $B$ do
        if $b$ is not the best bag then
            Generate a new bag $b'$ by randomly sampling a subset from the dataset
              with size within the range $[min\_bag\_size, max\_bag\_size]$;
            Replace $b$ with $b'$ with a certain probability;
        end
    end
end
Return the optimal bags: return $B$;
```

Algorithm 5: TLBO Optimization for Bagging Classifiers

```
Input  : accuracies, diversities,accuracy_weight, diversity_weight
Output: fitness
Function CalculateFitness(accuracies, diversities, accuracy_weight,
 diversity_weight):
    /* Initialize variables                                    */
    $fitness \leftarrow 0$;
    $n \leftarrow$ length of accuracies;
    $mean\_accuracy \leftarrow \frac{1}{n}\sum_{i=1}^{n}$ accuracies$[i]$;
    $mean\_diversity \leftarrow \frac{1}{n}\sum_{i=1}^{n}$ diversities$[i]$;
    /* Calculate fitness value                                 */
    $fitness \leftarrow$
     $accuracy\_weight \times mean\_accuracy + diversity\_weight \times mean\_diversity$;
     weight optimization with Algorithm 3
    return fitness
```

Algorithm 6: Fitness Function Computation

```
Data: List of accuracies, list of diversities, best bag index , worst bag index,
      accuracy weight , diversity weight $diversity\_weight$
if accuracies[worst_bag_index] < accuracies[best_bag_index] then
    $accuracy\_weight \leftarrow accuracy\_weight - 0.1$;
end
else
    $accuracy\_weight \leftarrow accuracy\_weight + 0.1$;
end
if diversities[worst_bag_index] < diversities[best_bag_index] then
    $diversity\_weight \leftarrow diversity\_weight - 0.1$;
end
else
    $diversity\_weight \leftarrow diversity\_weight + 0.1$;
end
return $accuracy\_weight, diversity\_weight$;
```

Algorithm 7: Dynamic Weight Optimization

to accuracy and reduces the weight assigned to diversity. This adjustment indicates that the ensemble is currently focusing on improving its accuracy to more precise predictions. When diversity is considered more important the algorithm increases the weight assigned to diversity and decreases the weight assigned to accuracy. This suggests that the ensemble is prioritizing a wider range of predictions to ensure robustness and coverage across various scenarios. Overall, these dynamic weight updates help the ensemble algorithm adapt its optimization strategy based on the relative importance of accuracy and diversity at different points during the optimization process. This adaptability enables the ensemble to strike an effective balance between accurate predictions and diverse perspectives, ultimately leading to improved performance in disease diagnosis.

To evaluate the effectiveness of the proposed model several p measures of performance were selected, including G-measure, precision, recall, and F-measure, accuracy, sensitivity, and specificity. These measures are discussed already in chapter 3.

5.2.3 Replacement of worst bag

Replace the worst bag (with index `worst_bag_index`) with a new bag generated by random sampling from the dataset. if optimization with respect to the best bag we need to replace the best bag (with index `best_bag_index`) with a new bag generated by random sampling from the dataset.

Classifier	Search space
LR	’C’: [0.1,0.01, 1.0, 10,100]
KNN	’n_neighbors’: [3, 5, 7,9,11,13]
SVM	’C’: [0.1, 1, 10,50,100], ’kernel’: [’linear’, ’rbf’]
DT	’max_depth’: [None, 5, 10]

Table 5.1: Search Space for hyperparameter optimization on PID, SHD, WBC, and SLC

Dataset	LR	KNN	SVM	DT

PID	C=0.01	#neighbours = 11	C =50
			gamma =0.001
			kernel = RBF
				SC = Gini
				Depth =6

SLC	C=0.001	#neighbours = 13	C =100
			gamma =0.001
			kernel = RBF
				SC = Gini
				Depth =5

SHD	C=0.01	#neighbours = 5	C =100
			gamma =0.001
			kernel = RBF
				SC = Gini
				Depth =5

WBC	C=0.001	#neighbours = 11	C =50
			gamma =0.01
			kernel = RBF
				SC = Gini
				Depth =6

C	Regularization Parameter	gamma	RBF kernel coefficient	
SC	Splitting Criteria	Depth	Maximum depth of DT	
RBF	Radial Basis Function			

Table 5.2: Hyper parameters of various classifiers over various datasets

5.3 Experimental Results

The proposed experimental setup uses Python and the Anaconda distribution to implement the BA-TLBO algorithm for ensemble classifier selection. The Anaconda environment provides a convenient platform for managing the required packages and dependencies.

The result analysis for the PID, SHD, SLC, and WBC datasets will depend on the specific implementation and experimental setup. However, provide a general outline of how you can analyze the results obtained from these datasets.

5.3.1 Results analysis

After preprocessing the disease datasets were split into training and testing sets with 80% and 20% respectively. 80% training dataset is used for training the proposed model and 20% is unseen data only used for final testing of the proposed model.

To further improve the accuracy and maintain the diversity, proposed BA-TLBO-based optimization in ensemble learning. The proposed approach is inspired by traditional TLBO and adopted in the ensemble optimization process and base models hyperparameters are optimized with a grid search approach using search space shown in Table 5.1. The fine tuned hyperparameters using grid search are shown in Table 5.2. The experimental process involves data splitting into training and test sets, defining the classifiers and their hyperparameter grids, implementing the BA-TLBO optimization to obtain the optimized bags, making ensemble predictions using the optimized bags, and evaluating the ensemble's performance using various metrics such as accuracy, AUC, precision, recall, and F1-measure. This study used four benchmark disease datasets such as PID, SLC, SHD, and WBC datasets.

Further individual classifiers performance is evaluated using 5-FCV these results are shown in Table 5.3. All these classifiers are evaluated in terms of Accuracy, AUC, Sensitivity, and Specificity and best results are highlighted.

The proposed model is described in Algorithm 5. In the proposed approach training data is split into bags (5 bags and 10 bags) and used dynamic bag size within the range of (S/2, S) where S is the training data size. The advantages of using dynamic bag size are enhanced diversity, adaptability, overfitting mitigation, exploration and exploitation balance, efficient resource utilization, and reduced sensitivity to hyperparameters. Once splitting into varied-size bags all those bags underwent for BA-TLBO optimization process in the ensemble. All the bags will optimize iteratively up to a certain number of iterations or termination criteria reached. The accuracy of the bag is computed using Five-Fold cross-validation (5-FCV). Once the classifiers predict the labels then pairwise diversity as the hamming distance is used in the proposed model. Mean accuracy and diversity are used in the optimization process. Fitness evaluation is described in Algorithm 6. Fitness evaluation is performed using Eq. 5.5.

The weights such as Accuracy weight and Diversity weights initialized at 0.5 each to

Dataset	Classifier	Accuracy	AUC	Sensitivity	Specificity
PID	LR	0.778	0.736	0.597	0.876
	KNN	0.811	0.793	0.735	0.852
	SVM	0.837	0.814	0.738	0.890
	DT	**0.877**	0.867	0.832	0.902
WBC	LR	0.945	0.976	0.962	0.991
	KNN	0.964	0.958	0.934	0.983
	SVM	**0.973**	0.970	0.957	0.983
	DT	0.917	0.916	0.910	0.921
SHD	LR	**0.863**	0.8567	0.800	0.880
	KNN	0.833	0.8300	0.0800	0.8750
	SVM	0.8370	0.8325	0.791	0.8375
	DT	0.7370	0.7292	0.658	0.883
SLC	LR	**0.928**	0.783	0.977	0.589
	KNN	0.906	0.759	0.955	0.564
	SVM	0.919	0.800	0.959	0.641
	DT	0.873	0.730	0.922	0.538

Table 5.3: 5-FCV Performance of various classifiers

give equal preference to both accuracy and diversity. During the optimization process, the weights update dynamically based on the accuracy and diversity of the previous iterations. If accuracy is higher than diversity then accuracy weight increased 10% and diversity decreased 10% and vice versa. Dynamic weight optimization is described in Algorithm 7 These dynamic weights can help the model to balance between exploration and exploitation. In every iteration the fitness of all bags is computed based on fitness function after that all fitness values are sorted and identify low fitness value index will represent a poor-performing bag in the ensemble. The proposed study optimized the worst bag. The idea behind optimizing the worst bag in each iteration is to focus on improving the weakest component of the ensemble. By continually updating the worst-performing bag, the algorithm aims to gradually enhance the overall ensemble's performance by addressing its weakest. The advantage is a progressive improvement, diversity maintenance, adaptation, enhanced robustness, and exploration-exploitation balance. For a better understanding purpose analyzed the results with best bag optimization and observed the best bag optimization such as rapid convergence, limited exploration, over fitting risk, lack of adaptability, dependence on initial conditions, and neglecting weaker components.

In the proposed approach diversity measure is used as Hamming distance in ensemble learning, quantifies the dissimilarity or differences between the predictions of individual base learners (classifiers) within an ensemble. It measures how often two classifiers produce different predictions for the same instance.

The proposed model is evaluated on four benchmark disease datasets such as PID, SHD, WBC, and SLC. A wide variety of experiments was performed using the proposed model with 5 bags and 10 bags as well as worst bag optimization (Proposed) and best bag optimization for analysis purposes. In addition to Hamming distance as a diversity measure analyzed other diversity measures in the literature such as Entropy, Bhattacharya distance, and Q statistics diversity measures are computed using Eq. 5.1, Eq. 5.2, Eq. 5.3 and Eq. 5.4. And also computed accuracy, diversity, fitness, accuracy weight, and diversity weight all these optimized results over the 100 iterations. The results of these diversity measures are tabulated every 20 iterations and observed BA-TLBO convergence. And also observed with these results how BA-TLBO reached global optimal over the iterations and also observed worst bag and best bag optimizations with 5 bags and 10 bags and with all diversity measures. Results are shown in Table 5.4 from these tables, BA-TLBO reaches global optimal with worst bag and gradual convergence and it focuses on weak component and other side best bag quick convergence and most of the cases overfitting. So from these analyses worst bag optimization is more robust and gradual increase of fitness over the iterations. similar way analyzed PID, SHD, WBC, and SLC disease datasets.

Finally, test results are shown in Table 5.5. The test results showed effective performance with the hamming distance measure compared to other diversity measures. In most cases, bag size 5 is optimal performance than bag size 10. The table observed that the proposed model with the worst bag optimization gave a superior performance on PID with 10 bags and hamming distance as diversity measure giving the highest accuracy as 0.945, SHD with 5 bags giving the highest accuracy of 0.944, WBC with 5 bags giving highest accuracy is 0.982 and finally, SLC with 5 bags giving highest accuracy which is 0.951. In the proposed approach with dynamic bag size proposed model gives adaptive results and robustness. Dynamic weight optimization improves significantly and also balances the accuracy and diversity with optimized ensemble configuration for better predictions on test data. And also analyzed best bag optimization performance with quick convergence and overfitting conditions arise due to focusing more on the best fitness bag and ignoring the weak bag.

Iter	**OB**	**NB**	**DM**	**Accuracy**	**Diversity**	**Fitness**	**AW**	**DW**
PID	Worst bag	5	Hamming	0.974	0.178	0.975	0.763	0.237
		10	Hamming	0.973	0.113	0.971	0.763	0.237
		5	Entropy	0.927	0.924	6.171	0.991	0.009
		10	Entropy	0.982	0.935	6.19	0.909	0.091
		5	Bhattacharya	0.921	0.935	0.936	0.775	0.225
		10	Bhattacharya	0.976	0.124	0.983	0.993	0.007
		5	Q Statistics	0.921	0.965	0.930	0.851	0.149
		10	Q Statistics	0.915	0.924	0.942	0.915	0.085
