Knowledge and Information Systems
https://doi.org/10.1007/s10115-024-02106-6

REGULAR PAPER

Improving Alzheimer's classification using a modified Borda count voting method on dynamic ensemble classifiers

K. P. Muhammed Niyas[1] · Thiyagarajan Paramasivan[2]

Received: 8 November 2022 / Revised: 16 March 2024 / Accepted: 20 March 2024

Abstract

Alzheimer's detection is a challenging task for physicians. There are subtle differences in the bio-marker characteristics of Alzheimers and mild cognitive impairment patients which is very difficult to detect by a physician. Machine learning approaches are widely used for predicting a patient as having either Alzheimer's or mild cognitive impairment. For developing models that distinguish between Alzheimer's and mild cognitive impairment patients, the researchers used a dynamic ensemble of classifier selection algorithms. These algorithms perform voting on ensemble classifiers without considering preferential choices of the Alzheimer's and mild cognitive impairment categories. Thus, this paper applies a modified Borda count voting weightage method instead of the majority voting and Borda voting for classifying Alzheimer's, healthy control, and mild cognitive impairment patients classification on dynamic ensemble of classifier selection algorithms. Six dynamic ensemble of classifier selection algorithms are used in the study. Ten pools of classifiers including random forest, bagged decision tree, extra trees, Adaboost, rotation forest, decision tree, bagged support vector machine, bagged multilayer perceptrons, majority voting ensemble, and stacking classifier are used as classifier input for the dynamic ensemble of classifiers. The results suggest that the application of the proposed method can improve the classification performance for Alzheimer's, mild cognitive impairment, and healthy patients when compared to the traditional voting methods after applying most of the dynamic ensemble of classifier selection algorithms used in the study. The application of a modified Borda count voting method on the dynamic ensemble of classifiers resulted in an increase of balanced classification accuracy ranging from 1 to 9%. The highest balanced classification accuracy of 86% is reported when random forest is applied to meta-learning for dynamic ensemble selection algorithms with the proposed voting method. It is also noted that there is an maximum increase in balanced classification accuracy of 9% is observed when applying rotation forest on K-nearest output profiles classifier using the proposed modified Borda count voting method. Thus, the

✉ Thiyagarajan Paramasivan
thiyagu.phd@gmail.com

K. P. Muhammed Niyas
muhammedniyas350@gmail.com

[1] School of Business Management, Narsee Monjee Institute of Management Studies, Jadcherla, Hyderabad, India

[2] Department of Computer Science, Central University of Tamil Nadu, Thiruvarur, India

Published online: 16 May 2024

increase in the balanced classification accuracy after applying the proposed modified Borda count voting method can make a positive impact on high-stakes healthcare applications like Alzheimer's detection.

Keywords Machine learning · Dynamic ensemble of classifiers · Borda

1 Introduction

Machine learning (ML) is widely used in the health care fields such as disease prediction, epidemic prediction, and hospital scheduling tasks [1–3]. The recent advances in the field of ML have made possible the greatest achievements in health care such as the prediction of the outbreak of an epidemic using social media data [4]. It is also possible to detect the presence of severe diseases such as Alzheimer's, Parkinson's, and various types of Cancers using the Big Data [4, 5]. The innovation of Big Data accelerated the field of health care analytics. The crucial tasks in the medical field has got more insights through the advent of wide variety of health care data [6, 7].

According to the World Health Organization (WHO), Mental Health Gap Action Programme 2008, Alzheimer's disease (AD) is considered to be one of the most dangerous and attention-seeking mental health illnesses in the world [7]. Lifestyle, aging, and genetic factors are responsible for the increasing progression of AD [8]. Moreover, the total number of AD patients are also estimated to triple by 2050 [9]. Among all Dementia, AD is considered to be one of the most severe and dangerous dementia which is most prevalent among the aged population around the globe [9]. The severity of Alzheimer's is such that the affected person is unable to remember and recall even the daily day-to-day activities of an individual. This makes the life of an Alzheimer's patient even more miserable [7, 10, 11].

There are no perfect medications currently available to cure AD disease progression. However, the rate of progression of the AD could be reduced using certain medications [12, 13]. This is where the advanced prediction of Alzheimer's is useful. If AD patients are known in advance, it is possible to provide suitable types of medications to future AD patients considering its severity. Hence, it is possible to reduce the progression level of AD that will affect the cognitive memory of an individual [14–16]. Mild cognitive impairment (MCI) is another stage of cognitive impairment where the affected individual can still able to recall basic day-to-day activities except the historical facts and figures. This is considered to be as less harm than the severe AD stage [17]. The benefits of the advanced prediction of AD are 1. It is possible to provide only the required medications to the patients depending upon the type of dementia, 2. Unnecessary health care costs can be avoided by a well-designed medication strategy. Physicians can manage and develop a good personalized medication plan for the patient depending upon their dementia level. It will help the family members also in maintaining a proper care for the patient [17, 18].

People with severe dementia such as AD face problems for recalling past events, remembering day-to-day activities, understanding the family relationships. The overall cognitive abilities of an individual such as understanding the mathematical logic, solving complex problems are reduced during AD progression. This is a dangerous situation where the individual require the assistance of a person [19]. Hence, the Alzheimer's patient faces extreme difficulty for doing common tasks that can be easily managed by a common person [19, 20]. The severity of cognitive impairment is slightly less for MCI patients. The MCI individuals

can still manage to do daily day-to-day assistance without the assistance of external person. However, they might forget the historical facts and figures [19, 20].

This paper explores the application of various dynamic ensemble algorithms with a novel weighting approach on the Borda voting method for the classification of AD, MCI, and healthy control (HC) patients. Unlike the typical dynamic ensemble classifiers with the majority voting technique for the classification of AD, MCI, and HC patients, the proposed ensemble classifier takes the final classification decision on the basis of preferential Borda voting method with a modified weighting mechanism. This paper is organized as follows: Section 2 contains the related works, Sect. 3 contains the materials and methods, Sect. 4 contains the results and discussions, and Sect. 5 contains the conclusion.

2 Related works

The application of several machine learning (ML) algorithms is effectively used in health care applications for various disease prediction tasks [21–24, 26]. Health care datasets are very complex in nature and require proper data management mechanisms. Data duplication is a major challenge and issue in such datasets which can be resolved using incremental clustering techniques [25]. Thus, it is difficult for a doctor or physician to find out the actual disease diagnosis status for the early prediction of diseases using relevant features from such large complex datasets [21, 26, 27]. Consequently, ML algorithms are widely used for the advanced prediction of diseases. As the disease datasets are very complex in nature, the researchers are relying on the advanced ensemble models for better predictions [28]. This is most commonly seen among ML researchers for the prediction of neuro-degenerative diseases especially Alzheimer's because the exact reason for the AD is not known among the clinical practitioners. Hence, clinical experts depend on the advanced ML techniques with ensemble modeling for the prediction of AD [28].

Researchers used multimodal data that consist of magnetic resonance imaging (MRI), positron emission tomography (PET), cognitive tests for the early detection of Alzheimer's [29–32]. A convolutional neural network (CNN) is employed on the MRI dataset for the detection of AD disease by the researchers in [29]. Further, an unsupervised CNN is also used by the researchers for distinguishing MCI and AD patients [30]. In another study conducted by the researchers in [31], cost-effective simple ML algorithms such as SVM, KNN, and LR are used for the detection of AD using medication and cognitive data. Multimodal data with longitudinal time-series features are also used for distinguishing MCI and AD patients. They used SVM as the classifier for the final classification of MCI and AD [32]. Deep learning based neural network architectures are widely used by the researchers for feature extraction from neuroimages of the brain [33]. Residual neural networks are utilized on the Central Lobe regions of the brain for identifying AD patients using residual neural networks [34].