	Best bag	5	Hamming	0.983	0.092	0.971	0.919	0.081
		10	Hamming	0.969	0.163	0.971	0.882	0.118
		5	Entropy	0.943	0.923	0.930	0.992	0.008
		10	Entropy	0.935	0.933	0.912	0.786	0.214
		5	Bhattacharya	0.976	0.124	0.983	0.835	0.165

Iter	OB	NB	DM	Accuracy	Diversity	Fitness	AW	DW
		10	Bhattacharya	0.976	0.086	0.977	0.934	0.066
		5	Q Statistics	0.910	0.925	0.930	0.763	0.237
		10	Q Statistics	0.916	0.925	0.910	0.991	0.009
SHD	Worst bag	5	Hamming	0.962	0.129	0.978	0.999	0.019
		10	Hamming	0.974	0.077	0.981	0.850	0.150
		5	Entropy	0.925	0.898	0.854	0.850	0.150
		10	Entropy	0.993	0.896	0.865	0.192	0.999
		5	Bhattacharya	0.968	0.234	0.967	0.999	0.001
		10	Bhattacharya	0.912	0.924	0.982	0.999	0.019
		5	Q Statistics	0.925	0.935	0.940	0.875	0.125
		10	Q Statistics	0.930	0.925	0.925	0.935	0.065
	Best bag	5	Hamming	0.924	0.140	0.970	0.999	0.001
		10	Hamming	0.911	0.310	0.983	0.999	0.019
		5	Entropy	0.921	0.942	6.85	0.925	0.075
		10	Entropy	0.930	0.945	6.54	0.935	0.065
		5	Bhattacharya	0.974	0.045	0.969	0.999	0.001
		10	Bhattacharya	0.914	0.965	0.990	0.999	0.001
		5	Q Statistics	0.930	0.924	0.910	0.785	0.215
		10	Q Statistics	0.965	0.930	0.852	0.925	0.075
WBC	Worst bag	5	Hamming	0.994	0.028	0.995	0.999	0.019
		10	Hamming	0.996	0.02	0.993	0.999	0.019
		5	Entropy	0.924	0.789	0.899	0.992	0.008
		10	Entropy	0.910	0.856	0.878	0.994	0.924
		5	Bhattacharya	0.965	0.511	0.999	0.999	0.021

Iter	OB	NB	DM	Accuracy	Diversity	Fitness	AW	DW
		10	Bhattacharya	0.997	0.032	0.905	0.917	0.082
		5	Q Statistics	0.965	0.689	0.992	0.865	0.135
		10	Q Statistics	0.935	0.622	0.994	0.999	0.019
	Best bag	5	Hamming	0.995	0.020	0.995	0.999	0.019
		10	Hamming	0.996	0.033	0.994	0.999	0.019
		5	Entropy	0.991	0.896	0.865	0.984	0.016
		10	Entropy	0.982	0.864	0.587	0.925	0.075
		5	Bhattacharya	0.981	0.921	0.897	0.900	0.100
		10	Bhattacharya	0.997	0.023	0.904	0.917	0.082
		5	Q Statistics	0.954	0.072	0.910	0.918	0.192
		10	Q Statistics	0.979	0.070	0.904	0.019	0.999

Table 5.4: Performance evaluation of proposed model on PID, SHD and WBC here $AW = Accuracy\_Weight$ and $DW = Diversity\_Weight$

DS	OB	#bags	DM	Acc	AUC	Sen	Spe	Pre	FM	GM
PID	Worst	5	Ham	0.935	0.902	0.854	0.949	0.903	0.878	0.900
		10	Ham	**0.945**	0.904	0.909	0.898	0.833	0.869	0.904
		5	Ent	0.909	0.884	0.800	0.969	0.936	0.862	0.880
		10	Ent	0.909	0.905	0.890	0.919	0.859	0.875	0.904
		5	BC	0.905	0.894	0.904	0.902	0.899	0.906	0.884
		10	BC	0.922	0.919	0.909	0.929	0.877	0.892	0.919
		5	QStat	0.844	0.907	0.899	0.865	0.910	0.924	0.912
		10	QStat	0.890	0.912	0.875	0.924	0.911	0.824	0.916
	Best	5	Ham	**0.922**	0.919	0.909	0.929	0.877	0.892	0.919
		10	Ham	0.902	0.895	0.872	0.919	0.857	0.864	0.895

DS	OB	#bags	DM	Acc	AUC	Sen	Spe	Pre	FM	GM
		5	Ent	0.902	0.911	0.872	0.949	0.905	0.888	0.910
		10	Ent	0.915	0.920	0.865	0.931	0.875	0.872	0.931
		5	BC	0.910	0.919	0.909	0.929	0.877	0.892	0.919
		10	BC	0.922	0.919	0.909	0.929	0.877	0.892	0.919
		5	QStat	0.845	0.824	0.798	0.855	0.780	0.867	0.874
		10	QStat	0.831	0.804	0.709	0.898	0.795	0.750	0.798
SHD	Worst	5	Ham	**0.944**	0.928	0.857	0.965	0.920	0.923	0.925
		10	Ham	0.934	0.883	0.857	0.909	0.857	0.857	0.882
		5	Ent	0.925	0.922	0.904	0.939	0.904	0.904	0.921
		10	Ent	0.922	0.910	0.915	0.936	0.915	0.921	0.910
		5	BC	0.924	0.910	0.856	0.925	0.945	0.911	0.935
		10	BC	0.925	0.904	0.809	0.965	0.930	0.894	0.899
		5	QStat	0.910	0.924	0.895	0.832	0.836	0.844	0.895
		10	QStat	0.913	0.922	0.845	0.899	0.866	0.898	0.911
	Best	5	Ham	0.925	0.904	0.809	0.924	0.931	0.894	0.899
		10	Ham	**0.981**	0.984	0.916	0.969	0.954	0.976	0.984
		5	Ent	0.898	0.863	0.877	0.910	0.920	0.924	0.856
		10	Ent	0.895	0.894	0.888	0.912	0.916	0.924	0.934
		5	BC	0.925	0.904	0.809	0.934	0.934	0.894	0.899
		10	BC	0.925	0.904	0.809	0.924	0.935	0.894	0.899
		5	QStat	0.854	0.836	0.865	0.841	0.865	0.921	0.932
		10	QStat	0.865	0.878	0.869	0.876	0.892	0.910	0.924
WBC	Worst	5	Ham	**0.982**	0.976	0.953	0.965	0.920	0.976	0.976
		10	Ham	0.973	0.969	0.953	0.985	0.976	0.964	0.969
		5	Ent	0.952	0.923	0.850	0.910	0.920	0.928	0.914
		10	Ent	0.928	0.917	0.865	0.928	0.931	0.962	0.938

DS	OB	#bags	DM	Acc	AUC	Sen	Spe	Pre	FM	GM
		5	BC	0.975	0.960	0.950	0.972	0.926	0.945	0.945
		10	BC	0.938	0.915	0.878	0.923	0.965	0.975	0.928
		5	QStat	0.981	0.927	0.899	0.875	0.867	0.927	0.914
		10	QStat	0.982	0.976	0.953	0.921	0.928	0.976	0.976
	Best	5	Ham	**0.973**	0.965	0.930	0.965	0.972	0.964	0.974
		10	Ham	**0.973**	0.974	0.976	0.971	0.954	0.965	0.974
		5	Ent	0.971	0.938	0.965	0.987	0.932	0.974	0.911
		10	Ent	0.935	0.924	0.974	0.921	0.974	0.938	0.961
		5	BC	0.938	0.941	0.927	0.945	0.928	0.945	0.930
		10	BC	0.964	0.958	0.930	0.985	0.975	0.952	0.957
		5	QStat	0.927	0.911	0.878	0.859	0.899	0.845	0.865
		10	QStat	0.941	0.969	0.953	0.985	0.976	0.964	0.969
SLC	Worst	5	Ham	0.887	0.775	0.925	0.625	0.943	0.934	0.760
		10	Ham	**0.951**	0.8125	0.927	0.625	0.947	0.972	0.790
		5	Ent	0.930	0.921	0.865	0.911	0.928	0.937	0.931
		10	Ent	0.921	0.927	0.899	0.879	0.874	0.921	0.930
		5	BC	0.890	0.668	0.962	0.375	0.912	0.936	0.600
		10	BC	0.901	0.668	0.962	0.375	0.9122	0.936	0.600
		5	QStat	0.912	0.914	0.899	0.916	0.911	0.856	0.896
		10	QStat	0.915	0.918	0.875	0.896	0.845	0.875	0.895
	Best	5	Ham	**0.925**	0.740	0.981	0.50	0.929	0.954	0.700
		10	Ham	0.903	0.731	0.962	0.50	0.928	0.945	0.693
		5	Ent	0.925	0.963	0.978	0.862	0.874	0.896	0.910
		10	Ent	0.912	0.930	0.865	0.912	0.925	0.930	0.926
		5	BC	0.912	0.668	0.962	0.375	0.912	0.936	0.921
		10	BC	0.920	0.668	0.962	0.375	0.912	0.936	0.921

DS	OB	#bags	DM	Acc	AUC	Sen	Spe	Pre	FM	GM
		5	QStat	0.915	0.937	0.936	0.937	0.918	0.945	0.910
		10	QStat	0.921	0.935	0.897	0.924	0.935	0.927	0.910

Table 5.5: Performance evaluation of proposed model on PID, SHD, WBC, and SLC

5.3.2 Comparative Analysis

Results were analyzed with various diversity measures such as Entropy, Bhattacharya distance, and Q-statistics in addition to hamming distance. In the majority of the cases, hamming distance diversity measures give better fitness than other diversity measures such as Entropy, Bhattacharya distance, and q statistics. Hamming distance-based diversity measures give superior performance with 5 bags and 10 bags. Analyzed worst bag performance and best bag optimization performance on all the disease datasets and results are plotted and shown in Fig 5.2 and highlighted the convergence point. From this figure plotted analysis of four disease datasets with worst bag and best bag optimizations in eight subplots from Fig. a-h. Subplot results are Fig a and Fig b analyzed the worst bag and best bag fitness with bag sizes 5 and 10 respectively on the WBC dataset and observed that the worst bag gradually converged over the iterations at 0.995 as the fitness value. Fig c and Fig d analyzed the worst bag and best bag fitness with bag sizes 5 and 10 respectively on the SHD dataset and observed that the worst bag converged optimal value with bag size 10 compared to bag size 5. Fig e and Fig f analyzed the worst bag and best bag fitness with bag sizes 5 and 10 respectively on the PID dataset and observed that the worst bag converged optimal value with bag size 10 compared to bag size 5 and also observed that worst bag convergence at 100th iteration and with 0.984. Fig g and Fig h analyzed the worst bag and best bag fitness with bag sizes 5 and 10 respectively on the SLC dataset and observed that the worst bag converged optimal value with bag size 10 compared to bag size 5 and also observed that the worst bag convergence at 0.974. Overall most of the datasets with bag size 10 convergence take more iterations compared to bag size 5. Worst bag optimization with bag size 10 convergence is optimally compared to bag size 5. And also analyzed the accuracies of various diversity measures such as hamming, entropy, Bhattacharya distance, and Q statistics and visualized with bar plots results are shown in Fig 5.3.

In the observation majority of the cases, the Hamming distance diversity measure outperforms compared to other diversity measures. All these results are plotted for four disease datasets of the worst and best bag compared with 5 bags and 10 bags with eight plots. All the plots are shown from Fig. 5.3 a-h and explained Figures a and b are the worst bag and best bag accuracies over various diversity measures such as Hamming, Entropy, Bhattacharya, and Q statistics of bag sizes 5 and 10 on the PID dataset from this Proposed model with Hamming distance diversity measure with 10 bags giving superior performance over 5 and also in other measures bag size 10 giving better performance compare to 5 bags. Figures c and d are the worst bag and best bag accuracies over various diversity measures such as Hamming, Entropy, Bhattacharya, and Q statistics of bag sizes 5 and 10 on the SHD dataset from this Proposed model with Hamming distance diversity measure with 5 bags giving superior performance over 10 and also in other measures bag size 5 and 10 giving an almost equal performance with worst bag and with best bag size 10 is giving a superior performance with hamming distance with other measures performing almost equal performance. Figures e and f are the worst bags and best bag accuracies over various diversity measures such as Hamming, Entropy, Bhattacharya, and Q statistics of bag sizes 5 and 10 on the WBC dataset from this Proposed model with Hamming distance and Q statistics diversity measure with the worst bag with a size 5 bags giving superior performance over 10 and also in with best bag hamming measure both 5 and 10 bag size giving equal performance and remaining measures Entropy with bag size 5 and Bhattacharya with bag size 10 giving better performance. Figures g and h are the worst bags and best bag accuracies over various diversity measures such as Hamming, Entropy, Bhattacharya, and Q statistics of bag sizes 5 and 10 on the WBC dataset from this Proposed model with worst bag Hamming distance with bag size 5 is poor performance compared to bag size 10. Next entropy gives a better performance with 5 bags and with the best bag hamming and entropy giving equal performance with bag size 5.

Finally compared the proposed model performance with SOTA ensemble and non-ensemble models in terms of accuracy, AUC, Sensitivity, Specificity, Precision, F measure, and G measure. The proposed model with worst bag optimization with 10 bags and 5 bags of hamming distance gives effective performance compared to the other diversity measures. The proposed model is compared with SOTA ensemble models and results are shown in Table 5.6 and the best results are highlighted. From this PID with 5 bags sensitivity is Superior compared to SOTA and the proposed model with 10 bags. And proposed model with 10 bags gives the highest accuracy and AUC compared to SOTA models and the proposed model with 5 bags. Similarly, the SHD dataset gives giving highest Accuracy, AUC, and Specificity with 5 bags compared to SOTA and the proposed model with 10 bags. On the WBC dataset proposed model with 5 bags gives the highest Accuracy, AUC, and Sensitivity, and with 10 bags highest sensitivity. The SLC dataset compared with 5 bags gives the highest performance in terms of Accuracy, AUC, Sensitivity, and specificity. The proposed model is compared with SOTA non-ensemble models and results are shown in Table 5.7 and the best results are highlighted. From this table observed that PID with 5 bags highest precision and with 10 bags highest Accuracy and AUC. Similarly, SHD with 5 bags achieved the highest Accuracy, AUC, Precision, and F1. WBC with 5 bags achieved the Highest Accuracy, AUC, and F1 with 10 bags the highest precision. Finally observed SLC data with 10 bags achieved the highest Accuracy, AUC, Sensitivity, Precision, and F1. All the best results are highlighted.

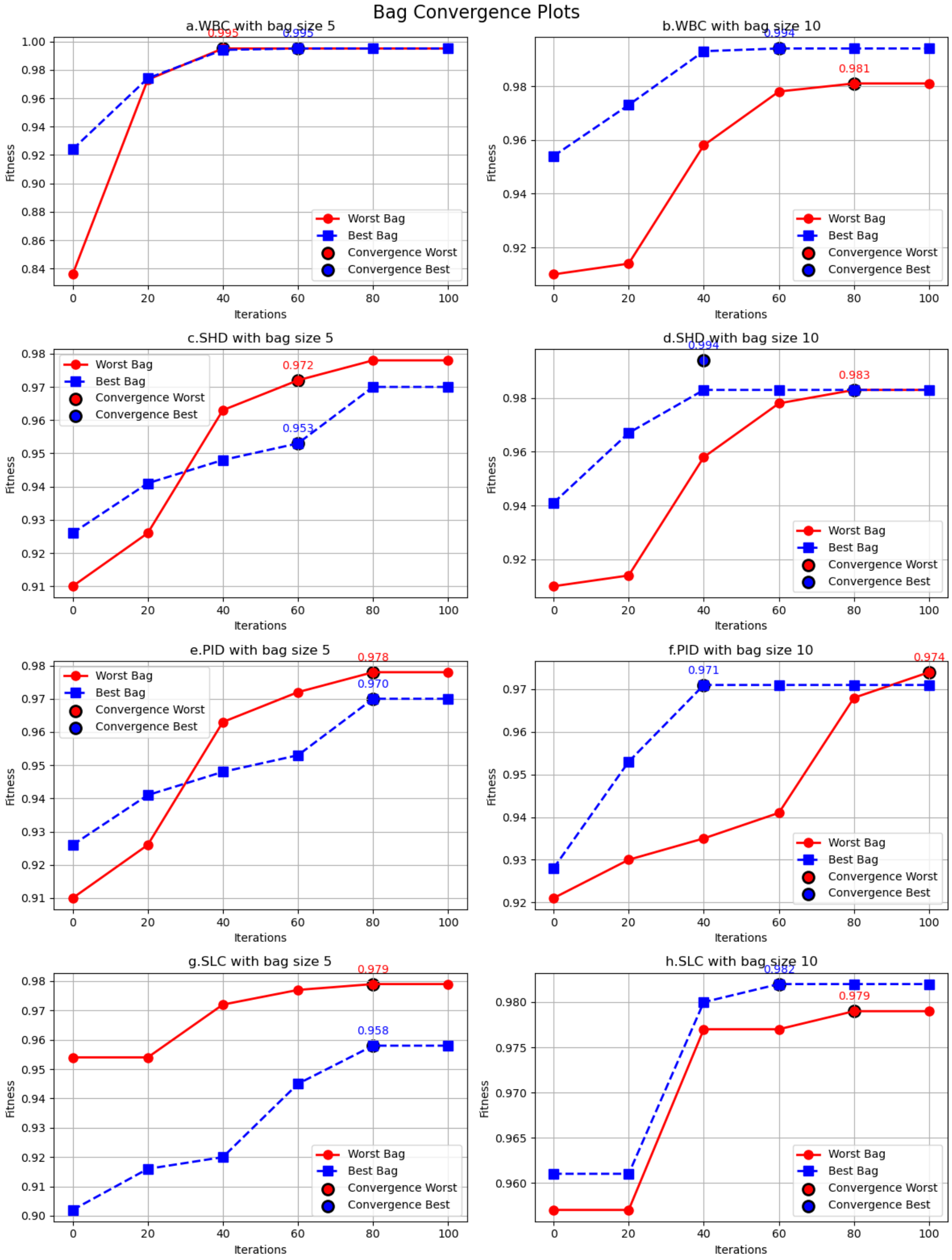

Figure 5.2: Convergence of worst bag vs best bag

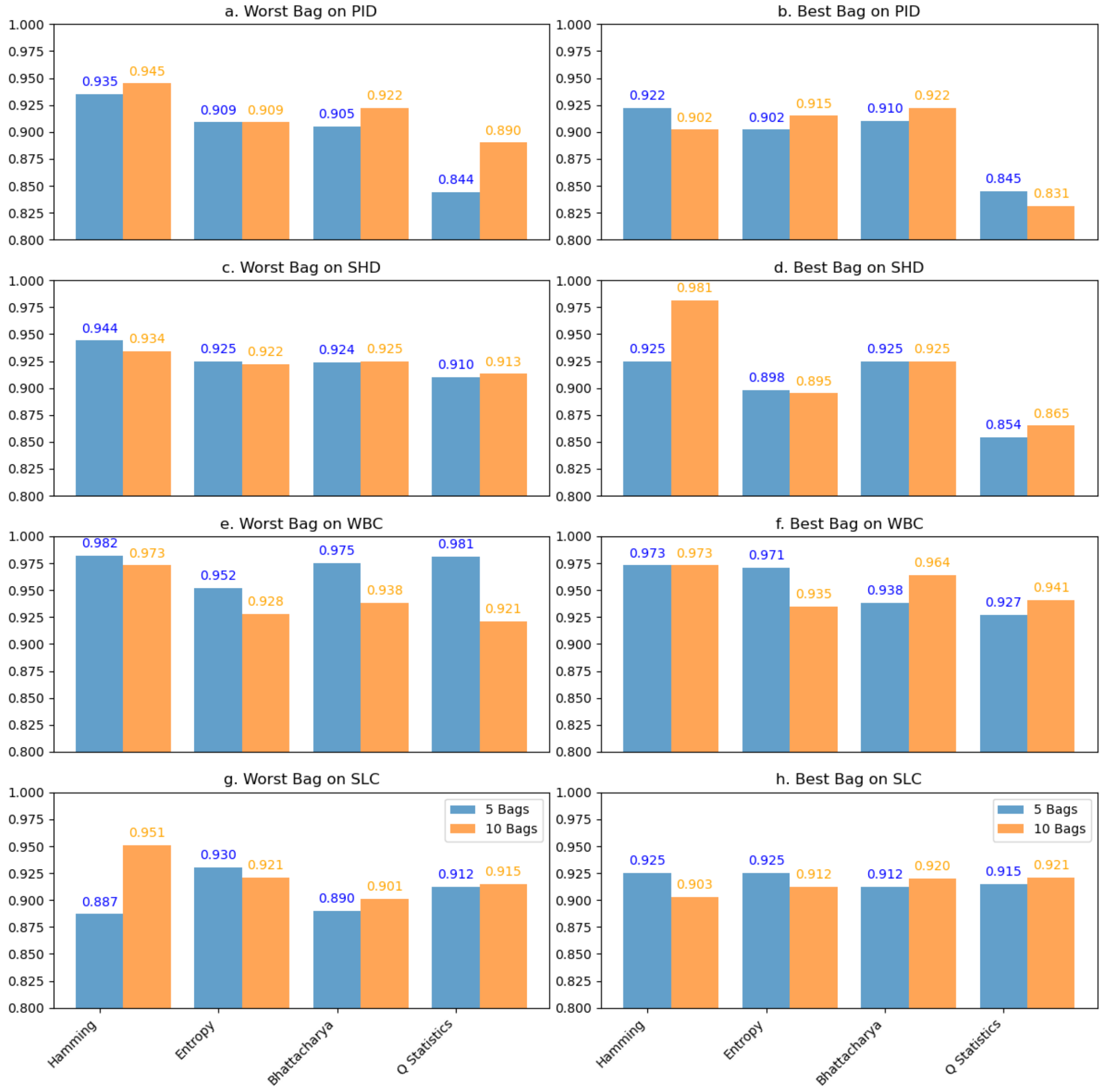

Figure 5.3: Accuracy comparison of the worst bag and best bag with various diversity measures

- **Interpretation**: Compare the calculated p-value with the chosen significance level. If the p-value is less than the significance level, reject the null hypothesis and conclude that there is a significant difference in performance between the methods. If the p-value is greater than the significance level, fail to reject the null hypothesis and conclude that there is no significant difference.

- **Repeat for Each Dataset**: Repeat the paired t-test analysis for each dataset, considering the specific performance metric chosen in step 2.

- **Results of T-test**: Summarized the results of the paired T-test for each dataset. Report the calculated t-values, degrees of freedom, p-values, and the decision regarding the null hypothesis. Provided a conclusion on whether the optimized ensemble exhibits a statistically significant difference in performance compared to the individual classifiers or baseline ensemble methods for each dataset.

Utilized statistical software or Python libraries (such as SciPy) to conduct the paired T-test. These libraries provide functions to calculate the t-value, degrees of freedom, and p-value. Calculated the t-value and p-value using the paired t-test function or module in the statistical software or Python libraries. The t-value represents the magnitude of the difference between the means of the paired samples, while the p-value indicates the probability of obtaining the observed difference if the null hypothesis is true.

By conducting paired T-tests, you can assess the statistical significance of the performance differences between the optimized ensemble and the other methods on the four datasets. This analysis provides valuable insights into the effectiveness of the proposed approach and its superiority.

5.3.3 Discussion

In the proposed study major focus is the impact of accuracy and diversity and the advantage of balancing the accuracy and diversity over the iterations and dynamic weight optimization in bagging optimization. As per our best knowledge, nobody attempted the optimization of bags using a TLBO-based approach and balances the accuracy and diversity with novel fitness functions. Our study used dynamic bag size over the optimization process with this every iteration bag contents were updated adaptive with diverse solutions and improved worst bag fitness gradually and also balanced the accuracy and diversity for an effective ensemble. With BA-TLBO results being diverse and reaching global optimal and also with optimal bags test performance of the proposed model is so effective in the disease diagnosis.

To evaluate the robustness of the proposed model the performance with four benchmark disease datasets such as PID, SHD, WBC, and SLC. The proposed model has improved the performance of class-imbalanced datasets as well.

This study evaluated various diversity measures such as hamming distance, Bhattacharya distance, entropy, and Q statistics. In the majority of cases, a proposed model with hamming distance as a diversity measure gives superior performance with 10 bags and promising results with 5 bags. And also evaluated the proposed model with worst bag optimization over the iterations analyzed the performance of best bag optimization and also analyzed the convergence nature of best and worst bags with respect to bag sizes 5 and 10. The worst bag converges gradually but the best bag converges quickly and mostly abruptly. With the best bag optimization, there is the possibility of overfitting risk and the possibility of less diverse solutions so we considered the worst bag optimization over the iterations giving the best diverse solutions in the ensemble. And also analyzed the worst and best bag accuracies with respect to four diversity measures in the analysis proposed model with hamming distance as diversity measure with bag size is 10 is giving superior performance over other diversity measures.

Overall proposed model of worst bag optimization with hamming distance diversity measure with 10 bags is giving superior performance and with 5 bags giving promising results.

Dataset	**Classifier/Model**	**Accuracy**	**AUC**	**Sensitivity**	**Specificity**	**Yr. Ref.**
PID	Stacking(LR)	0.761	0.838	0.871	0.559	2019 [64]
	Majority Voting(MV)	0.762	0.721	0.887	0.532	2019 [64]
	Bagging (Poly-SVM)	0.762	0.811	0.882	0.539	2019 [64]
	Stacking(NSGA-II)	0.838	0.859	**0.961**	0.791	2019 [64]
	Bagging (REP)	0.758	0.832	0.837	0.611	2019 [64]
	Random Subspace Method (RSM)	0.753	0.827	0.869	0.542	2019 [64]
	Random Forest	0.763	0.839	0.846	0.603	2019 [64]
	Stacking	0.688	0.665	0.742	0.587	2019 [64]
	Dia-Net	0.908	-	0.957	0.831	2020 [100]
	soft-voting	0.809	0.790	0.706	0.784	2021 [67]
	AdaBoost	0.749	0.753	0.682	0.601	2021 [67]
	Bagging	0.701	0.748	0.687	-	2021 [67]
	GradientBoost	0.718	0.753	0.487	-	2021 [67]
	XGBoost	0.69	0.75	0.675	-	2021 [67]
	CatBoost	0.745	0.753	0.65	-	2021 [67]
	Adaboost (DS)	0.750	0.810	0.849	0.566	2019 [64]
	Bagging (4.5)	0.754	0.825	0.855	0.565	2019 [64]
	Adaboost (C4.5)	0.725	0.78	0.804	0.578	2019 [64]
	Bagging (L-SVM)	0.764	0.813	0.889	0.541	2019 [64]
	Bagging (RBF-SVM)	0.681	0.734	0.867	0.333	2019 [64]
	Proposed Approach (Hamming (5 bags)	0.935	0.902	0.854	**0.949**	**This study**
	Proposed Approach (Hamming 10 bags)	**0.945**	**0.904**	0.909	0.898	**This study**
SHD	Stacking ensemble	0.923	0.922	0.934	0.910	2022 [9]
	Adaboost	0.834	0.831	0.886	0.776	2022 [9]
	GBM	0.842	0.839	0.902	0.776	2022 [9]
	Random Forest	0.902	0.899	**0.951**	0.848	2022 [9]
	Extra Tree Classifier	0.909	0.904	0.943	0.866	2022 [9]
	XGB	0.919	0.917	0.943	0.892	2022 [9]
	Proposed Approach (Hamming 5 bags)	**0.944**	**0.928**	0.857	**0.965**	**This study**
	Proposed Approach (Hamming 10 bags)	0.934	0.883	0.857	0.909	**This study**
WBC	RF	0.960	0.960	**0.950**	0.960	2021 [101]
	Gradient Boosting	0.930	0.980	0.930	0.940	2021 [101]
	Xgboost	0.970	0.970	**0.950**	**0.990**	2021 [101]
	Proposed Approach (Hamming 5 bags)	**0.982**	**0.976**	**0.953**	0.965	**This study**
	Proposed Approach (Hamming 10 bags)	0.973	0.969	**0.953**	0.985	**This study**
SLC	**Proposed Approach (SLC) (Hamming 5 bags)**	0.887	0.715	0.925	0.625	**This study**
	Proposed Approach (SLC) (Hamming 10 bags)	**0.951**	**0.812**	**0.927**	**0.898**	**This study**

Table 5.6: Proposed model comparison between SOTA ensemble models on PID, SHD, WBC, and SLC datasets

Dataset	**Classifier/Model**	**Accuracy**	**AUC**	**Sensitivity**	**Precision**	**F1**	**Yr. Ref.**
PID	SM rule miner	0.898	-	**0.946**	-	-	2017 [22]
	MLP	0.772	-	0.525	0.682	0.590	2021 [62]
	NB	0.726	-	0.661	0.759	0.707	2021 [103]
	SVM	0.741	0.740	0.712	0.754	0.732	2021 [103]
	KNN	0.719	0.663	0.612	0.583	0.597	2021 [67]
	DT	0.859	0.851	-	0.822	**0.903**	2022 [104]
	DCN	0.862	0.912	0.842	0.819	-	2022 [105]
	C4.5	0.751	0.792	0.829	0.716	0.768	2022 [103]
	RST-BAT miner	0.853	-	0.926	-	-	2018 [24]
	LR	0.751	-	0.710	0.689	0.699	2021 [102]
	DT	0.668	-	0.711	0.630	0.751	2021 [102]
	Proposed Approach (Hamming (5 bags)	0.935	0.902	0.854	**0.903**	0.878	**This study**
	Proposed Approach (Hamming 10 bags)	**0.945**	**0.904**	0.909	0.833	0.869	**This study**
SHD	KNN	0.643	0.665	**0.960**	0.590	0.730	2021 [72]
	LR	0.840	0.901	0.835	0.850	0.838	2023 [66]
	MLP	0.842	0.840	0.894	0.820	0.856	2022 [9]
	KNN	0.808	0.805	0.869	0.786	0.826	2022 [9]
	CART	0.842	0.841	0.869	0.835	0.852	2022 [9]
	LDA	0.840	0.906	0.835	0.850	0.838	2023 [66]
	SVM	0.837	0.903	0.838	0.849	0.834	2023 [66]
	Proposed Approach (Hamming 5 bags)	**0.944**	**0.928**	0.857	**0.920**	**0.923**	**This study**
	Proposed Approach (Hamming 10 bags)	0.934	0.883	0.857	0.879	0.892	**This study**
WBC	LR	0.956	-	0.958	0.971	0.965	2021 [106]
	DT	0.910	-	0.910	0.910	0.910	2021 [106]
	DT	0.910	0.890	0.890	0.910	0.910	2021 [101]
	GNB	0.940	0.940	0.930	0.940	0.940	2021 [101]
	SVM Linear	0.970	0.970	**0.980**	0.970	0.970	2021 [101]
	SVM RBF	0.970	0.960	0.930	0.960	0.960	2021 [101]
	SVM	0.971	-	0.958	0.971	0.965	2021 [106]
	KNN	0.928	-	0.9167	0.970	0.942	2021 [106]
	Proposed Approach (Hamming 5 bags)	**0.982**	**0.976**	0.953	0.920	**0.976**	**This study**
	Proposed Approach (Hamming 10 bags)	0.973	0.969	0.953	**0.976**	0.964	**This study**
SLC	**Proposed Approach (Hamming 5 bags)**	0.887	0.715	0.925	0.943	0.934	This study
	Proposed Approach (Hamming 10 bags)	**0.951**	**0.912**	**0.927**	**0.947**	**0.972**	This study

Table 5.7: Proposed model comparison with the non-ensemble model on PID, SHD, WBC, and SLC datasets

5.4 Summary

This chapter introduces a novel ensemble construction approach called BA-TLBO, which shows promising outcomes when applied to the PID, SHD, SLC, and WBC datasets. Through experimental analysis, it becomes evident that the optimized ensemble generated by BA-TLBO surpasses the performance of individual classifiers as well as other baseline ensemble methods. The research introduces an innovative fitness function that effectively balances accuracy and diversity, contributing to strong exploration and exploitation capabilities within the BA-TLBO optimization process.

A distinctive feature of the proposed model is its dynamic adjustment of accuracy and diversity weights, rendering it adaptable and robust. Diverse diversity measures such as Hamming, Bhattacharya, Entropy, and Q statistics were examined, among which the Hamming distance-based measure displayed superior performance. Additionally, the study investigated worst bag optimization in comparison to best bag optimization. The findings indicated that while the best bag approach might lead to overfitting risk and insufficient emphasis on weaker components, the worst bag optimization approach offers a more effective solution.

Overall, the BA-TLBO ensemble approach proves its potential to enhance predictive performance and produce reliable predictions across diverse datasets. Future research directions could encompass the inclusion of more classifiers, refining the optimization algorithm, or exploring alternative optimization techniques to further elevate ensemble performance.

Chapter 6

Nested Genetic Algorithm-based Classifier Selection and Placement in Multi-Level Ensemble Framework for Effective Disease Diagnosis

In the previous chapter 5 emphasizes the optimization of ensemble configurations for disease diagnosis, it's important to consider the broader context of ensemble design challenges. Effective disease diagnosis remains a formidable hurdle due to the complexities inherent in disease mechanisms. Despite the advancements achieved through ensemble-based ML models, selecting and arranging classifiers within multi-level ensembles presents intricate challenges.

Building on the dynamic ensemble optimization framework introduced in chapter 5, proposed a dynamic three-level ensemble framework in chapter 6. This framework takes the optimization process a step further by addressing the intricate process of selecting classifiers and placing them within the ensemble framework. The introduction of a nested GA and a novel fitness function enhances the ensemble design process by optimizing both classifier selection and positioning.

This extension from optimizing ensemble configurations in chapter 5 to addressing the complexities of classifier selection and placement in chapter 6 underscores the evolution of research. It showcases a comprehensive approach to ensemble-based disease diagnosis, progressing from addressing the nuances of classifier diversity and configuration optimization to resolving challenges in the strategic design of multi-level ensemble frameworks. By sequentially building upon each objective, your thesis delivers a cohesive narrative that encapsulates the journey from conceptualization to the development of advanced solutions for accurate and effective disease diagnosis. The work from chapter 5 is extended by introducing a multilevel ensemble framework optimal placement of classifiers and their selection using nested GA.

Chapter Organization: Section 6.1 provides the Preliminaries. The proposed methodology is presented in section 6.2. The experimental results and analysis are provided in section 6.3. A summary of the chapter is described in section 6.4.

6.1 Preliminaries

This section offers an overview of the background knowledge of various classifiers used in the proposed approach.

6.1.1 K-Nearest Neighbor (KNN)

According to the [122] and [123] kNN algorithm computes each training sample and each test sample distance in the dataset and returns the different k values that are closest and computational complexity is O(nd) where n and d are the sizes of the training dataset and the dimensionality. According to the [92] KNN is a non-parametric algorithm and it categorizes data points that are unlabeled.

6.1.2 Decision Tree (DT)

In accordance [124], learning a DT from a set of instances. If every instance belongs to the same class, the tree is represented by a leaf with the name of that class. If not, a test that is chosen distinct results for at least two of the cases; the instances are then divided based on this result. The root of the tree is a node that identifies the test, and the proper subtree for each outcome is obtained by repeating the algorithm on the subset of cases with that outcome.

6.1.3 Logistic Regression (LR)

According to the [125] LR is the simplest algorithm for classification. The sigmoid function converts a predicted real value into a between 0 and 1 of probability values '1'.

$$Q(z) = \frac{1}{1 + e^{-z}} \tag{6.1}$$

Where Q(z) is the estimation of the probability function as shown in Eq. 6.1.

6.1.4 Random Forest (RF)

According to the [126] Ensemble classifier family of Random Forest, It creates many trees and applies the technique of bootstrap of training data to each tree. Each tree in the forest receives procedure input prior to casting a vote for a particular class in the classification process. Finally, the class with the most votes is selected by the RF.

6.1.5 Ridge Classifier (RC)

According to the [127] Ridge regression method, converts the label data to [-1, 1], and using the regression method solves the problem. The class with the highest prediction value is chosen as the target class. For multiclass data, multiple output regression is used and a demonstration of the ridge classifier is given in sklearn documentation.

6.1.6 Gradient Boosting Classifier (GBC)

According to the [128] GBC is a significant benefit in a variety of practical applications. They are highly adaptable such as learning about different loss functions. The gradient boosting methods, with a strong emphasis on modeling machine learning aspects.

6.1.7 Extreme Gradient Boosting Classifier (XGB)

According to the [129] A popular and powerful machine learning technique is tree boosting. Data scientists use the tree-boosting system XGBoost to solve cutting-edge machine-learning problems.

6.1.8 Gaussian Naive-Bayes (GNB)

According to the [130] the GNB classifier uses the Bayes theorem using the probabilistic classifier. Using supervised learning and used in challenging real-world scenarios this classifier can be effectively learned.

6.1.9 Support Vector Machine (SVM)

Vapnik developed SVM for the classification of kernel-based machine learning models. According to the [131] [132] SVM is a popular ML-based supervised learning algorithm. [90] proposed the mathematical formula for maximizing the margin is shown in Eq. 6.2, which signifies the bias, input vector, and weight vector. In Eq. 6.3 shown linear kernel function where c is a constant.