Ensemble models are widely used for the detection of AD, MCI, and HC patients [35–41]. An ensemble deep learning model consisting of convolutional autoencoders (CAE) is used by the researchers for the classification of AD, MCI, and HC patients [35, 37]. An ensemble model consists of random forest (RF) and extreme gradient boosting (XGBoosting) algorithm is used for the detection of AD, HC patients [36]. Researchers also investigated the effectiveness of a patch-based ensemble of convolutional neural network (CNN) for the detection of AD and MCI patients [39]. A combination of several deep CNNs are used for the early detection of AD patients [38]. In another similar study, the researchers developed an ensemble multistage classifier for the prediction of AD, MCI, and HC. The model consists of

Naive Bayes (NB), SVM, and KNN classifiers [40]. Further, an ensemble of SVM classifiers are applied on MRI and psychological test data for the prediction of MCI and AD patients [41].

The typical ensemble algorithms are focused on the prediction for test data after considering every data in the training set. This is the reason why researchers used dynamic ensemble of classifier selection (DES) algorithms for the classification of AD, MCI, and HC patients [42]. Performance analysis of 6 DES algorithms is performed by the researchers in [42]. They found out that the application of most of the DES algorithms is capable of increasing the classification accuracy for AD, MCI, and HC classification [42]. It is observed from the literature that the previous studies involving dynamic ensemble models used a non-preferential voting scheme for the final classification of AD, MCI, and HC patients [35–41].

Two drawbacks are observed from the literature. They are:

- Lack of application of novel ensemble algorithms for the detection of AD, MCI, and HC.
- Lack of application of novel preferential voting mechanisms in the previous ensemble models used for distinguishing AD, MCI, and HC patients.

This study investigate the application of various dynamic ensemble classifier algorithms along with a novel and modified Borda count preferential voting mechanism for the detection of AD, MCI, and HC patients. The weighting method of the Borda count approach is modified using the concept of Lift.

3 Materials and methods

This section contains the detailed information of the subjects and datasets used in the study.

3.1 Dataset description

This study used the standard dataset, namely Alzheimer's Disease Neuro-imaging Initiative-TADPOLE (ADNI-TADPOLE) dataset for the study.

ADNI-TADPOLE dataset

ADNI-TADPOLE is a challenge initiative taken by the researchers of Alzheimer's disease neuroimaging initiative (ADNI) for finding out the Alzheimer's disease progressors at an early stage of their life [43–45]. The physicians are keen to find out the AD progressors on the baseline visit of an individual. The ADNI-TADPOLE dataset unites the researchers around the globe who are working on clinical, ML, statistics data for the early prediction of Alzheimer's disease. The aim of this dataset is to find out the future AD progressors using multimodal data that consist of magnetic resonance imaging (MRI), positron emission tomography (PET), functional magnetic resonance imaging (fMRI), cerebro-spinal fluid (CSF), and other cognitive test data using advanced ML techniques [44, 45]. There are over 1737 patient data collected from the various visits of patients [43–45]. Table 1 contains the description about the ADNI-TADPOLE dataset.

Following are the features used for our study from the ADNI-TADPOLE dataset:

- MRI features such as Hippocampus volume, Ventricles volume, Entorhinal volume, Fusiform volume, and Middle Temporal Gyrus Volume are extracted. Further, the area of the 19 Region of Interests (ROI) as extracted from the FreeSurfer Software such as Right Pallidum, Right Paracentral, Right Parahippocampal, Right Pars Opercularis, Right Pars Orbitalis, Right Pars Triangularis, Right Pericalcarine, Right Postcentral, Right Poste-

Table 1 Age statistic of the samples in the ADNI-TADPOLE dataset

Statistic	HC	AD	MCI
Mean	72.90	73.19	71.9
Median	89	88	90.3
Standard Deviation	59	55	55.6

rior Cingulate, Right Precentral, Right Precuneus, Right Putamen, Right Rostral Anterior Cingulate, Right Rostral Middle Frontal, Right Superior Frontal, Right Supramarginal, Right Temporal Pole, Right Thalamus, Right Transverse Temporal are selected for the study [44, 45].

- PET features such as Fluorodeoxyglucose (FDG), and AV-45 Florbetapir measurements are used as features. Further, the cerebral metabolic rate for glucose (CMRgL) of 32 ROIs as extracted by the FreeSurfer software is also used as the feature. The 32 ROIs are: Hippocampus Right, Frontal Superior Gyrus, Middle Frontal Gyrus, Para Hippocampal, Fusiform, Middle Occipital Lobe, Angular Lobe, Inferior Parietal Lobule, Supramarginal Lobe, Temporal Middle Lobe, Precuneus Lobe, Cingulum Posterior, Lingual Gyrus, Frontal Middle Lobe, Frontal Inferior Lobe, Superior Parietal Lobule, Insular Lobe, Cingulum Anterior, Cingulum Middle, Temporal Superior Lobe, Temporal Inferior Lobe, Frontal Superior Lobe, Frontal Middle Lobe, Cingulum Posterior, Frontal Superior Medial Lobe, Middle Frontal Gyrus Orbital Part, Angular Gyrus, Superior Temporal Gyrus, Rectus Gyrus, Temporal Superior, Parietal Superior Lobe, and Supramarginal Gyrus [44, 45].
- Cognitive test features such as Mini-Mental State Examination (MMSE), Clinical Dementia Rating Scale Box (CDRSB), Alzheimer's Disease Assessment Scale 11 (ADAS11), Alzheimer's Disease Assessment Scale 13 (ADAS13), Functional Activities Questionnaire (FAQ), Montreal Cognitive Assessment Test scores are used as features for the study [44, 45].
- Cerebro spinal fluid (CSF) biomarkers such as Abeta Amyloid Peptides and Tau Protein data are used as features for the study [44, 45].
- Demographic data such as age, sex, and education are also used for the study [44, 45].

3.2 Methodology

This paper used an ML approach for the classification of AD, MCI, HC classification. The methodology used for the ML design in the sequential order is as follows:

- **Data pre-processing** The raw data consist of missing data that are handled for further processing.
- **Feature selection** The most important and relevant features are selected for the study using already existing feature selection techniques such as least absolute shrinkage and selection operator (LASSO), extreme gradient boosting (XGBoost) methods. These two methods are implemented separately in the study.
- **Data transformation** The feature values are scaled into a particular range for easy execution and processing for ML algorithms.
- **Data segregation** The entire dataset is divided into 80% of training-validation and 20% of the unseen test set. The ML models are trained and validated on the training-validation set and tested separately on the unseen test data.

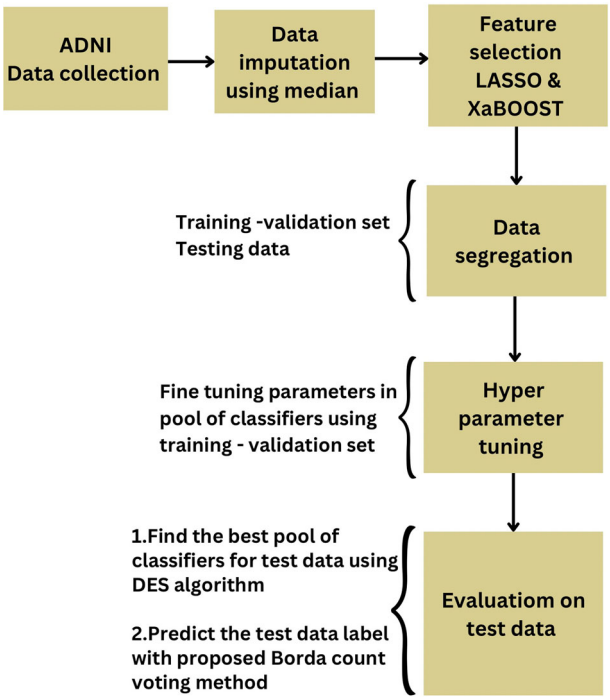

Fig. 1 Overall workflow of the study

- **Predictive modeling** A stratified 10-fold cross-validation is performed on the training-validation set. The hyper-parameters for the experiments are fine-tuned during the cross-validation stage. Finally, the trained ML model is executed on the unseen test data. Figure 1 contains the execution of the proposed methodology.

3.3 Data pre-processing

Data pre-processing techniques are vital in identifying the noises, outliers, and inconsistencies of the data. The occurrence of such mistakes often leads to bad training by the ML algorithms. Therefore, it is necessary to resolve the problems associated with the unprocessed data before starting the execution of ML algorithms [63]. The experiments are also conducted for missing values in the dataset with a median imputation strategy. The median imputation technique is a widely used technique for imbalanced datasets [63].