$$\begin{aligned} k(y, y_i) &= \frac{exp - ||y - y_i||^2}{2\sigma^2} \\ \gamma &= \frac{1}{2\sigma^2} \end{aligned} \tag{6.2}$$

The equation of soft margin is followed as Eq. 6.3

$$Minimize = J(w, d, \eta) = \frac{1}{2}\|w\|^2 + c\Sigma_{i=1}^{N}\eta_i \tag{6.3}$$

$$subject\ to\ x_i(w^T y_i + d) \geq 1 - \eta_i$$

Where, $||y - y_i||$ (L2-norm) of the Euclidean distance between two points y and y_i σ is variance,

6.1.10 Stochastic Gradient Descent (SGDC)

According to [133] [134] [135] [136] [137] Fitting linear regressors and classifiers to convex loss functions, such as (linear) SVM and Logistic Regression, is done with SGDC.

6.1.11 Bagging Classifier (BC)

Leo Breiman proposed the bagging classifier as an ensemble technique in 1994 [31] [138] [124].

6.2 Proposed Methodology

In this work, a dynamic three-level ensemble framework is proposed. The proposed model has

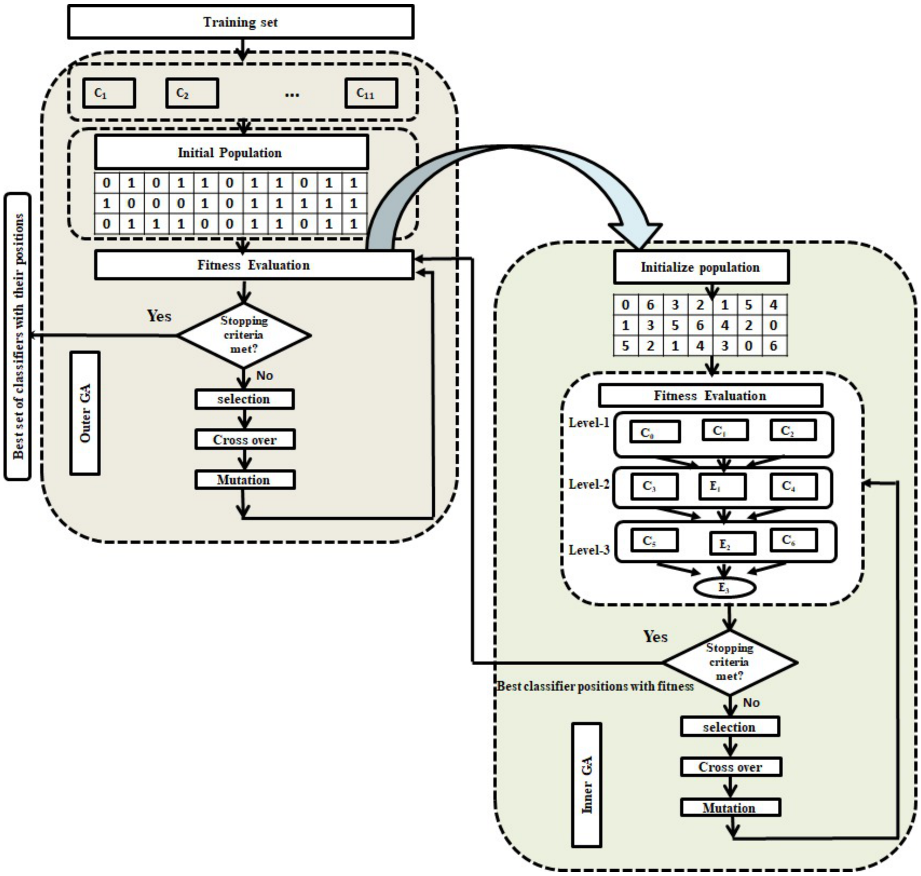

Figure 6.1: Proposed nested GA model

1. A three-level dynamic ensemble framework is proposed.

   (a) level-1 has three base classifiers C_0, C_1, C_2.

   (b) level-2 has two classifiers C_4, C_5 along with level-1 (ES_1) output.

   (c) level-3 has two classifiers C_6, C_7 along with level-2 (ES_2) output.

   (d) final output is given by ensemble score of level-3 (ES_3).

2. A total of the best seven classifiers and their positions are chosen from eleven classifiers using a nested GA. Outer GA selects the best classifiers out of eleven classifiers and inner GA optimizes the positions of classifiers in the ensemble framework.

3. A new fitness function is proposed to find the best solution.

In this, a three-level dynamic ensemble framework is proposed and it is shown in Fig. 6.1. According to the figure at each level, three classifiers are ensembled. So in total seven classifiers are required. A nested GA is employed for classifier selection and its placement in a three-level ensemble framework. Outer GA is used to find the best seven classifiers and inner GA finds the best positions for selected classifiers.

The proposed model selects 7 classifiers out of 11 but there are $\binom{11}{7}$ 7! ways. It is challenging to search for the 7 best classifiers and their positions that maximize the three-level ensemble framework. In this, a nested GA is used for this challenging task. GA is a very popular meta-heuristic evolutionary algorithm and it will better search over traditional optimization algorithms.

6.2.1 Multi Level Ensemble Framework

Classifier positions in the proposed framework are shown in Fig. 6.2. According to the figure C_0, C_1, C_2 classifiers are in level-1 and C_3 and C_4 are in Level-2 and C_5 and C_6 are

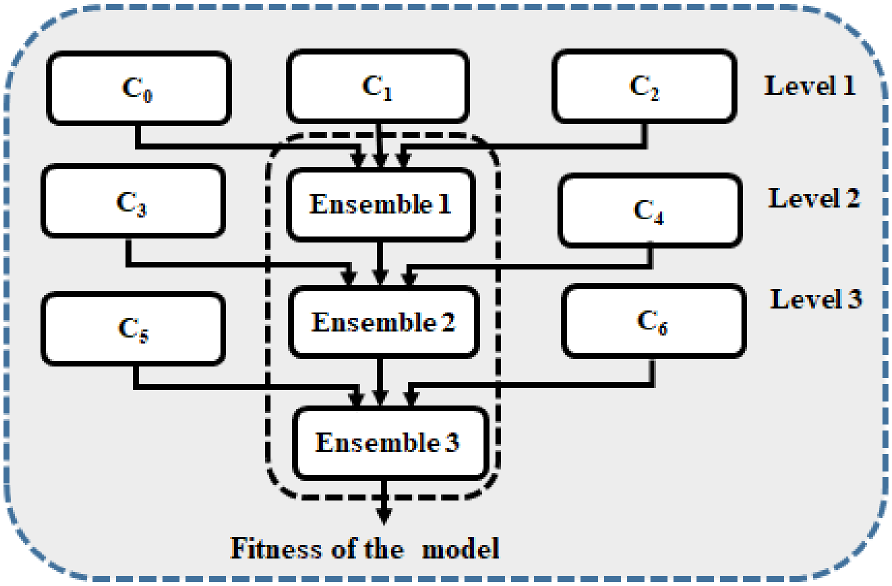

Figure 6.2: Three-level ensemble model

in level-3. Finally, level-3 gives the final outcome (Ensemble 3). Where Ensemble 1 and Ensemble 2 are the ensemble outputs of level-1 and level-2 respectively.

6.2.2 Encoding Mechanism

The main objective of the proposed model is to select seven classifiers from the pool of eleven classifiers for a three-level ensemble framework. To solve this optimization problem firstly eleven classifiers are encoded in binary chromosome format. Initially, a random population is generated and each of these individuals in the population is evaluated for fitness based on the classifier mapping and its positions. The mapping function converts eleven size chromosomes into seven size chromosomes. These selected classifiers are passed to the inner GA to compute their best positions in a three-level ensemble framework.

The inner GA takes selected classifiers as input and generates the initial population.

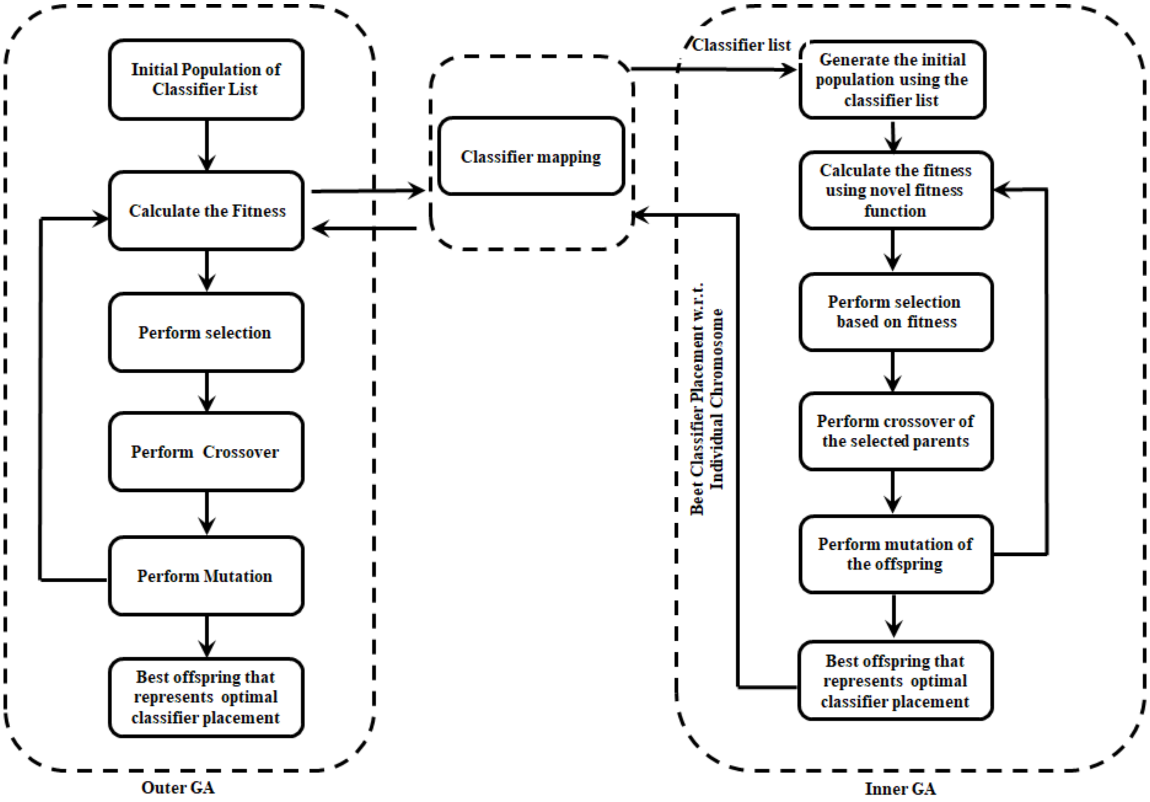
Initial Population of Classifier List
Calculate the Fitness
Perform selection
Perform Crossover
Perform Mutation
Best offspring that represents optimal classifier placement
Outer GA
Classifier mapping
Classifier list
Beet Classifier Placement w.r.t. Individual Chromosome
Generate the initial population using the classifier list
Calculate the fitness using novel fitness function
Perform selection based on fitness
Perform crossover of the selected parents
Perform mutation of the offspring
Best offspring that represents optimal classifier placement
Inner GA

Figure 6.3: Schematic diagram of nested GA

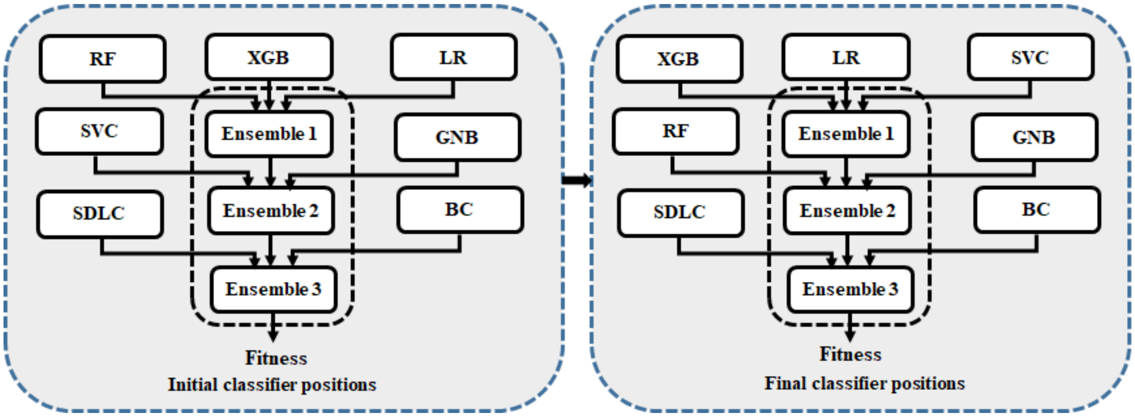

Figure 6.4: Classifier position optimization

These populations are encoded in the form of positional chromosomes, where 'x' entry at the m^{th} position in the chromosome represents the classifier is at x^{th} number must be present at the m^{th} position in the three-level ensemble framework. Both inner GA and outer GA select parents based on the best fitness score. These selected parents undergo crossover and mutation operations to generate next-generation offspring. This process is repeated up to a maximum number of iterations. At each iteration best classifiers are selected by outer GA and their positions are optimized using inner GA.

```
Data: a population of candidate solutions
Result: maximum fitness value according to classifier positions
$p_s$: population size;
$c_p$: crossover probability;
$m_p$: mutation probability;
$n_g$: generations for stopping;
Function_GA($p_s$, $c_p$, $m_p$, $n_g$)
while stop condition is false do
    Evaluate the fitness using Eq. 6.4
    Parents selection;
    Crossover probability $p_c$ perform crossover;
    Mutation probability $p_m$ perform mutation;
    Generation of the new solution with crossover probability and mutation
      probability;
    If new fitness is better than the previous, then accept the new solutions;
    Select the current best for new generations;
    Update new solution;
end
return the best fitness value-based classifier positions;
```

Algorithm 9: Classifier position optimization of 3-level ensemble framework using GA

6.2.2.1 GA-based Model Selection

In the multi-level ensemble approach, the proposed approach encoded the chromosome as a string of positions and the size is eleven. For example, the chromosome is [1, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1] this chromosome is mapped to classifiers. This classifier list is passed to inner GA to get their best positions that maximize the framework performance. The same thing is shown in Fig. 6.3. In Fig, 6.4 classifier positions before applying inner GA and after applying inner GA are shown. Here classifier positions are used as 0,1,2...6 and optimized classifier position using GA. The same is shown in Algorithm 8.

6.2.3 Novel Fitness Function

The proposed GA for optimizing classifier selection and its position uses a novel fitness function for best results. It is computed using a three-level ensemble framework and it computes the fitness of all the chromosomes in the population. The proposed novel fitness function is given in Eq. 6.4.

$$Fitness\ score = \begin{cases} ES(C_1, C_2, C_3), & \text{if } i = 1 \\ ES(C_{2i}, S_{i-1}, C_{2i+1}), & \text{if } i > 1 \end{cases} \tag{6.4}$$

where, S_i is level i Ensemble Score (ES), here i = 2,3....
ES is the ensemble score obtained using majority voting.

6.3 Experimental Results

6.3.1 Experimental setup

The Intel(R) Core(TM) i3-6006U CPU @ 2.00GHz processor and 4GB RAM was used in this experiment. The proposed methodology's modules and results analysis are carried out using Python and the Anaconda distribution 23.1.0 version used in the experiment.

Dataset	LR	GBC	RF	RC	SVC	XGB	KNN
PID	C = 0.01 Solver = liblinear	0.1 Depth = 9 Estimators = 1000	SC = Gini Depth =6	Alpha = 0.01	C =100 gamma =0.01 kernel = RBF	Depth = 17	metric = minkowski neighbors = 5 weights = uniform
WBC	C = 0.01 Solver = liblinear	0.01 Depth = 7 Estimators = 100	SC = Gini Depth =8	Alpha = 0.01	C =100 gamma =0.1 kernel = RBF	Depth = 11	metric = minkowski neighbors = 9 weights = uniform
CKD	C = 0.01 Solver = liblinear	0.1 Depth = 9 Estimators = 100	SC = Gini Depth =8	Alpha = 0.1	C =100 gamma =0.1 kernel = RBF	Depth = 16	metric = minkowski neighbors = 4 weights = uniform
SHD	C = 0.01 Solver = liblinear	0.1 Depth = 9 Estimators = 500	SC = Gini Depth =8	Alpha = 0.01	C =100 gamma =0.01 kernel = RBF	Depth = 12	metric = minkowski neighbors = 7 weights = uniform

Table 6.1: Hyper parameters of various classifiers over various datasets

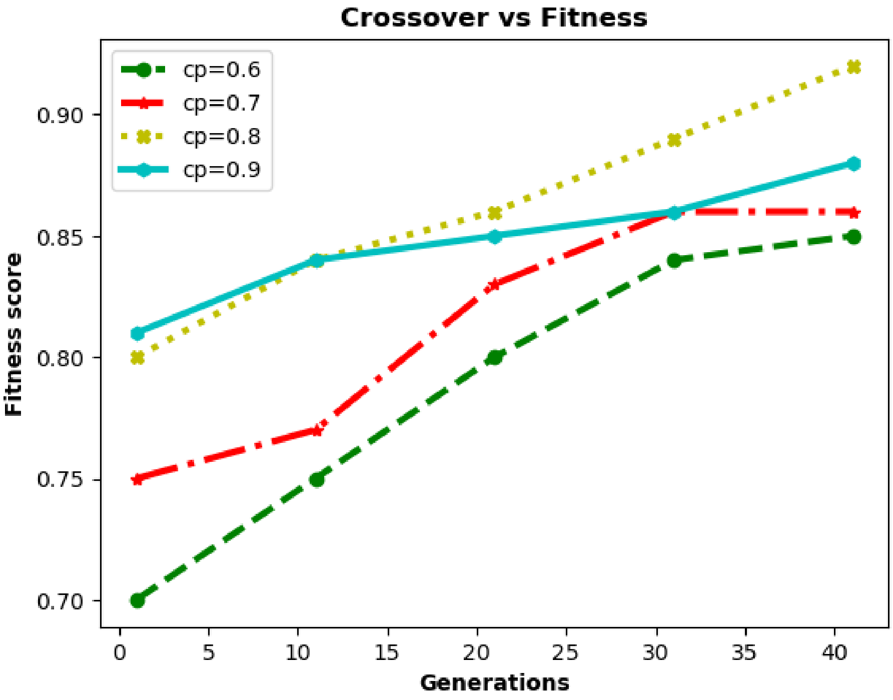

Figure 6.5: Cross over vs fitness score

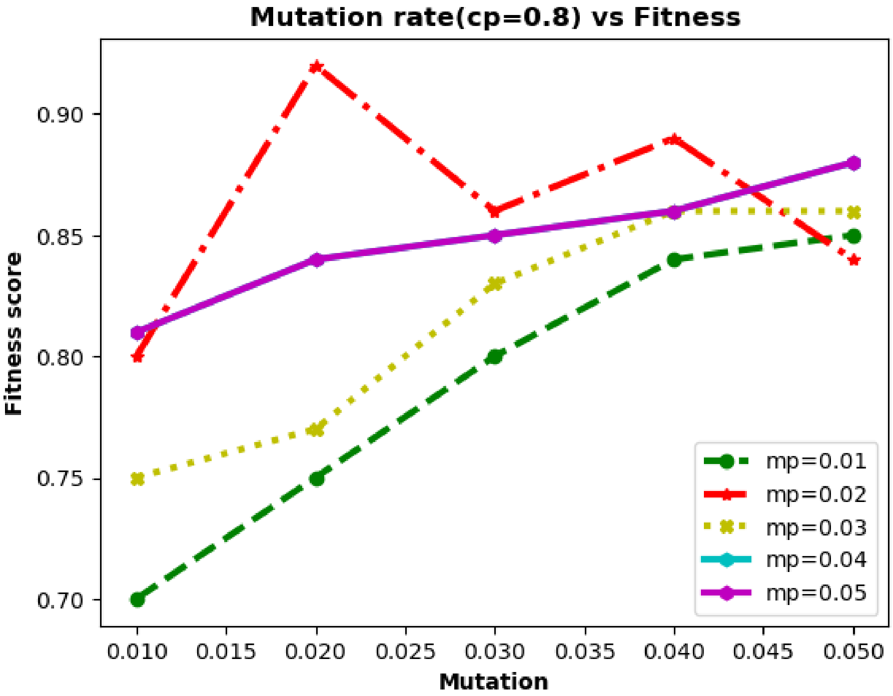

Figure 6.6: Mutation vs. fitness score

Parameter	**Value**
Maximum no of iterations	50
Crossover probability (cp)	0.80
Population size (ps)	10
Mutation probability (mp)	0.02

Table 6.2: Fine-tuned parameters used in GA for classifier placement

Dataset	LR	GNB	KNN	DT	SVM	SGDC	RC	RF	BC	GBC	XGB
PID	0.776	0.760	0.808	0.839	0.845	0.743	0.781	0.865	0.872	0.887	0.871
WBC	0.885	0.882	0.894	**0.921**	**0.923**	**0.921**	0.900	0.920	0.924	0.921	0.931
CKD	**0.900**	**0.905**	0.732	**0.921**	0.625	0.890	**0.934**	**0.987**	**0.975**	0.947	0.985
SHD	0.852	0.858	0.854	0.864	0.875	0.884	0.875	0.872	0.912	0.921	0.925

Table 6.3: Accuracy score of classifiers on various disease data sets using 10-FCV

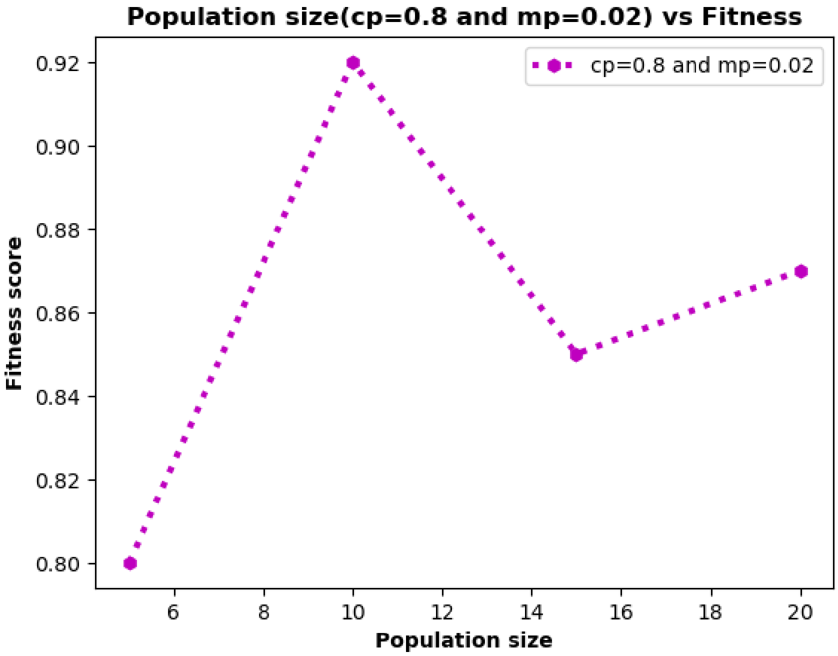

Figure 6.7: Population size Vs fitness score

6.3.2 Results

The preprocessed datasets are split into 80:20 ratio to constitute train and test datasets respectively. Our proposed model considered 11 classifiers: RF, XGB, LR, RC, GBC, SVC, GNB, KNN, DT, SGDC, and BC. Grid search is used to fine-tune these classifiers for improved performance and displays these hyperparameter values and it is shown in Table. 6.1.

Next, nested GA is used to identify the best classifiers and their positions in the three-level ensemble framework. The outer GA selects the best classifiers and the inner GA produces the best positions for selected classifiers. GA-based hyperparameters will impact the solution. Hence, fine-tuning GA parameters will significantly improve the solution of the model-tuned values tabulated in Table. 6.2. In Table. 6.3 accuracy of the 11 classifiers

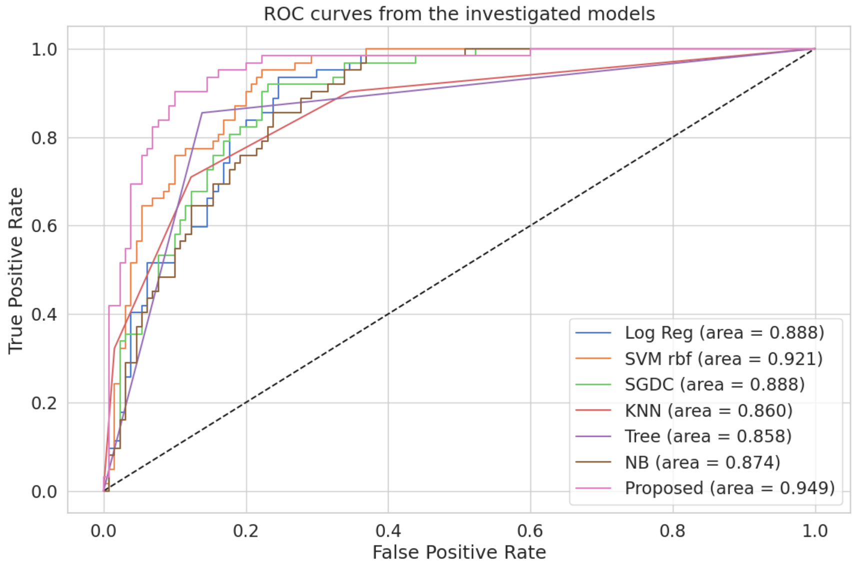

Figure 6.8: AUC of the proposed model with the base model on PID data

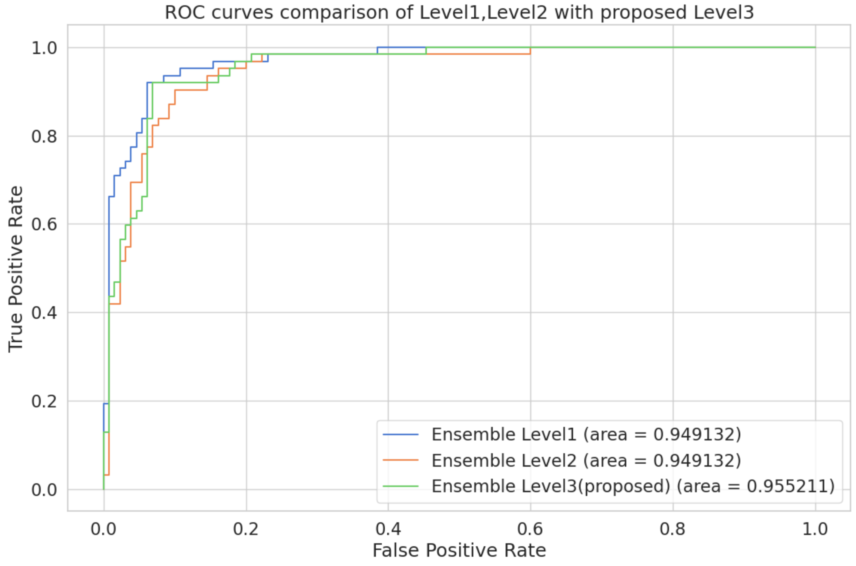

Figure 6.9: Comparison of Level-1,Level-2 with Proposed model on PID

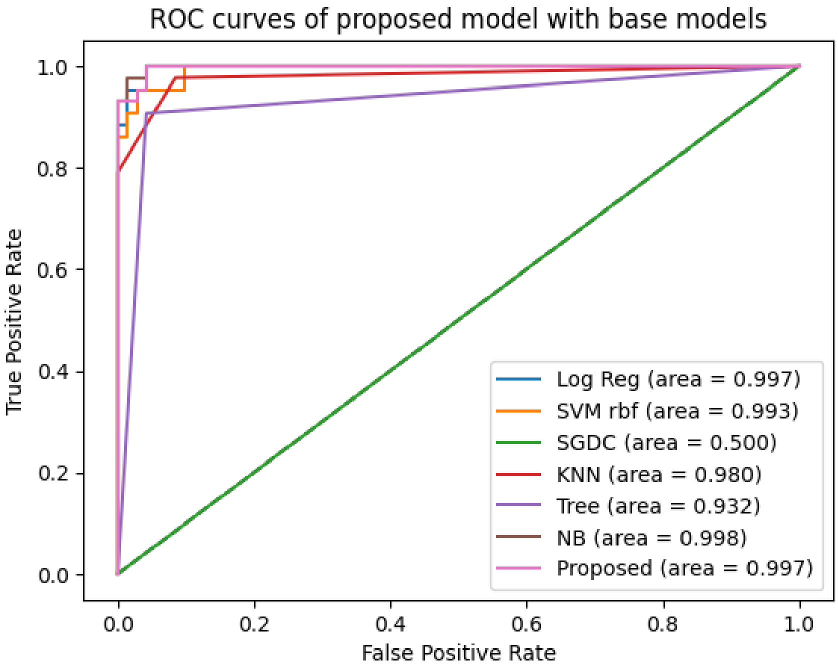

Figure 6.10: AUC of the proposed model with the base models on WBC

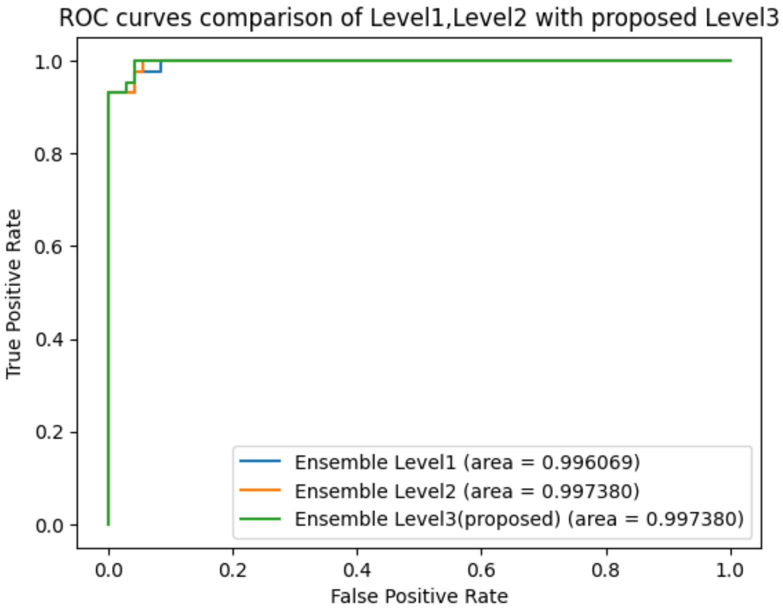

Figure 6.11: Comparison of level1, level 2 with proposed model on WBC

Dataset	**Ensemble Level**	**Accuracy**	**AUC**	**Precision**	**Recall**	**Specificity**	**F1 score**	**G Measure**
PID	1	0.895	0.935	0.717	0.532	0.90	0.611	0.692
	2	0.906	0.934	0.835	**0.903**	0.915	0.868	0.909
	3	**0.927**	**0.940**	**0.875**	**0.903**	**0.938**	**0.888**	**0.920**
WBC	1	0.960	0.995	0.975	0.928	0.980	0.950	0.952
	2	0.960	**0.997**	0.975	**0.930**	0.985	**0.952**	**0.957**
	3	**0.964**	**0.997**	**1.00**	0.906	**1.00**	0.951	0.952
CKD	1	0.972	0.935	1.0	0.980	1.0	0.989	0.989
	2	0.937	0.934	0.900	**1.0**	0.833	0.952	0.912
	3	**0.987**	**0.940**	**1.0**	**0.980**	**1.0**	**0.989**	**0.989**
SHD	1	0.890	0.921	0.914	**0.931**	**0.874**	0.912	0.922
	2	0.900	0.910	0.921	0.930	**0.894**	0.900	0.910
	3	**0.920**	**0.931**	**0.925**	**0.944**	**0.965**	**0.918**	**0.911**

Table 6.4: Level-wise performance comparison of three level ensemble

Dataset	Accuracy(%)	AUC(%)	Precision(%)	Recall(%)	F1 Score(%)	G-Measure(%)
PID	0.927	0.945	0.875	0.903	0.888	0.920
WBC	0.964	0.997	**1.00**	0.906	0.951	0.952
CKD	**0.987**	0.940	**1.00**	0.980	**0.989**	**0.989**
SHD	0.920	0.931	0.925	0.944	0.918	0.911

Table 6.5: Proposed model performance

on four disease datasets is evaluated before applying our proposed model. These results are shown in the table and the best results are highlighted. From the table, it is observed that most of the classifier's performance is good in one or two disease datasets and the majority of the dataset's classifiers' performance is poor Also ensemble-based classifiers RF, BC, GBC, and XGB performance is competitive in a few datasets.