3.3.1 Feature selection

Feature selection is the key to finding the most relevant features of the dataset. The training of ML algorithms can be increased by using only the informative features [48]. The advanced feature selection techniques that take into consideration the multivariate feature interaction such as LASSO and XGBoost methods are used separately for finding the most important features for the study [48].

LASSO
LASSO is used for finding out the features based on their LASSO regression coefficient value. Initially, the LASSO regression is applied to the entire dataset. Then, the features whose LASSO regression is different from 0 is selected for the final prediction task [49]. The main idea behind the LASSO regression is to minimize the cost function by finding an optimal value for the LASSO coefficients of the features [49].
XGBOOST
XGBOOST technique uses a boosting ensemble where the features are selected sequentially one by one based on their importance using boosting technique [50, 51]. The boosting technique will improve the classification performance for the wrongly unclassified samples in each iteration. The XGBOOST technique is based on the principle of the gradient boosting trees [50, 51]. This method selects those features that perform well in each constructed DT in the consequent iterations. The subset of features that are correctly classified in all the DTs are taken as the final features using the XGBoost method [50, 51].

3.3.2 Data transformation

All the values that are considered for the study are normalized within a range of values using Z-score. Every feature value is scaled to a value between mean and standard deviation of the respective features [52, 53]. The benefits of this approach are that the values of a feature are scaled to a particular range. Hence, the execution and operation of the ML models become easier. Equation 1 contains the formula for data transformation.

$$scaled\,value = (feature\,value - mean(features))/standard\,deviation(features) \quad (1)$$

where scaled value is the new value after scaling, mean of features is the mean of respective feature, and standard deviation is the standard deviation of respective features.

3.4 Data segregation

The data after pre-processing are split into training, validation, and testing stages. The primary aim of the splitting is to train using some portion of the data, validate using the other divided portion of the data. Then, the resultant model is tested with unseen test data. Thus, the issues related with overfitting is handled after data segregation [54, 55].

A stratified 10-fold cross-validation is performed on the 80% patient's data that are allotted for training-validation. As our dataset is imbalanced, the stratified 10-fold cross-validation is used for our study. Moreover, it is observed from the literature that the stratified 10-fold cross-validations are mostly used technique for imbalanced health datasets also motivated us to use this strategy for our evaluation purpose [56, 57]. The halving grid search is used as the hyper-parameter technique for the cross-validation techniques on the training-validation set. The remaining 20% of the data are used for evaluating the models on the unseen test data.

Figure 2 illustrates the data segregation performed for the study.

3.5 Predictive modeling

This section contains detailed information on the ML models used in our study. The experiments are conducted on ensemble classifiers. The proposed modified novel Borda count is applied to the following ensemble classifiers:

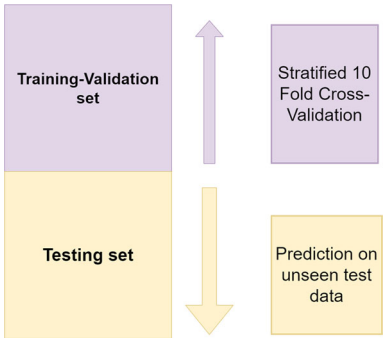

Fig. 2 Data segregation used in the study

3.5.1 Ensemble classifiers

Ensemble models makes the prediction on a test data after considering the decisions made by all the pool of classifiers. It is one of the efficient way of bringing diversity of information on to a machine learning classifier while making the classification decisions. The following ensemble classifiers are used in the study:

- **DES algorithms**
  DES algorithms find out an optimal set of ensemble classifiers for each test data dynamically based on its predictive ability on the region of competence [58–61]. A region of competence is the nearest neighbor of the respective test data. Thus, DES algorithms dynamically find a set of classifiers for each test data based on the ensemble of the classifier's performance in the region of competence. Figure illustrates the framework of a DES algorithm [58–61].
  K-nearest oracle-eliminate (KNORAE)
  KNORAE selects all those classifiers that correctly predict every training data in the region of competence. If there are no such classifiers, then the classifiers with the best performance will be assigned to the test data [58–61].
  K-nearest oracle union (KNORAU)
  KNOARU selects all those classifiers that correctly predict at least one of the training data in the region of competence. If there are no such classifiers, then none of the classifiers are considered for the final decision-making process [58–61].
  K-nearest-output profiles (KNOP)
  KNOP select those classifiers that select at least one sample in the region of competence. Unlike KNORAU, this method calculates the region of competence using the decision of the base classifier [58, 59, 62].
  Dynamic ensemble selection K-nearest neighbor (DES-KNN) DES-KNN method utilizes both the accuracy and diversity of the base classifiers while making the final decision. Initially, the most 'n' accurate classifiers from the region of competence is calculated, then the most 'm' diverse classifiers from these accurate classifiers are shortlisted for making the final classification for the test data [58–61].
  Dynamic ensemble selection performance (DESP) DESP method selects those classifiers from the base classifiers whose performance is higher than a random classifier. An ensemble set of classifiers from the base classifiers whose performance is higher than the random classifier is assigned for the new test data [58–61].
  Dynamic ensemble selection multi-imbalanced (DES-MI) DES-MI method uses a weight-based approach for controlling the dissemblance of the training set data in the

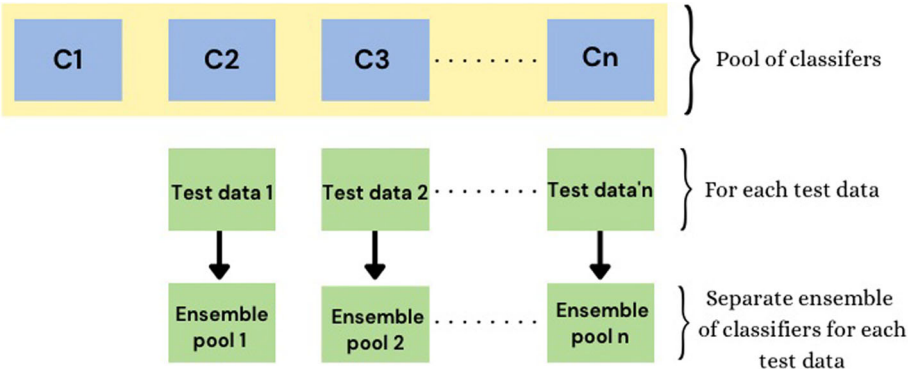

Fig. 3 Overall framework of a DES algorithm

region of competence. Then, the classifiers that predict every training data in the region of competence is selected after handling the imbalanced data in the region of competence [58–61].

Meta-learning for dynamic ensemble selection (META-DES) META-DES method finds out the meta-features associated with the base classifiers from the pool of classifiers. The meta-features are extracted for each classifier from the region of competence like the overall local accuracy, the posterior probability for each label, classifier's confidence which is the perpendicular distance between the input sample and decision boundary of the classifier [58–61].

DES Kullback–Leibler divergence

This method calculates the competence of a classifier after calculating the Kullback–Leibler divergence between the output produced by the vector of the class produced by the base classifier and a random classifier. The classifiers whose competence is higher than the random classifier are selected for the given test data [65].

- **Stacked classifier** A stacked classifier is the one where the individual classifications of all the classifiers are taken as the input for the meta-classifier. The meta-classifier consider the individual classifications as the feature inputs and the inputs are fed as features for final classification task [66, 67].

Figure 3 illustrates the overall framework of a DES algorithm.

3.5.2 Proposed modified novel Borda method

This study proposes a novel modified Borda count method for predicting the final classification of AD, MCI, and HC patients while using ensemble classifiers. The detailed explanation of the proposed modified novel Borda method is as follows:

Modified weighted Borda count

The typical Borda count method uses preferential voting in which the voters select candidates based on their quantified preferences. The similar Borda count concept is widely used by the researchers in the ensemble voting classifiers [69, 70]. The most commonly used majority voting classifiers suffer from the drawback of selecting the label on the basis of majority decisions [69–72]. However, the simple Borda count voting method considers numerical preferences for every label while making the final classification decision.

The overall preferences of every classifier are added together and the label with the highest preference is selected. The weighted preferences are found using the posterior probabilities

for every label and the respective label-wise preferences are summed up for each classifier. Then, the label with the highest preferences are selected. This study proposes a modified Borda weighting mechanism based on the concept of lift which can improve the drawback of the already existing Borda count [69–71].

The pseudocode for typical Borda count is given in Algorithm 1:

Borda count weighting mechanism using lift Lift is a probability concept used in market basket analysis for finding the frequent item sets in a purchase [73–75]. It is a measurement in association rule mining for assessing how likely the customers are going to purchase a set of items together [73–75]. This measure finds the complementary between a set of items in a purchase made by customer [73–75]. This study proposes a novel Borda count voting using the concept of lift in association rule mining for ensemble classifiers.

The modified Borda count method with the Lift weighting mechanism is illustrated in Algorithm 2.

The voting mechanism of the conventional Borda voting method involves the selection of labels based on their posterior probabilities. The label with the highest posterior probability is assigned to the new test data. The traditional ensemble models involve the application of majority voting technique for the final classification. However, the drawback of the majority voting is that it won't consider the preferences and weightage choice of the classifier while making the decision [76, 77]. The Borda count method rectifies this issue by using a preferential voting mechanism where a weightage is given for every classifier's choice.

The proposed approach improves the typical Borda count method by finding the measure of association of the ensemble classifiers in making final prediction. It applies the concept of lift in the weighting procedure of Borda voting. The Lift measures the complimentary ratio between the likelihood posterior probability of predicting a test label 'A' using an ensemble of classifiers from a set 'S' to the likelihood posterior probability of predicting the same test label 'A' with the individual classifiers on the set 'S'. Thus, the concept of Lift is able to measure the association complimentary between a set of classifiers for predicting a label.

The equation for finding the final classifier using Borda count method for a new test data is given in Eq. (2):

$$Testdata_Label = max(\sum_{i=1}^{n}\sum_{j=1}^{k} Posterior_probability_label_i(Classifier_j)) \quad (2)$$

Equation (2) indicates that the test data are assigned with a label which is having the maximum aggregate posterior probability. The aggregate posterior probability of a label is the total sum of the posterior probability reported after executing with every classifier separately. Then, the label with the highest posterior probability is selected and assigned for the given test data. In Eq. (2), Testdata_Label is the label assigned for a test data, Posterior_probability_label is the Posterior probability assigned for a given label 'j', Classifier_j is the j^{th} classifier considered.

The simple equation for finding the association between two purchasing items using Lift is as follows:

$$Lift(X/Y) = P(X \cap Y)/P(X) * P(Y) \quad (3)$$

Let us consider X and Y are two items. Then, Lift(X/Y) is the Lift associated between the two items, the numerator of Eq. (3) is the event of both X and Y are bought together, denominator of Eq. (3) is the probability of happening of buying X, and P(Y) is the probability of buying of Y. Thus, Eq. (3) measures the complimentary ratio between the probability of occurrence of both X and Y if bought together by the customer to the probability of occurrence

```
Data: Pool of classifiers, Test dataset
Result: Final classification of a patient as HC or AD or MCI on the test dataset
/* 1. Initializing the size of pool of classifiers*/ ;
Initialize int size = Number of classifiers in the ensemble pool ;
/* 2. Using a string array pool of classifiers for storing the pool of classifiers */ ;
Str Pool of classifiers[size] = Pool of classifiers ;
/* 3. Initializing a variable i, j equal to zero for storing posterior probabilities */ ;
Initialize int i=j=0 ;
/* 4. Initializing a variable named label_size for storing the count of labels */ ;
Initialize int label_size = 3 ;
/* 5. Initialize the three labels ie, AD, MCI, HC */ ;
Initialize int label[3]=[AD:0, MCI:1, HC:2] ;
/* 6. This main loop assigns a label out of HC, MCI, AD for each test data dynamically */ ;
while not the end of Test dataset do
    /* 6.1 Finding best set of ensemble classifiers using DES */ ;
    New Set = Best set of classifiers from ensemble pool of classifiers for the given test data using
      DES classifiers ;
    size1 = length(New Set) ;
    /* 6.2 Initializing a multidimensional array posterior probability for storing every pool of
      classifier's posterior probability for the three labels */ ;
    Declare float posterior_probability[size1][label_size] ;
    /* 6.2.1 Assigning the float posterior probability values of the selected classifiers */ ;
    while i< size1 do
        while j<label_size do
            posterior_probability[i][j] = Posterior Probability of classifier i with respect to individual
              label 'j' ;
            j = j+1 ;
        end
        i = i+1 ;
    end
    /* 6.3 Initializing variable for storing the sum of posterior probability of every classifier */ ;
    Declare float sum_posterior[size1] ;
    Initialize int U=1 ;
    /* 6.4 This inner loop finds the label with the highest posterior probability and assigns to the test
      data */ ;
    while U<size1 do
        Store the sum of posterior probability scores with respect to every label in sum_posterior[U] ;
        U += 1
    end
    Initialize FinalLabel = Label with maximum posterior probability value ;
    Return FinalLabel for given test data ;
end
```

Algorithm 1: Pseudocode for typical Borda count.

of buying two items separately by the customer. The higher value of Lift(X/Y) indicates there is a strong association between the two items.

Taking inspiration from the concept of the Lift, a modified Lift formula is developed for the voting mechanism of the Borda count. The modified Lift formula is given in Eq. (4):

$$Lift_new(x1, x2, x3, ..x_n/y) = P(x1 \cap x2 \cap x3 \cap ... \cap x_n/y)/(P(x1/y) * P(x2/y) * p(x3/y) * ... * p(x_n/y) \quad (4)$$

In Eq. (4), the Lift_new(x1, x2, x3,..x_n/y) is the Lift value associated when a set of classifiers say x1, x2, x3,....., x_n are used for classifying a label 'y'. It is the ratio of the posterior probability associated when all the classifiers are used classifying a label 'y' to the product of individual posterior probability associated with every single classifier for predicting the label 'y'. Thus, the numerator in Eq. (4) is a measure of posterior probability for predicting a label 'y' provided whole the classifiers are used. The denominator in Eq. (4) is a measure of posterior probability for predicting a label 'y' provided if every classifier is

```
Data: Pool of classifiers, Test dataset
Result: Final classification of a patient as HC or AD or MCI on the test dataset
/* 1. Initializing the size of pool of classifiers */ ; ;
Initialize int size = Number of classifiers in the ensemble pool ;
/* 2. Use a string array pool of classifiers for storing the pool of classifiers */ ;
str Pool of classifiers[size] = Pool of classifiers ;
/* 3. Initializing a variable i,j equal to zero for storing posterior probabilities */ ;
Initialize int i=j=0 ;
/* 4. Initializing the number of labels */ ;
Initialize int label_size = 3 ;
/* 5. Initialize the three labels ie, AD, MCI, HC */ ;
Initialize label[3]=[AD:0, MCI:1, HC:2] ;
/* 6. This main loop assigns a label out of HC, MCI, AD for each test data dynamically */ ;
while not the end of Test dataset do
    New Set = Best set of classifiers from ensemble pool of classifiers for the given test data ;
    size1 = length(New Set) ;
    /* 6.1 Initializing a multidimensional array posterior probability for storing every pool of classifier's
      posterior probability for the three labels */ ;
    Declare float posterior_probability[size1][label_size] ;
    /* 6.2 Initializing the classifier pointer to zero. This variable is then used for traversing all the pool of
      classifiers */ ;
    Initialize classifier pointer = 0 ;
    /* 6.3 Initially the posterior probability for every pool of classifiers are found out for AD, MCI, HC
      categories. This inner loop is used for calculating the posterior probability for every classifier and
      all the labels */ ;
    while classifier pointer < size do
        Initialize k=0 ;
        while k<=label_size do
            posterior_probability[classifier pointer][k] = Posterior Probability of classifier 'classifier
              pointer' with respect to individual label 'k' ;
            k+=1 ;
        end
        classifier pointer += 1 ;
    end
    /* 6.4 Initializing variable for storing the sum of posterior probability of every classifier */ ;
    Initialize int m=1;
    Declare Posterior_probability_pool_of_every_classifiers[3] ;
    /* This inner loop finds the posterior probability for every label */ ;
    while m<=label_size do
        Initialize Posterior_probability_pool_of_every_classifiers[m]= Posterior Probability using every
          classifier on label 'm' ;
        m+=1 ;
    end
    Initialize int j=0 ;
    /* 6.5 Declare a float variable for storing the lift value for every classifier and the label */ ;
    Declare float Lift_label[size][label_size] ;
    /* This inner loop finds the lift score for every classifiers and the corresponding labels */ ;
    while j<label_size do
        Initialize v=product=1 ;
        while v<size do
            Lift_label[v][j] = Posterior_probability_pool_of_every_classifiers[j] / (Product *
              posterior_probability[v] ;
            product = Product * posterior_probability[v] ;
            v+=1;
        end
        j+=1 ;
    end
    /* 6.6 Declaring a variable for storing the sum of lift scores for a label. This inner loop assigns the
      test data with the highest Lift value */ Initialize float sum_lift[size1] ;
    Initialize U=1 ;
    while U<size do
        For given classifier, store the sum of lift scores with respect to every label in sum_lift[U] ;
    end
    Initialize FinalLabel = Label with maximum lift value ;
    Return FinalLabel for given test data ;
end
```