To fine-tune crossover, mutation, and population size a sensitivity analysis is carried out and these plots are shown in Fig. 6.5, Fig. 6.6, Fig. 6.7.

Nested GA is used to determine the top seven classifiers and their positions among eleven classifiers. The proposed model's performance on various datasets is demonstrated in Table 6.4. The accuracy, AUC, precision, recall, specificity, F1 score, and G-measure of these outcomes are compared.

Dataset	**Classifiers**	**Accuracy (%)**	**AUC (%)**	**Sensitivity (%)**	**Specificity (%)**	**Yr. Ref.**
PID	Stacking(LR)	76.10	83.80	87.10	55.90	2019 [64]
	Majority Voting(MV)	76.20	72.10	88.70	53.20	2019 [64]
	Bagging (Poly-SVM)	76.20	81.10	88.20	53.90	2019 [64]
	Random Forest	76.30	83.90	84.60	60.30	2019 [64]
	Stacking	68.80	66.50	74.20	58.70	2019 [64]
	soft-voting	80.90	79.00	70.60	78.40	2021 [67]
	Bagging	70.10	74.80	68.70	-	2021 [67]
	Bagging (4.5)	75.40	82.50	85.50	56.50	2019 [64]
	Adaboost (C4.5)	72.50	78.00	80.40	57.80	2019 [64]
	Bagging (L-SVM)	76.40	81.30	88.90	54.10	2019 [64]
	Bagging (RBF-SVM)	68.10	73.40	86.70	33.30	2019 [64]
	Modified Bee	90.70	91.60	80.90	96.90	2022 [139]
	Ensemble selection	70.30	68.80	61.90	75.70	2016 [140]
	Proposed Approach	**94.90**	**94.00**	90.30	**93.80**	**This study**
CKD	Extra Tree Classifier	94.00	-	96.00	91.00	2021 [72]
	Random Tree	91.43	**96.10**	94.00	-	2021 [72]
	Proposed Approach	**98.70**	94.00	98.00	**97.00**	**This study**
SHD	Stacking ensemble	92.30	92.20	93.40	91.00	2022 [9]
	Adaboost	83.40	83.10	88.60	77.60	2022 [9]
	Random Forest	90.20	89.90	**95.10**	84.80	2022 [9]
	Extra Tree Classifier	90.90	90.40	94.30	86.60	2022 [9]
	Modified Bee	95.60	99.50	95.30	95.70	2022 [139]
	Ensemble selection	89.40	89.70	90.60	88.70	2016 [140]
	Proposed Approach	**95.10**	**93.10**	94.40	**96.50**	**This study**
WBC	RF	96.00	96.00	**95.00**	96.00	2021 [101]
	Gradient Boosting	93.00	98.00	93.00	94.00	2021 [101]
	Xgboost	97.00	97.00	**95.00**	**97.00**	2021 [101]
	Ensemble based BA	95.60	99.50	95.30	95.70	2022 [139]
	Ensemble selection	89.40	89.70	90.60	88.70	2016 [140]
	Proposed Approach	**97.80**	**97.70**	90.60	98.80	**This study**

Table 6.6: Comparison between SOTA ensemble models and proposed model

Dataset	**Classifier**	**Accuracy (%)**	**AUC (%)**	**Sensitivity (%)**	**Precision (%)**	**F-measure (%)**	**Yr. Ref.**
PID	SM rule miner	89.87	-	84.60	-	-	2017 [22]
	RST-BAT miner	85.33	-	92.6	-	-	2018 [24]
	LR	75.1	-	71.0	68.90	69.90	2021 [102]
	DT	66.80	-	71.1	63.0	75.1	2021 [102]
	MLP	77.20	-	52.50	68.2	59.00	2021 [62]
	NB	72.69	-	66.10	75.90	70.70	2021 [103]
	SVM	74.10	74.08	71.20	75.40	73.20	2021 [103]
	KNN	71.92	66.31	61.25	58.33	59.75	2021 [67]
	DT	85.98	85.11	-	82.12	80.32	2022 [104]
	DCN	86.29	91.20	84.2	81.90	-	2022 [105]
	Proposed Approach	**94.90**	**94.00**	**90.30**	**87.50**	88.80	**This study**
CKD	LR	71.71	78.40	88.60	56.48	71.80	2021 [72]
	KNN	64.39	66.50	96.00	59.01	73.09	2021 [72]
	Proposed Approach	**98.70**	**94.00**	**98.00**	**97.32**	98.90	**This study**
SHD	LR	84.07	90.10	83.58	85.06	83.80	2023 [66]
	SVM	83.70	90.30	83.08	84.92	83.40	2023 [66]
	MLP	84.25	84.00	89.43	82.08	85.60	2022 [9]
	KNN	80.85	80.54	86.99	78.67	82.62	2022 [9]
	Proposed Approach	**95.10**	**93.10**	94.40	**92.50**	91.80	**This study**
WBC	DT	91.00	-	91.00	91.00	91.00	2021 [106]
	DT	91.00	89.00	88.00	91.00	91.00	2021 [101]
	GNB	94.00	94.00	93.00	94.00	94.00	2021 [101]
	SVM Linear	97.00	97.00	88.00	97.00	97.00	2021 [101]
	SVM RBF	97.00	96.00	93.00	96.00	96.00	2021 [101]
	Proposed Approach	**97.80**	**97.70**	90.60	**97.32**	95.10	**This study**

Table 6.7: Comparison between SOTA non ensemble models and proposed model

6.3.3 Ablation study

On 4 disease data sets, the performance of the proposed model with one, two, and three levels was evaluated. Our proposed technique with three levels evaluated and from the results our proposed model performance is superior comparison with Level-1 and Level-2.

6.3.4 Receiver Operating Characteristic-Area Under Curve (ROC-AUC)

The most crucial evaluation statistic to use when assessing the effectiveness of any classification model is ROC-AUC. The AUC-ROC curve is a performance indicator for classification issues at various threshold levels. AUC is a measure of separability, and ROC is a probability curve. It reveals how well the model can differentiate between classes. The model with a higher AUC is better at classifying positive classes as positive and negative classes as negative. By analogy, the higher the AUC, the more effective the model is at distinguishing between patients with the condition and those who do not. On the ROC curve, TPR is plotted against FPR, with FPR on the x-axis and TPR on the y-axis.

A full explanation of sensitivity, specificity, and FPR will be provided via the ROC-AUC curve. The relationship between specificity and sensitivity is inverse. Therefore, Specificity declines when Sensitivity rises, and vice versa. As a result of getting more positive values when the threshold is lowered, the sensitivity increases while the specificity decreases. Similarly to, increasing the threshold results in more negative values and increased specificity at the expense of decreased sensitivity. FPR is 1 - specificity. Thus, increasing TPR also raises FPR, and vice versa.

Sensitivity and specificity analysis is crucial, especially when diagnosing diseases, as both of these performance trade-offs are necessary for efficient illness diagnosis. Hence, the ROC-AUC of the proposed model has measured w.r.t. base classifiers. These results for PID, HDD, and WBC datasets are shown in Fig. 6.8, Fig. 6.9, Fig. 6.10, and Fig. 6.11. From these figures, it is observed that the proposed model outperformed in terms of AUC among base models. Further, results on the remaining datasets are tabulated in Table 6.5.

Further, the level-wise performance of the proposed model in terms of AUC is measured on PID, and WBC datasets. These performance plots are shown in Fig. 6.9, and Fig. 6.11 respectively.

Our proposed model has been thoroughly evaluated against state-of-the-art (SOTA) ensemble and non-ensemble models. The comparative analysis revealed that our proposed model consistently outperformed both the SOTA ensemble and non-ensemble models. The detailed results can be found in Table 6.6 and Table 6.7 respectively.

6.3.5 Discussion

Our proposed 3-level ensemble model selected the best seven classifiers out of eleven classifiers and also optimized the positions of the 3-level ensemble model. To evaluate the stability of the model tested on 4 benchmark disease datasets from UCI and Kaggle repositories. In the proposed 3-level ensemble model chosen optimal classifiers were along with their positions. There are plenty of ways to choose classifiers according to their positions. It is a tedious task to choose the best combinations. To address this a GA-based approach with fine-tuned GA parameters such as crossover rate, mutation, and population size based on the novel fitness function shown in Eq. 6.4. Fine-tuning of parameters in GA is problem-specific so fine-tuning based on the problem will significantly improve the performance and also convergence effectively. In our proposed model a crossover probability is 0.80, the mutation probability is 0.02, the population size is 10, and the maximum number of iterations is 50.

The proposed model used 3 levels for the best solution and was analyzed level-wise. In the majority of the datasets, our proposed approach performance gradually increased by levels. In terms of various performance measures such as accuracy, AUC, precision, recall, specificity, F1 score, and G-measure. Most all the datasets in terms of accuracy, AUC, and precision are significantly improved with three levels whereas recall, specificity, F1 score, and G measure moderate improvements and also observed that SLC dataset accuracy and G-measure are very poor compared to the other datasets performance because SLC is severely imbalanced compare to other datasets so to improve the performance in SLC should apply oversampling techniques.

A notable thing about our proposed model is that its performance is superior even without feature selection.

Firstly, it's essential to acknowledge that the pursuit of marginal improvements is a common challenge in machine learning based research, especially when dealing with datasets where the underlying patterns are already well captured by simpler models like logistic regression or SVM. In such scenarios, the primary objective shifts from achieving significant performance gains to exploring the limits of model complexity and its potential to extract nuanced patterns or handle more intricate datasets. Therefore, while the observed improvements may seem minimal, the significance lies in the exploration of a novel framework that integrates diverse classifiers and employs sophisticated optimization techniques, such as nested genetic algorithms, to refine classifier composition and positioning. Additionally, it's worth noting that the evaluation metrics used in the study, such as accuracy significantly improved in PID dataset.

We acknowledge the instances where other methods, such as Ensemble based BA and Modified Bee, outperform our proposed method in terms of AUC and accuracy, respectively. These findings prompt a closer examination of our approach's limitations and potential areas for improvement. Factors such as algorithmic design choices and dataset characteristics may have influenced the observed performance differences. We are committed to addressing these discrepancies transparently and are exploring avenues for enhancing our method's effectiveness, including refining parameter tuning strategies and conducting further experimentation on diverse datasets. Your feedback is invaluable, and we are dedicated to ensuring the rigor and quality of our research findings moving forward. And also, in our proposed nested GA giving significant improvement in PID dataset and also effectively useful classifier selection and placement effectively. Overall proposed nested GA is useful in dynamic ensemble approaches in other domain also in addition to the health acre domain.

6.4 Summary

In this chapter, a dynamic three-level ensemble framework is introduced and thoroughly evaluated. The proposed framework leverages the power of nested GA to optimize both classifier selection and their positions within the three-level ensemble structure. The outer GA is responsible for identifying the most suitable classifiers, while the inner GA fine-tunes their arrangement in the ensemble. This optimization process is guided by a novel fitness function developed to yield improved solutions.

Within this framework, eleven diverse classifiers are considered, and a subset of seven classifiers is strategically chosen based on the maximization of the novel fitness function. The performance of the proposed model is extensively compared against state-of-the-art ensemble and non-ensemble models. The evaluation encompasses a range of metrics including accuracy, AUC, precision, recall, specificity, and G-measure. Notably, the proposed approach consistently outperforms the alternative models across these metrics, establishing its superiority in terms of predictive accuracy and robustness.

This work not only advances medical diagnostics through the introduction of a sophisticated ensemble framework but also underscores the pivotal role of ensemble methods in harnessing the collective predictive capabilities of diverse classifiers. Through comprehensive evaluations and comparisons, this research underscores the potential for improved disease diagnosis and highlights the merits of ensemble-based approaches in the realm of medical decision-making.

Chapter 7

Conclusion and Future Scope

This chapter presents the summary of the contributions of this thesis, the conclusion of each objective, and the future scope of research for further direction of this thesis is presented.

7.1 Conclusions

This thesis presents a comprehensive exploration of disease diagnosis through innovative ensemble frameworks. Beginning with a three-level stacking approach enhanced by diverse preprocessing techniques and advanced optimization methods, the work showcases consistent outperformance across various disease datasets. Building upon this success, subsequent chapters refine the ensemble strategies, addressing limitations in computation time and diversity while introducing novel fitness functions and optimization techniques. The culmination is a dynamic three-level ensemble framework, honed using nested GA and a novel fitness function, delivering exceptional accuracy, AUC, precision, recall, specificity, and G-measure results. Rigorous comparisons against state-of-the-art ensemble and non-ensemble models, supported by robust statistical analyses, firmly establish the proposed models as superior choices for disease diagnosis. In chapter 1 discussed introduction of the thesis and In chapter 2 Related work of thesis.

In chapter 3, In order to improve the disease diagnosis performance, a three-level stacking framework is proposed. In this stacking framework, level 0 learners (LR, KNN, SVM, DT, KNN, and MLP) and level 1 learners (Bagged DT, KNN, and LR) are optimized using grid search. The level 2 learner i.e., SVM is optimized with PSO. The proposed model experimented on PID, SHD, CHD, CKD, and WBC datasets. The proposed model is compared with different combinations of base learners and outperformed in terms of all the performance measures. Further, the proposed model is compared with SOTA ensemble and non-ensemble methods in terms of accuracy, AUC, specificity, and precision and it outperformed all the models in terms of AUC and accuracy on all the datasets. Finally, to prove the robustness of the proposed model a paired statistical t-test is performed. The statistical test proved that the proposed model significantly differs from all the base-level models.

In chapter 4, In the previous chapter (from chapter 3) stacking framework gives promising results but has limitations of taking more computation time and also needs to improve the diversity. In this study, we have evaluated the performance of the individual classifiers on various types of disease datasets. To improve the performance of the models we have performed bootstrapped aggregation of the training set and evaluated the performance of individual classifiers w.r.t to data bag to further improve the version we have used an ensemble approach using genetic algorithm and computed fitness using the proposed novel fitness function. In our proposed approach we have used 4 classifiers such as LR, KNN, SVM, and DT with fine-tuned hyperparameters using grid search. Further, 5-FCV was applied on the training part and divided into 5 folds and applied GA as an evolutionary search for optimal ensemble candidates of 20 learners trained on the bootstrapped data. Using 5-FCV the validation set is used to evaluate the fitness of each chromosome. Finally, we have tested the model performance in terms of Precision, Recall, Accuracy, AUC, Specificity, and G-measure. To test the robustness of the model we have tested the model with PID, CKD, SHD, and WBC datasets. we have also compared our model with state-of-the-art ensemble models and state-of-the-art non-ensemble models. Our model is giving superior performance in terms of precision, Accuracy, and Specificity. To evaluate the model performance we have tested with four other disease datasets. In all the datasets our proposed model is giving promising results. So our proposed model is recommended for use in disease diagnosis.

In chapter 5, the proposed BA-TLBO approach gives promising results in constructing optimized ensembles across the PID, SHD, SLC, and WBC datasets. The experimental analysis demonstrates that the optimized ensemble exhibits improved performance compared to individual classifiers and potentially other baseline ensemble methods. In the proposed study we have introduced a novel fitness function that balances accuracy and diversity and gives good exploration and exploitation in the BA-TLBO optimization process. And also dynamically updated the accuracy and diversity weight, making the proposed model adaptable and robust. And also analyzed various diversity measures such as Hamming, Bhattacharya, Entropy, and Q statistics out of these Hamming distance-based diversity measure performance is superior compared to others. And also analyzed the worst bag optimization and compared it with the best bag optimization and observed that with the best bag possibility of overfitting risk and not focused weak component in the ensemble. so finally hamming distance-based diversity and worst bag-based optimization will give effective results in disease diagnosis. so it is recommendable to use hamming distance-based diversity with BA-TLBO-based worst bag optimization. Further research could focus on exploring additional classifiers, enhancing the optimization algorithm, or considering other optimization techniques to improve ensemble performance.

In chapter 6, a dynamic three-level ensemble framework is proposed. All these disease datasets have undergone the pre-processing stage. After the pre-processing stage, nested GA is employed to optimize the classifiers and their positions in the proposed three-level ensemble framework. Outer GA selects the best classifiers and inner GA is used to optimize the selected classifiers' positions in the framework. Further, proposed a novel fitness function for a better solution. Our approach used eleven classifiers and chose seven classifiers by maximizing the novel fitness function.