Algorithm 2: Pseudocode for novel modified Borda count.

used separately. Hence, the ratio of the numerator and denominator in Eq. (4) denotes the association between the combination of classifiers and individual classifiers for predicting the label 'y'.

Equation 5 illustrates the label assigning for a new test data based on the typical Borda count method. Equation 5 illustrates the label assigning for a new test data based on the Lift_new method given in Equation 3.

$$Label(Testdata) = max((label1), (label2), ...(label_n)) \tag{5}$$

$$Label(Testdata) = max(Lift_new(label1), Lift_new(label2), ...Lift_new(label_n)) \tag{6}$$

If there are 'n' labels, Eq. (5) suggests that the final label of the test data is the label with the maximum Lift_new (see equation 4) value. The higher value of Lift_new for a label say 'x' indicates that the combination of the given set of classifiers have higher complementary in predicting that label 'x'. Thus, the complimentary score for the combination of classifiers are considered for the final label assignment rather than considering the sum of posterior probability of single classifiers of every label which is performed on the typical Borda count method.

3.5.3 Pool of classifiers

Following classifiers are used in the ensemble models. They are:

- **Random forest (RF)** RF consists of a set of decision tree (DT) where each DT is a combination of various random observations placed with replacement from the original dataset. Moreover, random features are considered on every DT. The dynamic ensemble models select the set of best performing DT's and use it for testing with the unseen data [78, 79].
- **Bagged decision tree (BDT)** BDT consists of a combination of DT's where each DT consists of a set of a random subset of both features and samples. The samples and features of the DT's in BDT is taken with replacement. The dynamic ensemble models select the set of best performing DT from the BDT and use it for testing with the unseen data [80, 81].
- **Extra tree (ET)** ET is a different version of BDT and RF in a way because rather than selecting random subsets for each DT, the whole training dataset is utilized for creating the DT in ET. The random subsets of the features are selected for each DT in the ET. Moreover, the splitting points of every DT is selected randomly using this method [81, 97].
- **Adaboost** Adaboost is a feedback based DT method where an initial weightage is assigned to every sample in the training set. If there are any misclassified samples in the initial DT, the weights of the misclassified samples are adjusted and send back to the consequent DT's. The dynamic classifier select those combinations of DT's that maximizes and correctly classifies the given sample data [81].
- **Rotation forest (rot-forest)** Each DT in this method constitute both random subset of samples and features. In addition, a Principal Component Analysis (PCA) is applied on each feature set of every DT. Then, the final decision is taken on the principle of RF [82–85].
- **Multilayer Perceptrons (MLP):** MLPs are examples of feed-forward neural networks where the inputs fed into the input layer are shifted into the next layers using activation

functions. The final results are then fed into the last final layer for prediction purposes. Moreover, the error in the final layer is calculated using a loss function. The loss function retraces and reinitializes the weights if the error goes below a low threshold [86, 87]. The bagging procedure is performed on the MLP and the dynamic ensemble model will select those sets of MLP's with greater performance.

- **Support vector machine (SVM)** SVM algorithms find a suitable hyperplane for classifying AD, MCI, and HC patients. The optimal hyperplane is the one that can separate the data points belonging to MCI, HC, and AD patients with greater separability [88, 89]. The best performing SVM's are selected by the dynamic ensemble models for evaluating the performance with the unseen test data.
- **Hetero1** Hetero1 pool of classifiers consists of a combination of heterogeneous classifiers such as Naive Bayes (NB), K-nearest neighbors (KNN), K means algorithms. The Naive Bayes classifier predicts the label for the test data on the basis of Bayesian probability [91]. KNN algorithm finds out the majority label in a nearest neighborhood for the final decision making process [90]. K means will cluster the given samples into three clusters such as AD, MCI, and HC on the basis of the data [92, 93]. Then, a typical Borda majority voting is applied on the prediction results of NB, KNN, and K means for the final prediction of unseen test data.
- **Stacked classifier** Logistic Regression (LR) is applied as a meta-classifier to the hetero1 classifiers for final distinguishing of the AD, MCI, HC patients. The proposed modified and novel Borda count method is applied onto the LR classifier as input for making the final decision [94–96].

Figure 4 depicts the diagram for the modified Borda count voting method.

3.5.4 Evaluation metrics

The important terminologies of a confusion matrix (CM) for AD is as follows:

- **True positives (TP)** It is the count of the total number of predictions that correctly predict the AD as AD itself.
- **False positives (FP)** It is the count of the total number of predictions that incorrectly predicts the non-Alzheimer's (MCI/HC) instances as AD.
- **False negatives (FN)** It is the count of the total number of predictions that incorrectly predicted the AD as non-Alzheimer's instances (MCI/HC).
- **True negatives (TN)** It is the count of the total number of predictions that correctly identify non-Alzheimer's (MCI/HC) instances as non-Alzheimer's (MCI/HC) instances.

The above similar explanations can be given for MCI and HC patients.

The following evaluation metrics are measured from the CM for our experiments.

Balanced classification accuracy (BCA)

It is a measure that combines both the sensitivity and specificity of a classifier. It is a better metric for evaluating the performance of an imbalanced dataset [98, 99]. Equation 1 contains the BCA for a class 'i' equation using sensitivity and specificity.

$$BCA = 1/2 * ((TP/TP + FN) + (TN/TN + FP)) \tag{7}$$

$$BCA = 1/2 * (sensitivity + specificity) \tag{8}$$

Sensitivity or true positive rate

It is the measure of the ability of a classifier to classify correctly a given diagnosis status (MCI/HC/AD) of a patient if they have that diagnosis status. It is the ratio of TP to the sum