The performance of the proposed model is compared with SOTA ensemble and non-ensemble models, and the proposed approach gave better results in terms of accuracy, AUC, precision, recall, specificity, and G-measure. Next, ROC-AUC analysis is carried out. Further, sensitivity and specificity analysis of the proposed model on the top 5 performed disease datasets is carried out. In terms of sensitivity and specificity, our proposed model performance is superior when compared with SOTA ensemble and non-ensemble models.

Overall, this work not only advances the field of medical diagnostics but also highlights the significance of ensemble approaches in harnessing the predictive power of multiple classifiers.

7.2 Future Scope

The future scope for the proposed design and development of ensemble approaches for disease diagnosis can be further extended in the following directions.

- **Integration of Advanced Classifiers**: Incorporating emerging machine learning algorithms, such as deep learning models or hybrid algorithms, could potentially lead to even higher predictive performance, enabling the framework to capture intricate patterns in complex medical data.

- **Robustness and Generalization**: While the proposed ensemble frameworks exhibit remarkable results, validating their performance on larger and more diverse datasets, including real-world clinical data, will bolster the generalization capabilities and real-world applicability of the models.

- **Interpretable Ensembles**: Developing techniques to interpret and visualize the decisions made by the ensemble models could offer valuable insights to medical practitioners, increasing the trustworthiness and adoption of the models in clinical settings.

- **Transfer Learning**: Investigating transfer learning techniques to leverage knowledge from related medical domains could expedite model development and enhance performance, especially when faced with limited labeled data in certain diseases.

- **Real-Time Diagnosis**: Adapting the ensemble frameworks for real-time diagnosis, potentially on edge devices, can drastically reduce decision time and facilitate swift medical interventions.

By delving into these future directions, the work can continue to push the boundaries of disease diagnosis using ensemble methods, contributing to both the academic understanding and practical application of machine learning in the medical domain.

Bibliography

[1] Pravali Manchala and Manjubala Bisi. Diversity based imbalance learning approach for software fault prediction using machine learning models. *Applied Soft Computing*, 124:109069, 2022.

[2] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[3] Hui Han, Wen-Yuan Wang, and Bing-Huan Mao. Borderline-smote: a new over-sampling method in imbalanced data sets learning. In *International conference on intelligent computing*, pages 878–887. Springer, 2005.

[4] Haibo He, Yang Bai, Edwardo A Garcia, and Shutao Li. Adasyn: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*, pages 1322–1328. IEEE, 2008.

[5] Yasutaka Kamei, Akito Monden, Shinsuke Matsumoto, Takeshi Kakimoto, and Ken-ichi Matsumoto. The effects of over and under sampling on fault-prone module detection. In *First international symposium on empirical software engineering and measurement (ESEM 2007)*, pages 196–204. IEEE, 2007.

[6] Leo Breiman. Bagging predictors. *Machine learning*, 24:123–140, 1996.

[7] David H Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992.

[8] Robert E Schapire. A brief introduction to boosting. In *Ijcai*, volume 99, pages 1401–1406. Citeseer, 1999.

[9] Achyut Tiwari, Aryan Chugh, and Aman Sharma. Ensemble framework for cardiovascular disease prediction. *Computers in Biology and Medicine*, 146:105624, 2022.

[10] Li Yang and Abdallah Shami. On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, 415:295–316, 2020.

[11] BH Shekar and Guesh Dagnew. Grid search-based hyperparameter tuning and classification of microarray cancer data. In *2019 second international conference on advanced computational and communication paradigms (ICACCP)*, pages 1–8. IEEE, 2019.

[12] Douglas Rodrigues, Joao P Papa, and Hojjat Adeli. Meta-heuristic multi-and many-objective optimization techniques for solution of machine learning problems. *Expert Systems*, 34(6):e12255, 2017.

[13] Cheng-Lung Huang and Jian-Fan Dun. A distributed pso–svm hybrid system with feature selection and parameter optimization. *Applied soft computing*, 8(4):1381–1391, 2008.

[14] Steffen Zschaler and Lawrence Mandow. Towards model-based optimisation: Using domain knowledge explicitly. In *Software Technologies: Applications and Foundations: STAF 2016 Collocated Workshops: DataMod, GCM, HOFM, MELO, SEMS, VeryComp, Vienna Austria, July 4-8, 2016, Revised Selected Papers*, pages 317–329. Springer, 2016.

[15] Hui Chen, Chao Tan, Zan Lin, and Tong Wu. The diagnostics of diabetes mellitus based on ensemble modeling and hair/urine element level analysis. *Computers in biology and medicine*, 50:70–75, 2014.

[16] Ioannis Kavakiotis, Olga Tsave, Athanasios Salifoglou, Nicos Maglaveras, Ioannis Vlahavas, and Ioanna Chouvarda. Machine learning and data mining methods in diabetes research. *Computational and structural biotechnology journal*, 15:104–116, 2017.

[17] Alceu S Britto Jr, Robert Sabourin, and Luiz ES Oliveira. Dynamic selection of classifiers—a comprehensive review. *Pattern recognition*, 47(11):3665–3680, 2014.

[18] Rafael MO Cruz, Robert Sabourin, and George DC Cavalcanti. Dynamic classifier selection: Recent advances and perspectives. *Information Fusion*, 41:195–216, 2018.

[19] Ludmila I Kuncheva. A theoretical study on six classifier fusion strategies. *IEEE Transactions on pattern analysis and machine intelligence*, 24(2):281–286, 2002.

[20] Tin Kam Ho. The random subspace method for constructing decision forests. *IEEE transactions on pattern analysis and machine intelligence*, 20(8):832–844, 1998.

[21] David H Wolpert and William G Macready. No free lunch theorems for optimization. *IEEE Transactions on evolutionary computation*, 1(1):67–82, 1997.

[22] Ramalingaswamy Cheruku, Damodar Reddy Edla, and Venkatanareshbabu Kuppili. Sm-ruleminer: Spider monkey based rule miner using novel fitness function for diabetes classification. *Computers in biology and medicine*, 81:79–92, 2017.

[23] Zhi-Hua Zhou. *Ensemble methods: foundations and algorithms*. CRC press, 2012.

[24] Ramalingaswamy Cheruku, Damodar Reddy Edla, Venkatanareshbabu Kuppili, and Ramesh Dharavath. Rst-batminer: A fuzzy rule miner integrating rough set feature selection and bat optimization for detection of diabetes disease. *Applied Soft Computing*, 67:764–780, 2018.

[25] Yu Su, Shiguang Shan, Xilin Chen, and Wen Gao. Hierarchical ensemble of global and local classifiers for face recognition. *IEEE Transactions on image processing*, 18(8):1885–1896, 2009.

[26] Dipak Kumar, Jogendra Garain, Dakshina Ranjan Kisku, Jamuna Kanta Sing, and Phalguni Gupta. Unconstrained and constrained face recognition using dense local descriptor with ensemble framework. *Neurocomputing*, 408:273–284, 2020.

[27] Kuldeep Singh, Shantanu Rajora, Dinesh Kumar Vishwakarma, Gaurav Tripathi, Sandeep Kumar, and Gurjit Singh Walia. Crowd anomaly detection using aggregation of ensembles of fine-tuned convnets. *Neurocomputing*, 371:188–198, 2020.

[28] Dan Xue, Xiaomin Zhou, Chen Li, Yudong Yao, Md Mamunur Rahaman, Jinghua Zhang, Hao Chen, Jinpeng Zhang, Shouliang Qi, and Hongzan Sun. An application of transfer learning and ensemble learning techniques for cervical histopathology image classification. *IEEE Access*, 8:104603–104618, 2020.

[29] Quan Gu, Yong-Sheng Ding, and Tong-Liang Zhang. An ensemble classifier based prediction of g-protein-coupled receptor classes in low homology. *Neurocomputing*, 154:110–118, 2015.

[30] Ron Kohavi, David H Wolpert, et al. Bias plus variance decomposition for zero-one loss functions. In *ICML*, volume 96, pages 275–283. Citeseer, 1996.

[31] Leo Breiman. Random forests. *Machine learning*, 45:5–32, 2001.

[32] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine learning*, 63:3–42, 2006.

[33] Tin Kam Ho. A data complexity analysis of comparative advantages of decision forest constructors. *Pattern Analysis & Applications*, 5:102–112, 2002.

[34] Peter Bühlmann and Bin Yu. Analyzing bagging. *The annals of Statistics*, 30(4):927–961, 2002.

[35] Jerome H Friedman and Peter Hall. On bagging and nonlinear estimation. *Journal of statistical planning and inference*, 137(3):669–683, 2007.

[36] Gonzalo Martínez-Muñoz and Alberto Suárez. Out-of-bag estimation of the optimal sample size in bagging. *Pattern Recognition*, 43(1):143–152, 2010.

[37] Shohei Hido, Hisashi Kashima, and Yutaka Takahashi. Roughly balanced bagging for imbalanced data. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 2(5-6):412–426, 2009.

[38] Jerzy Błaszczyński and Jerzy Stefanowski. Neighbourhood sampling in bagging for imbalanced data. *Neurocomputing*, 150:529–542, 2015.

[39] Bo Sun, Haiyan Chen, Jiandong Wang, and Hua Xie. Evolutionary under-sampling based bagging ensemble method for imbalanced data classification. *Frontiers of Computer Science*, 12:331–350, 2018.

[40] Thomas Back. *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford university press, 1996.

[41] Rohitash Chandra, Marcus Frean, and Mengjie Zhang. Adapting modularity during learning in cooperative co-evolutionary recurrent neural networks. *Soft Computing*, 16:1009–1020, 2012.

[42] Rodrigo Coelho Barros, Márcio Porto Basgalupp, Andre CPLF De Carvalho, and Alex A Freitas. A survey of evolutionary algorithms for decision-tree induction. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(3):291–312, 2011.

[43] Diego Oliva, Salvador Hinojosa, Valentín Osuna-Enciso, Erik Cuevas, Marco Pérez-Cisneros, and Gildardo Sanchez-Ante. Image segmentation by minimum cross entropy using evolutionary methods. *Soft Computing*, 23:431–450, 2019.

[44] Rohitash Chandra, Abhishek Gupta, Yew-Soon Ong, and Chi-Keong Goh. Evolutionary multi-task learning for modular knowledge representation in neural networks. *Neural Processing Letters*, 47:993–1009, 2018.

[45] Rohitash Chandra, Yew-Soon Ong, and Chi-Keong Goh. Co-evolutionary multitask learning with predictive recurrence for multi-step chaotic time series prediction. *Neurocomputing*, 243:21–34, 2017.

[46] Salvador García and Francisco Herrera. Evolutionary undersampling for classification with imbalanced datasets: Proposals and taxonomy. *Evolutionary computation*, 17(3):275–306, 2009.

[47] Seyed Ehsan Roshan and Shahrokh Asadi. Improvement of bagging performance for classification of imbalanced datasets using evolutionary multi-objective optimization. *Engineering Applications of Artificial Intelligence*, 87:103319, 2020.

[48] Jared Sylvester and Nitesh V Chawla. Evolutionary ensembles: Combining learning agents using genetic algorithms. In *AAAI workshop on multiagent learning*, pages 46–51, 2005.

[49] Jared Sylvester and Nitesh V Chawla. Evolutionary ensemble creation and thinning. In *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, pages 5148–5155. IEEE, 2006.

[50] Consuelo V García-Mendoza, Omar J Gambino, Miguel G Villarreal-Cervantes, and Hiram Calvo. Evolutionary optimization of ensemble learning to determine sentiment polarity in an unbalanced multiclass corpus. *Entropy*, 22(9):1020, 2020.

[51] R Venkata Rao, Vimal J Savsani, and DP Vakharia. Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. *Computer-aided design*, 43(3):303–315, 2011.

[52] Michael P LaValley. Logistic regression. *Circulation*, 117(18):2395–2399, 2008.

[53] Gongde Guo, Hui Wang, David Bell, Yaxin Bi, and Kieran Greer. Knn model-based approach in classification. In *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2003, Catania, Sicily, Italy, November 3-7, 2003. Proceedings*, pages 986–996. Springer, 2003.

[54] Anthony J Myles, Robert N Feudale, Yang Liu, Nathaniel A Woody, and Steven D Brown. An introduction to decision tree modeling. *Journal of Chemometrics: A Journal of the Chemometrics Society*, 18(6):275–285, 2004.

[55] Matthias Ring and Bjoern M Eskofier. An approximation of the gaussian rbf kernel for efficient classification with svms. *Pattern Recognition Letters*, 84:107–113, 2016.

[56] Damodar Edla Ramalingaswamy Cheruku and Venkatanareshbabu Kuppili. An optimized and efficient radial basis neural network using cluster validity index for diabetes classification. *group (cluster)*, 7:10.

[57] A Sampathkumar, Ravi Rastogi, Srinivas Arukonda, Achyut Shankar, Sandeep Kautish, and M Sivaram. An efficient hybrid methodology for detection of cancer-causing gene using csc for micro array data. *Journal of Ambient Intelligence and Humanized Computing*, 11:4743–4751, 2020.

[58] Srinivas Arukonda and S Sountharrajan. Investigation of lung cancer detection using 3d convolutional deep neural network. In *2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, pages 763–768. IEEE, 2020.

[59] Lior Rokach. Ensemble-based classifiers. *Artificial intelligence review*, 33:1–39, 2010.

[60] Carlos A Escobar, Daniela Macias, and Ruben Morales-Menendez. Process monitoring for quality—a multiple classifier system for highly unbalanced data. *Heliyon*, 7(10):e08123, 2021.

[61] Zeinab Arabasadi, Roohallah Alizadehsani, Mohamad Roshanzamir, Hossein Moosaei, and Ali Asghar Yarifard. Computer aided decision making for heart disease detection using hybrid neural network-genetic algorithm. *Computer methods and programs in biomedicine*, 141:19–26, 2017.

[62] Satish Kumar Kalagotla, Suryakanth V Gangashetty, and Kanuri Giridhar. A novel stacking technique for prediction of diabetes. *Computers in Biology and Medicine*, 135:104554, 2021.

[63] Ram D Joshi and Chandra K Dhakal. Predicting type 2 diabetes using logistic regression and machine learning approaches. *International journal of environmental research and public health*, 18(14):7346, 2021.

[64] Namrata Singh and Pradeep Singh. Stacking-based multi-objective evolutionary ensemble framework for prediction of diabetes mellitus. *Biocybernetics and Biomedical Engineering*, 40(1):1–22, 2020.

[65] Subasish Mohapatra, Sushree Maneesha, Subhadarshini Mohanty, Prashanta Kumar Patra, Sourav Kumar Bhoi, Kshira Sagar Sahoo, and Amir H Gandomi. A stacking classifiers model for detecting heart irregularities and predicting cardiovascular disease. *Healthcare Analytics*, 3:100133, 2023.

[66] Burak Kolukisa and Burcu Bakir-Gungor. Ensemble feature selection and classification methods for machine learning-based coronary artery disease diagnosis. *Computer Standards & Interfaces*, 84:103706, 2023.

[67] Saloni Kumari, Deepika Kumar, and Mamta Mittal. An ensemble approach for classification and prediction of diabetes mellitus using soft voting classifier. *International Journal of Cognitive Computing in Engineering*, 2:40–46, 2021.

[68] Saba Bashir, Usman Qamar, Farhan Hassan Khan, and M Younus Javed. An efficient rule-based classification of diabetes using id3, c4. 5, & cart ensembles. In *2014 12th International Conference on Frontiers of Information Technology*, pages 226–231. IEEE, 2014.

[69] Priyanka Rajendra and Shahram Latifi. Prediction of diabetes using logistic regression and ensemble techniques. *Computer Methods and Programs in Biomedicine Update*, 1:100032, 2021.

[70] Prakash Nadkarni. Chapter 10-core technologies: Data mining and, 2016.

[71] Yan-Yan Song and LU Ying. Decision tree methods: applications for classification and prediction. *Shanghai archives of psychiatry*, 27(2):130, 2015.