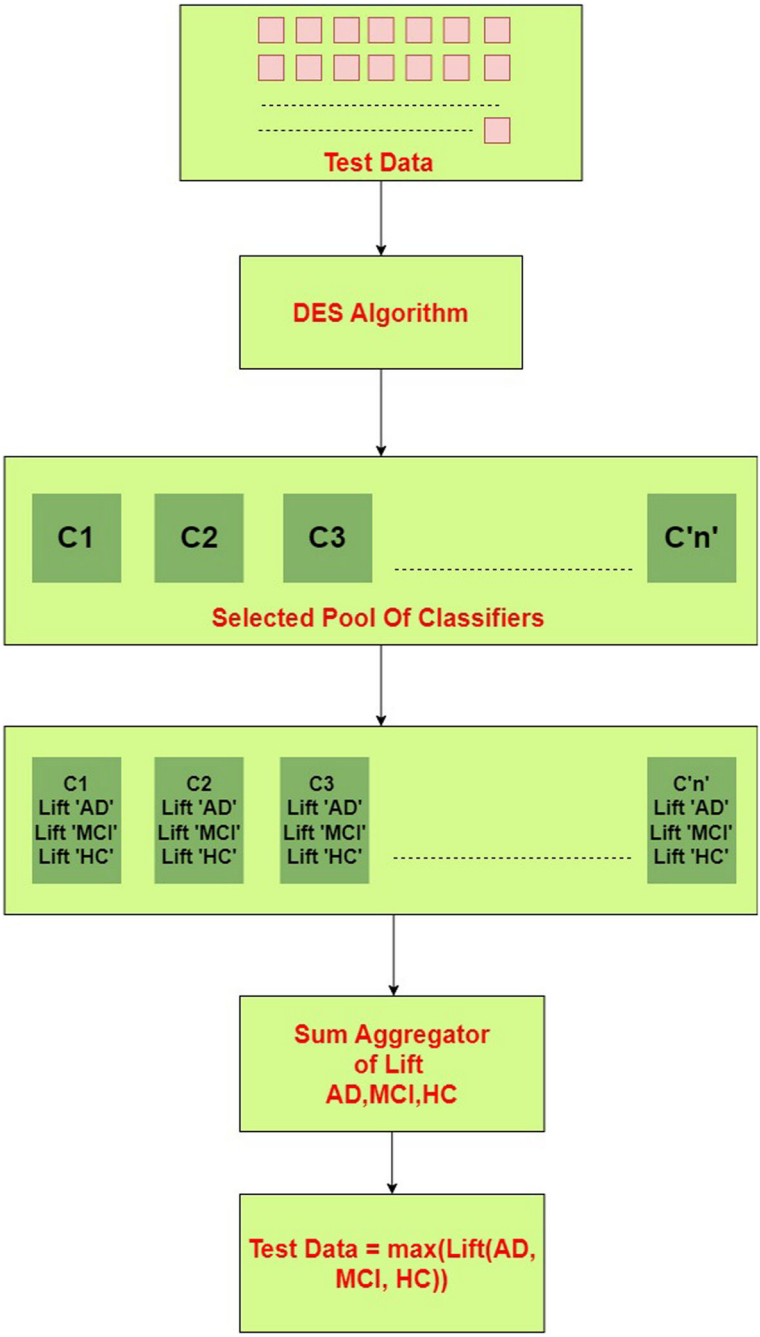

Fig. 4 Diagram for the modified Borda count voting method

Table 2 Count of AD, MCI, HC patients in the training-validation and testing set

Training-validation set	AD	273
	MCI	697
	HC	418
Test set	AD	69
	MCI	175
	HC	105

of TP and FN. Equation 3 contains the sensitivity for a class 'i'.

$$Sensitivity = TP/(TP + FN) \tag{9}$$

Specificity

It is the ratio of TN to the sum of TN and FP. Equation 4 contains the specificity for a class 'i'.

$$Specificity = TN/(TN + FP) \tag{10}$$

4 Experimental results and discussions

This section contains a detailed information about the experimental results and the discussion.

4.1 Implementation details

The experiments are performed on the ADNI-TADPOLE dataset. The dataset consists of 1737 patient's data. The entire dataset is divided into training-validation and testing sets. The 80% and 20% of the data are considered for the training-validation and testing sets, respectively. A stratified 10-fold cross-validation strategy is used for hyper-parameter tuning of the classifiers on the training-validation set. The stratified 10-fold cross-validation is a better technique for maintaining equal proportion of training and validation set during cross-validation stage for imbalanced datasets [100, 101]. The best combination of parameters found from the hyper-parameter tuning is used for the respective models for evaluation with the unseen test dataset.

Training-validation set using stratified 10-fold cross-validation technique The entire training-validation set is divided into 10 equal folds of data. A stratified sampling is performed on every fold of the training set inside the training-validation set for maintaining equal proportion of the labels. The training-validation set consists of 1110 observations. These data are then divided into 10 equal folds. Out of these 10 folds, nine folds are used for training. In each of these nine folds, a stratified sampling is performed for maintaining equal proportion of AD, MCI, and HC. The remaining one fold out of the ten folds are used as a validation set. This process is repeated 10 times until every fold become a training and validation set.

4.2 Hyper-parameter tuning results

The grid search cross-validation technique is used for the hyper-parameter tuning process on the training-validation set. Grid search is one of the most commonly used technique for hyper-

parameter fine-tuning for machine learning tasks [102, 103]. The best combination of hyper-parameters of the classifiers that can maximize the overall BCA is selected for evaluation with the unseen test data. BCA is a better evaluation metric for assessing the performance of a multiclass classification and imbalanced dataset. It can indicate both the true positive rate and true negative rate of the multiclassification [42]. The grid search cross-validation is performed using the Python library sklearn.model_selection.GridSearchCV library.

The hyper-parameter fine-tuning results for the classifiers are as follows:

- The hyper-parameters considered for the tree-based classifiers such as RF, BDT, ET, Adaboost, and Rot-Forest are the number of trees, and the maximum depth of the tree. The Gini index is considered as the splitting criteria for all the tree-based classifiers. The number of trees considered for the classifier while fine-tuning are [100, 1000, 2000, 3000, 4000, 5000, 6000]. The maximum depth of tree considered while fine-tuning are [3, 5, 7, 9, 11, 13, 15].
- The highest BCA of 86% is reported for RF when the number of trees and maximum depth are 3000, 9, respectively. The highest BCA of 87% is reported for BDT and ET when the number of trees and the maximum depth is 2000, 9, respectively. The highest BCA of 87% is reported for Adaboost when the number of trees is 6000 and maximum depth is 9. Similarly, the highest BCA of 85% is reported Rot-Forest when the number of trees is 3000 and the maximum depth is 7. Thus, these tree-based models are built upon these values of the hyper-parameters and then evaluated with the unseen test data. The sklearn.ensemble Python library is used for the implementation of all the homogeneous tree classifiers.
- The grid search is performed on the SVM classifier for the hyper-parameters such as C and Gamma. The possible values considered for C are [0.1, 1, 10, 100] and Gamma are [1, 0.1, 0.01, 0.001, 0.0001]. The best optimal hyper-parameter values are found when the C, Gamma value is 100, 0.0001, respectively, reported a BCA of 72%. The Python Sklearn library is used for implementing the SVM classifier.
- The hyper-parameters fine-tuned for MLP are number of MLP estimators, batch size and epoch size. The learning rate is fixed at a constant value of 0.01. The reason for setting the learning rate as 0.01 is because it is found out to be a commonly used constant rate for complex datasets [46, 47]. The possible values considered for batch size are [200, 400, 600, 800, 1000], epoch size are [50, 100, 150, 200, 250, 300]. The best BCA of 68% is found to be when the learning rate, number of estimators, batch size, epoch size is 0.01, 400, 300, and 200, respectively, during hyper-parameter tuning. Python sklearn.neural_network library is used for implementing the MLP architecture.
- As far as heterogeneous ensemble classifiers are considered, the value of 'k' in KNN and DES algorithms are the same. The dynamic classifier selection algorithms are evaluated using varying values of k such as [1,2,3,4,5,6,7,8,9]. The nearest neighbor value with the highest BCA achieved during the training validation is selected for the testing phase as well. For example, if the highest BCA is reported for KNORAE with k=7 for RF, then the k=7 is considered for evaluating with the test data for KNORAE with RF as the input pool of classifiers.

4.3 Feature selection using LASSO and XGBoost

Both the LASSO and XGBoost techniques are used separately in the study. The reason for using LASSO and XGBoost is that both the methods are widely used by the researchers for finding out the multivariate feature interactions in high dimensional datasets [104–108].

Table 3 Feature importance values using LASSO

Features	LASSO feature value	Modality
Hippocampus volume	113.67	MRI
Middle temporal gyrus volume	97.55	MRI
Area of right pallidium	95.45	MRI
Area of right putamen	93.49	MRI
Area of right rostral middle frontal	90.13	MRI
Area of right thalamus	89.67	MRI
CMRgL of cingulum posterior	87.57	PET
CMRgL of cingulum posterior	80.23	PET
CMRgL of rectus gyrus	55.34	PET
CMRgL of temporal superior lobe	23.57	PET
CMRgL of temporal superior	19.79	PET
CMRgL of temporal superior	17.79	PET
CDRSB	11.9	Cognitive tests
FAQ	7.8	Cognitive tests

The following features are found to be important after using the LASSO feature selection: Hippocampus Volume, Middle Temporal Gyrus Volume, Area of Right Pallidium, Area of Right Putamen, Area of Right Rostral Middle Frontal, Area of Right Thalamus, CMRgL of Cingulum Posterior, CMRgL of Cingulum Posterior, CMRgL of Rectus Gyrus, CMRgL of Temporal Superior Lobe, CMRgL of Temporal Superior, CMRgL of Parietal Superior Lobe, CDRSB, FAQ. The weight of the remaining features are found to be 0. Hence, they are eliminated for the model building. Table 3 illustrates the LASSO feature weights for the selected features.

The following features are found to be important after using the XGBoost feature selection: Hippocampus Volume, Middle Temporal Gyrus Volume, Area of Right Pallidium, Area of Right Putamen, Area of Right Rostral Middle Frontal, Area of Right Temporal Pole, Area of Right Thalamus, Area of Right Transverse Temporal, CMRgL of Hippocampus Right, CMRgL of Para Hippocampal, CMRgL of Cingulum Posterior, CMRgL of Angular Gyrus, CMRgL of Frontal Middle Lobe, CMRgL of Temporal Superior, MMSE, CDRSB, FAQ, Sex. The weight of remaining features are found to be less than 0. Hence, they are removed from the model building process. Table 4 illustrates the XGBoost feature weights for the selected features.

4.4 Performance evaluation on the unseen data using novel Borda count voting method on DES algorithms:

Table 5 contains the comparison of BCA using various DES classifiers with the typical Borda count method and the proposed modified Borda count method using the LASSO regression feature selection. Table 6 contains the comparison of BCA using various DES classifiers with the typical Borda count method and the proposed modified Borda count method using the XGBoost feature selection. Table 7 contains the comparison of sensitivity using various DES classifiers with the typical Borda count method and the proposed modified Borda count method using the LASSO regression feature selection. Table 8 contains the comparison

Table 4 Feature importance values using XGBoost

Features	XGBoost feature value	Modality
Hippocampus volume	8.83	MRI
Middle temporal gyrus volume	8.23	MRI
Area of right pallidium	7.12	MRI
Area of right putamen	6.45	MRI
Area of right rostral middle frontal	6.57	MRI
Area of right temporal pole	6.52	MRI
Area of right thalamus	6.45	MRI
Area of right transverse temporal	6.25	MRI
CMRgL of hippocampus right	5.95	PET
CMRgL of para hippocampal	5.89	PET
CMRgL of cingulum posterior	5.65	PET
CMRgL of angular gyrus	5.45	PET
CMRgL of frontal middle lobe	4.98	PET
CMRgL of temporal superior	4.51	PET
MMSE	3.78	Cognitive tests
CDRSB	3.25	Cognitive tests
FAQ	2.99	Cognitive tests
Sex	2.20	Demography

of sensitivity using various DES classifiers with the typical Borda count method and the proposed modified Borda count method using the XGBoost feature selection. Table 9 contains the comparison of specificity using various DES classifiers with the typical Borda count method and proposed modified Borda count method using the LASSO regression feature selection. Table 10 contains the comparison of specificity using various DES classifiers with the typical Borda count method and the proposed modified Borda count method using the XGBoost feature selection.

The highest BCA of 84% is reported for Rot-Forest after applying the META-DES classifier using the features selected with the LASSO technique (see table 5). The highest BCA of 86% is reported after applying the META-DES classifier on the RF (see Table 6). The BCA using the proposed modified Borda count is either increased or equal to that of the typical Borda count method in the majority of the cases using both the LASSO and XGBoost method (see Tables 5, 6). In general, the range of increase of BCA is from 1 to 6% for most of the ensemble classifiers using the DES algorithms with both the LASSO and XGBoost feature selection methods. The BCA is also reduced using the proposed modified Borda count in some cases. For some classifiers like Adaboost, the BCA using LASSO is reduced from 77 to 76% using the DES-KNN. Similarly, the BCA is reduced from 79 to 77% using KNORAE with the Rot-Forest classifier with the LASSO method. The BCA is also reduced from 73 to 72% using META-DES classifier for SVM classifier with the LASSO method (see Tables 5, 6).

The highest sensitivity of 86% is reported after applying META-DES classifier on Rot-Forest classifier using the features selected with the LASSO technique (see Table 7). The range of sensitivity is increased from 1 to 11% for most of the ensemble classifiers using both the LASSO and XGBoost method (see Tables 7, 8). For Adaboost, the sensitivity is

Table 5 Comparison of BCA using the typical Borda count and the modified Borda count for LASSO on the unseen test data using the ensemble classifiers

Classifier	Method	KNORAE	KNORAU	META-DES	DES-KNN	DESP	KNOP	KLD
RF	Typical	77%	75%	80%	78%	74%	72%	73%
	Proposed	**78%**	**76%**	**82%**	**80%**	**74%**	**73%**	**75%**
BDT	Typical	78%	78%	81%	79%	75%	72%	72%
	Proposed	78%	**80%**	**83%**	**81%**	75%	**73%**	**73%**
ET	Typical	77%	77%	80%	77%	76%	69%	69%
	Proposed	**78%**	**79%**	80%	77%	**77%**	**70%**	69%
Adaboost	Typical	77%	75%	80%	**77%**	77%	69%	69%
	Proposed	77%	75%	**82%**	76%	77%	**71%**	**70%**
Rot-Forest	Typical	**79%**	78%	80%	78%	76%	70%	69%
	Proposed	77%	**80%**	**84%**	**80%**	**82%**	**79%**	**71%**
DT	Typical	69%	70%	69%	71%	69%	71%	69%
	Proposed	**70%**	**71%**	**72%**	**73%**	69%	71%	**72%**
SVM's	Typical	71%	70%	**73%**	74%	69%	71%	69%
	Proposed	71%	**71%**	72%	**75%**	**70%**	**73%**	**70%**
MLP's	Typical	66%	67%	68%	68%	70%	67%	69%
	Proposed	**69%**	**69%**	68%	**69%**	70%	**68%**	69%
Hetero1	Typical	57%	58%	58%	59%	60%	61%	63%
	Proposed	**58%**	**59%**	**61%**	**60%**	60%	**62%**	63%
Stacking	Typical	60%	62%	60%	63%	60%	61%	62%
	Proposed	60%	**65%**	**61%**	**64%**	60%	**62%**	**63%**

Bold values indicate the increase in BCA

decreased from 70 to 69% using ET for the KLD method after executing the LASSO method. A similar result is also observed for Rot-Forest where the sensitivity is decreased from 80 to 78% with LASSO and KNORAE. The sensitivity is also reduced for SVM from 73 to 72% with LASSO and DES-KNN classifier. Further, it is also noted that there is an increase in sensitivity from 1 to 11% with most of the ensemble classifiers for heterogeneous and stacking classifiers (see Tables 7, 8).

The highest sensitivity of 88% is reported after applying META-DES on the RF classifier using the XGBoost selected features (see table 8). A highest sensitivity of 88% is reported after applying META-DES on the RF classifier with the proposed modified Borda Count voting method. There is an observed range of increase in sensitivity from 1 to 9% using the XGBoost selected features. The proposed voting method reported an increase in the sensitivity for every pool of classifiers using the KNORAU classifier (see tables 7, 8).

The highest specificity of 83% is reported for META-DES using BDT classifier with the features selected from LASSO technique after implementing the proposed approach. For homogeneous tree based classifiers, there is an increase in the specificity by 9% for Rot-Forest using KNOP algorithm with the proposed voting algorithm (see table 9). Following results are also observed with the proposed voting algorithm: 1. There is an increase in the specificity of MLP, Hetero classifiers by 10%, 11%, respectively; 2. There is an improvement of 10% in specificity using KNORAE classifier on the MLP. There is an improvement of 11% in specificity on the Hetero1 classifier using DESP classifier (see table 9).