[72] Pankaj Chittora, Sandeep Chaurasia, Prasun Chakrabarti, Gaurav Kumawat, Tulika Chakrabarti, Zbigniew Leonowicz, Michał Jasiński, Łukasz Jasiński, Radomir Gono, Elżbieta Jasińska, et al. Prediction of chronic kidney disease-a machine learning perspective. *IEEE Access*, 9:17312–17334, 2021.

[73] Abdullah Alqahtani, Shtwai Alsubai, Mohemmed Sha, Lucia Vilcekova, and Talha Javed. Cardiovascular disease detection using ensemble learning. *Computational Intelligence and Neuroscience*, 2022, 2022.

[74] Srinivas Arukonda and Ramalingaswamy Cheruku. A novel diversity-based ensemble approach with genetic algorithm for effective disease diagnosis. *Soft Computing*, pages 1–20, 2023.

[75] Elif Derya Übeyli. Implementing automated diagnostic systems for breast cancer detection. *Expert systems with Applications*, 33(4):1054–1062, 2007.

[76] A Saifudin, UU Nabillah, T Desyani, et al. Bagging technique to reduce misclassification in coronary heart disease prediction based on random forest. In *Journal of Physics: Conference Series*, volume 1477, page 032009. IOP Publishing, 2020.

[77] Ibomoiye Domor Mienye, Yanxia Sun, and Zenghui Wang. An improved ensemble learning approach for the prediction of heart disease risk. *Informatics in Medicine Unlocked*, 20:100402, 2020.

[78] Pooja Rani, Rajneesh Kumar, Nada MO Sid Ahmed, and Anurag Jain. A decision support system for heart disease prediction based upon machine learning. *Journal of Reliable Intelligent Environments*, 7(3):263–275, 2021.

[79] Karna Vishnu Vardhana Reddy, Irraivan Elamvazuthi, Azrina Abd Aziz, Sivajothi Paramasivam, and Hui Na Chua. Heart disease risk prediction using machine learning with principal component analysis. In *2020 8th International Conference on Intelligent and Advanced Systems (ICIAS)*, pages 1–6. IEEE, 2021.

[80] Ibrahim M Nasser and Samy S Abu-Naser. Lung cancer detection using artificial neural network. *International Journal of Engineering and Information Systems (IJEAIS)*, 3(3):17–23, 2019.

[81] Leijun Li, Qinghua Hu, Xiangqian Wu, and Daren Yu. Exploration of classification confidence in ensemble learning. *Pattern recognition*, 47(9):3120–3131, 2014.

[82] Omer Sagi and Lior Rokach. Ensemble learning: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4):e1249, 2018.

[83] Tanay Agrawal. *Hyperparameter optimization in machine learning: make your machine learning and deep learning models more efficient*. Springer, 2021.

[84] Alden H Wright. Genetic algorithms for real parameter optimization. In *Foundations of genetic algorithms*, volume 1, pages 205–218. Elsevier, 1991.

[85] Graham Upton and Ian Cook. *Understanding statistics*. Oxford University Press, 1996.

[86] Russell Eberhart and James Kennedy. Particle swarm optimization. In *Proceedings of the IEEE international conference on neural networks*, volume 4, pages 1942–1948. Citeseer, 1995.

[87] Mohammed Gollapalli, Aisha Alansari, Heba Alkhorasani, Meelaf Alsubaii, Rasha Sakloua, Reem Alzahrani, Mohammed Al-Hariri, Maiadah Alfares, Dania AlKhafaji, Reem Al Argan, et al. A novel stacking ensemble for detecting three types of diabetes mellitus using a saudi arabian dataset: Pre-diabetes, t1dm, and t2dm. *Computers in Biology and Medicine*, 147:105757, 2022.

[88] Italo Zoppis, Giancarlo Mauri, and Riccardo Dondi. Kernel methods: support vector machines. 2019.

[89] Yinglin Xia. Correlation and association analyses in microbiome study integrating multiomics in health and disease. *Progress in Molecular Biology and Translational Science*, 171:309–491, 2020.

[90] Alex J Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222, 2004.

[91] Kunal Roy, Supratik Kar, and Rudra Narayan Das. Selected statistical methods in qsar, 2015.

[92] Yang Song, Jian Huang, Ding Zhou, Hongyuan Zha, and C Lee Giles. Iknn: Informative k-nearest neighbor pattern classification. In *European conference on principles of data mining and knowledge discovery*, pages 248–264. Springer, 2007.

[93] Ronald C Neath and Matthew S Johnson. Discrimination and classification. 2010.

[94] Steven L Salzberg. C4. 5: Programs for machine learning by j. ross quinlan. morgan kaufmann publishers, inc., 1993, 1994.

[95] Gary Stein, Bing Chen, Annie S Wu, and Kien A Hua. Decision tree classifier for network intrusion detection with ga-based feature selection. In *Proceedings of the 43rd annual Southeast regional conference-Volume 2*, pages 136–141, 2005.

[96] Olaiya Folorunsho. Comparative study of different data mining techniques performance in knowledge discovery from medical database. *International Journal of Advanced Research in Computer Science and Software Engineering*, 3(3), 2013.

[97] Vikas Chaurasia and Saurabh Pal. Stacking-based ensemble framework and feature selection technique for the detection of breast cancer. *SN Computer Science*, 2:1–13, 2021.

[98] Bertrand Clarke. Comparing bayes model averaging and stacking when model approximation error cannot be ignored. *Journal of Machine Learning Research*, 4(Oct):683–712, 2003.

[99] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[100] Ramalingaswamy Cheruku and Damodar Reddy Edla. Selector: Pso as model selector for dual-stage diabetes network. *Journal of Intelligent Systems*, 29(1):475–484, 2020.

[101] Nosayba Al-Azzam and Ibrahem Shatnawi. Comparing supervised and semi-supervised machine learning models on diagnosing breast cancer. *Annals of Medicine and Surgery*, 62:53–64, 2021.

[102] Ibomoiye Domor Mienye and Yanxia Sun. Performance analysis of cost-sensitive learning methods with application to imbalanced medical data. *Informatics in Medicine Unlocked*, 25:100690, 2021.

[103] Nur Ulfa Maulidevi, Kridanto Surendro, et al. Smote-lof for noise identification in imbalanced data classification. *Journal of King Saud University-Computer and Information Sciences*, 34(6):3413–3423, 2022.

[104] Chandrashekhar Azad, Bharat Bhushan, Rohit Sharma, Achyut Shankar, Krishna Kant Singh, and Aditya Khamparia. Prediction model using smote, genetic algorithm and decision tree (pmsgd) for classification of diabetes mellitus. *Multimedia Systems*, 28(4):1289–1307, 2022.

[105] Suja A Alex, J Nayahi, H Shine, and Vaisshalli Gopirekha. Deep convolutional neural network for diabetes mellitus prediction. *Neural Computing and Applications*, 34(2):1319–1327, 2022.

[106] Muhammad Sakib Khan Inan, Rizwan Hasan, and Fahim Irfan Alam. A hybrid probabilistic ensemble based extreme gradient boosting approach for breast cancer diagnosis. In *2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 1029–1035. IEEE, 2021.

[107] Maurice Clerc and James Kennedy. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE transactions on Evolutionary Computation*, 6(1):58–73, 2002.

[108] Terry Windeatt. Diversity measures for multiple classifier system analysis and design. *Information fusion*, 6(1):21–36, 2005.

[109] Ludmila I Kuncheva and Christopher J Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine learning*, 51:181–207, 2003.

[110] Ludmila I Kuncheva and Chris J Whitaker. Ten measures of diversity in classifier ensembles: limits for two classifiers. In *A DERA/IEE Workshop on Intelligent Sensor Processing (Ref. No. 2001/050)*, pages 10–1. IET, 2001.

[111] Shuo Wang and Xin Yao. Relationships between diversity of classification ensembles and single-class performance measures. *IEEE Transactions on Knowledge and Data Engineering*, 25(1):206–219, 2011.

[112] Ludmila I Kuncheva. Using diversity measures for generating error-correcting output codes in classifier ensembles. *Pattern Recognition Letters*, 26(1):83–90, 2005.

[113] Wenyao Liu, Zhaohui Wu, and Gang Pan. An entropy-based diversity measure for classifier combining and its application to face classifier ensemble thinning. In *Chinese Conference on Biometric Recognition*, pages 118–124. Springer, 2004.

[114] Jiangbo Zou, Xiaokang Fu, Lingling Guo, Chunhua Ju, and Jingjing Chen. Creating ensemble classifiers with information entropy diversity measure. *Security and Communication Networks*, 2021:1–11, 2021.

[115] HamidReza Kadkhodaei and Amir Masoud Eftekhari Moghadam. An entropy based approach to find the best combination of the base classifiers in ensemble classifiers based on stack generalization. In *2016 4th International Conference on Control, Instrumentation, and Automation (ICCIA)*, pages 425–429. IEEE, 2016.

[116] Euisun Choi and Chulhee Lee. Feature extraction based on the bhattacharyya distance. *Pattern Recognition*, 36(8):1703–1709, 2003.

[117] Thomas Kailath. The divergence and bhattacharyya distance measures in signal selection. *IEEE transactions on communication technology*, 15(1):52–60, 1967.

[118] Jingyi Lu, Jikang Yue, Lijuan Zhu, and Gongfa Li. Variational mode decomposition denoising combined with improved bhattacharyya distance. *Measurement*, 151:107283, 2020.

[119] RA Kempton and LR Taylor. Models and statistics for species diversity. *Nature*, 262(5571):818–820, 1976.

[120] Liying Yang. Classifiers selection for ensemble learning based on accuracy and diversity. *Procedia Engineering*, 15:4266–4270, 2011.

[121] Marina Skurichina, Liudmila I Kuncheva, and Robert PW Duin. Bagging and boosting for the nearest mean classifier: Effects of sample size on diversity and accuracy. In *International workshop on multiple classifier systems*, pages 62–71. Springer, 2002.

[122] Xindong Wu, Chengqi Zhang, and Shichao Zhang. Database classification for multi-database mining. *Information Systems*, 30(1):71–88, 2005.

[123] Xindong Wu and Shichao Zhang. Synthesizing high-frequency rules from different data sources. *IEEE Transactions on Knowledge and Data Engineering*, 15(2):353–367, 2003.

[124] J Ross Quinlan et al. Bagging, boosting, and c4. 5. In *Aaai/Iaai, vol. 1*, pages 725–730, 1996.

[125] Raimundo Real, A Márcia Barbosa, and J Mario Vargas. Obtaining environmental favourability functions from logistic regression. *Environmental and Ecological Statistics*, 13:237–245, 2006.

[126] Mahesh Pal. Random forest classifier for remote sensing classification. *International journal of remote sensing*, 26(1):217–222, 2005.

[127] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[128] Alexey Natekin and Alois Knoll. Gradient boosting machines, a tutorial. *Frontiers in neurorobotics*, 7:21, 2013.

[129] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794, 2016.

[130] Hajer Kamel, Dhahir Abdulah, and Jamal M Al-Tuwaijari. Cancer classification using gaussian naive bayes algorithm. In *2019 International Engineering Conference (IEC)*, pages 165–170. IEEE, 2019.

[131] Shan Suthaharan and Shan Suthaharan. Support vector machine. *Machine learning models and algorithms for big data classification: thinking with examples for effective learning*, pages 207–235, 2016.

[132] Junping Wang, Quanshi Chen, and Yong Chen. Rbf kernel based support vector machine with universal approximation and its application. In *Advances in Neural Networks–ISNN 2004: International Symposium on Neural Networks, Dalian, China, August 2004, Proceedings, Part I 1*, pages 512–517. Springer, 2004.

[133] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010: 19th International Conference on Computational StatisticsParis France, August 22-27, 2010 Keynote, Invited and Contributed Papers*, pages 177–186. Springer, 2010.

[134] Shai Shalev-Shwartz, Yoram Singer, and Nathan Srebro. Pegasos: Primal estimated sub-gradient solver for svm. In *Proceedings of the 24th international conference on Machine learning*, pages 807–814, 2007.

[135] Yoshimasa Tsuruoka, Jun'ichi Tsujii, and Sophia Ananiadou. Stochastic gradient descent training for l1-regularized log-linear models with cumulative penalty. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 477–485, 2009.

[136] Wei Xu. Towards optimal one pass large scale learning with averaged stochastic gradient descent. *arXiv preprint arXiv:1107.2490*, 2011.

[137] Tong Zhang. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *Proceedings of the twenty-first international conference on Machine learning*, page 116, 2004.

[138] Marina Skurichina and Robert PW Duin. Bagging for linear classifiers. *Pattern Recognition*, 31(7):909–930, 1998.

[139] Ashwaq Qasem, Siti Norul Huda Sheikh Abdullah, Shahnorbanun Sahran, Dheeb Albashish, Shidrokh Goudarzi, and Shantini Arasaratnam. An improved ensemble pruning for mammogram classification using modified bees algorithm. *Neural Computing and Applications*, 34(12):10093–10116, 2022.

[140] Jae Young Choi, Dae Hoe Kim, Konstantinos N Plataniotis, and Yong Man Ro. Classifier ensemble generation and selection with multiple feature representations

for classification applications in computer-aided detection and diagnosis on mammography. *Expert Systems with Applications*, 46:106–121, 2016.

List of Publications

Publications from the thesis

Journal papers:

1. **Srinivas Arukonda**, and Ramalingaswamy Cheruku, "A novel diversity-based ensemble approach with genetic algorithm for effective disease diagnosis." *Soft Computing*, 27, pp.9907–9926, Springer, 2023. **(Published, SCIE, I. F.: 4.1, Springer)**. DOI 10.1007/s00500-023-08393-5

2. **Srinivas Arukonda**, and Ramalingaswamy Cheruku, "A novel stacking framework with PSO optimized SVM for effective disease classification." *Journal of Intelligent and Fuzzy Systems*, 45(3),4105-4123, 2023. **(Published, SCIE, I.F. : 2.0, IOS Press)**.
DOI: 10.3233/JIFS-232268

3. **Srinivas Arukonda**, and Ramalingaswamy Cheruku, "Nested Genetic Algorithm-based Classifier Selection and Placement in Multi-Level Ensemble Framework for Effective Disease Diagnosis." *Computer Methods in Biomechanics and Biomedical Engineering*, 2023; **(Published, SCIE, I. F.: 1.6, Tayler and Fransis)**.
DOI: 10.1080/10255842.2023.2294264.

4. **Srinivas Arukonda**, and Ramalingaswamy "Enhancing Disease Diagnosis Accuracy and Diversity through BA-TLBO Optimized Ensemble Learning" *Bio medical signal processing and control*, 2023. (**Minor Revision submitted, I.F.: 5.1, SCIE, Elsevier)**

Conference papers:

1. **Srinivas Arukonda**, Ramalingaswamy Cheruku, TLBO Based Bagging Approach for Effective Disease Diagnosis, *CODS-COMAD, ACM India Joint International Conference on Data Science , 4-7 January 2024 at IIIT Bangalore* **(Published)**.

2. **Srinivas Arukonda**, and Ramalingaswamy Cheruku, A Novel Stacking Framework with GWO-based Feature Selection for Effective Disease Diagnosis." *INDICON: 20th India Council International Conference*, IEEE, 14-17 December 2023 at CM-RIT Hyderabad **(Published)**.

3. **Srinivas Arukonda**, and Ramalingaswamy Cheruku. "Hybrid Optimization of Bag Composition for Disease Diagnosis: Integrating Teaching-Learning-Based Optimization with Genetic Algorithm." *Advanced Engineering Optimization Through Intelligent Techniques (AEOTIT)*, Springer, 28 – 30 September 2023 at NIT Surat (**Accepted**).

4. **Srinivas Arukonda**, and Ramalingaswamy Cheruku. "Diversified Ensemble Learning: Integrating Bagging and Teaching Learning-Based Optimization with Pairwise Dissimilarity Measure." *Advanced Engineering Optimization Through Intelligent Techniques (AEOTIT)*, Springer, 28 – 30 September 2023 at NIT Surat (**Accepted**).

Book Chapter:

1. **Srinivas Arukonda**, Ramalingaswamy Cheruku, Book chapter Title is" Dynamic Weighted Ensemble Approach with Entropy-based Diversity Measure for Disease Diagnosis using TLBO Optimization" accepted in the book titled **"Decision Sciences In Bioinformatics: Theory And Practice**" **CRC Press Taylor and Francis Group**. **(Accepted**).