Table 6 Comparison of BCA using the typical Borda count and the modified Borda count for XGBoost and the homogeneous ensemble classifiers on the unseen test data using the ensemble classifiers

Classifier	Method	KNORAE	KNORAU	META-DES	DES-KNN	DESP	KNOP	KLD
RF	Typical	77%	73%	82%	79%	77%	73%	75%
	Proposed	**78%**	**77%**	**86%**	**82%**	**80%**	**75%**	**78%**
BDT	Typical	79%	80%	82%	79%	73%	73%	76%
	Proposed	**80%**	**82%**	**84%**	**83%**	**79%**	**79%**	**80%**
ET	Typical	79%	80%	82%	79%	79%	70%	70%
	Proposed	**80%**	**81%**	**83%**	**80%**	**80%**	**73%**	**73%**
Adaboost	Typical	78%	77%	82%	79%	80%	71%	71%
	Proposed	**80%**	77%	**84%**	**80%**	**80%**	**73%**	**72%**
Rot-Forest	Typical	79%	78%	80%	80%	79%	73%	71%
	Proposed	79%	**80%**	**82%**	80%	**82%**	**75%**	**74%**
DT	Typical	69%	70%	69%	71%	73%	74%	69%
	Proposed	**70%**	**71%**	**72%**	**73%**	**74%**	**75%**	**72%**
SVM's	Typical	71%	70%	74%	75%	70%	74%	69%
	Proposed	**73%**	**74%**	**76%**	**77%**	**73%**	**75%**	**70%**
MLP's	Typical	70%	72%	73%	71%	70%	70%	71%
	Proposed	**73%**	**74%**	**77%**	**72%**	**72%**	**73%**	**75%**
Hetero1	Typical	58%	58%	58%	59%	60%	61%	63%
	Proposed	**59%**	**60%**	**61%**	**60%**	**64%**	**62%**	63%
Stacking	Typical	60%	62%	60%	64%	60%	**64%**	64%
	Proposed	60%	**65%**	**61%**	**67%**	60%	63%	**65%**

Bold value indicates the increase in BCA

The highest specificity of 84% is reported after using RF and Adaboost on the META-DES classifier with the proposed voting method. After using the proposed voting, there is an increase in specificity of 7% on RF using the KNORAU classifier (see Table 10). Moreover, it is also observed that there is also an increase in specificity of 7% on BDT using the DESP classifier (see Table 10).

Overall, there is an increase in the BCA from 1 to 7% after applying the proposed modified Borda count voting method. For most of the ensemble classifiers using both the LASSO and XGBoost method (see Tables 9, 10). However, the specificity is also reduced in some cases using LASSO and an ensemble pool of classifiers such as Adaboost, rotation forest, decision tree, SVM's, MLP, and stacked classifiers. In general, the reduction of specificity is from 1 to 6% in all these cases (see Tables 9, 10).

The reason for an increase in the BCA, sensitivity, and specificity for most of the pool of classifiers with DES algorithms after executing with the modified Borda count is because it considers the association between the posterior probability of prediction of MCI, AD, and HC labels when all the classifiers are used to the posterior probability if the classifiers are used individually. Thus, the proposed modified Borda count method can increase the classification performance for AD, MCI, and HC patients. Moreover, it is also noticed that there is a decrease in the BCA, sensitivity, and specificity in some ensemble pool of classifiers (see Tables 5, 6, 7, 8, 9, 10). This pinpoints the importance of redesigning the modified Borda count equation in such a way that it can also incorporate the various classifier combination associations. Another key observation is that the highest BCA is achieved by tree based

Table 7 Comparison of sensitivity using the typical Borda count and the modified Borda count for LASSO and the homogeneous ensemble classifiers on the unseen test data using the dynamic ensemble classifiers

Classifier	Method	KNORAE	KNORAU	META-DES	DES-KNN	DESP	KNOP	KLD
RF	Typical	77%	73%	79%	78%	74%	74%	**75%**
	Proposed	**78%**	**75%**	**82%**	**80%**	74%	**75%**	73%
BDT	Typical	78%	78%	81%	79%	75%	72%	72%
	Proposed	78%	**80%**	**83%**	**81%**	75%	**73%**	**73%**
ET	Typical	77%	77%	82%	77%	76%	68%	**70%**
	Proposed	**78%**	**80%**	82%	77%	**79%**	68%	69%
Adaboost	Typical	77%	75%	82%	77%	77%	70%	70%
	Proposed	77%	75%	**84%**	**78%**	77%	70%	70%
Rot-Forest	Typical	**80%**	80%	80%	78%	76%	72%	70%
	Proposed	78%	80%	**86%**	**82%**	**82%**	**80%**	**73%**
DT	Typical	69%	70%	69%	71%	69%	71%	69%
	Proposed	**70%**	**71%**	**72%**	**73%**	69%	71%	**72%**
SVM's	Typical	71%	70%	**73%**	74%	69%	71%	69%
	Proposed	71%	**71%**	72%	**75%**	**70%**	**73%**	**70%**
MLP's	Typical	64%	**67%**	**68%**	64%	66%	64%	66%
	Proposed	**66%**	66%	66%	**72%**	**72%**	**72%**	**67%**
Hetero1	Typical	54%	60%	56%	**60%**	**67%**	**65%**	**64%**
	Proposed	**60%**	**67%**	**65%**	58%	56%	63%	58%
Stacking	Typical	58%	**66%**	**65%**	**66%**	**64%**	65%	66%
	Proposed	**62%**	63%	58%	57%	63%	65%	66%

Bold value indicates the increase in sensitivity

classifiers when compared to the MLP's classifiers (see Tables 5, 6, 7, 8, 9, 10). This is a positive factor for physicians as tree based classifiers are also one of the most interpretable models.

4.5 Limitations and future works

The main limitations and future work of the proposed work are as follows:

- The study is conducted on the ADNI-TADPOLE dataset. However, we are planning to conduct the study on various other global datasets belonging to various parts of the globe.
- We utilized the cross-sectional data of the patients (baseline visit) of patients for the study. A study on the longitudinal data could extract many hidden insights for predicting MCI and AD patients. The reason for considering the cross-sectional data is to create an efficient ML model at the baseline visit itself so that the physicians need not wait for consequent visit of a patient. However, it is a challenging task to implement an ensemble model for longitudinal data. This is to be considered as a future work.
- We are also planning to implement advanced deep learning ensembles for extracting features from unstructured data. This study do not extracts the feature from medical images. Hence, the future studies are focused on implementing deep learning for extracting features from unstructured data.

Table 8 Comparison of sensitivity using the typical Borda count and the modified Borda count for XGBoost on the unseen test data using the dynamic ensemble classifiers

Classifier	**Method**	**KNORAE**	**KNORAU**	**META-DES**	**DES-KNN**	**DESP**	**KNOP**	**KLD**
RF	Typical	78%	76%	84%	81%	79%	74%	75%
	Proposed	78%	**77%**	**88%**	**82%**	**82%**	**77%**	**80%**
BDT	Typical	**81%**	82%	82%	79%	75%	74%	76%
	Proposed	80%	**84%**	**84%**	**85%**	**80%**	**77%**	**80%**
ET	Typical	77%	80%	82%	80%	79%	72%	69%
	Proposed	**80%**	**81%**	**85%**	80%	**80%**	**75%**	**75%**
Adaboost	Typical	80%	**77%**	84%	**79%**	81%	72%	72%
	Proposed	80%	76%	84%	77%	**82%**	**75%**	**74%**
Rot-Forest	Typical	79%	78%	80%	80%	77%	75%	71%
	Proposed	**80%**	**80%**	**84%**	**81%**	**82%**	**78%**	**72%**
DT	Typical	**71%**	72%	68%	73%	73%	75%	**71%**
	Proposed	70%	72%	**74%**	**75%**	**75%**	**77%**	68%
SVM's	Typical	71%	68%	75%	77%	72%	**76%**	**72%**
	Proposed	**74%**	**74%**	**77%**	77%	**73%**	75%	69%
MLP's	Typical	72%	72%	75%	72%	70%	72%	73%
	Proposed	**73%**	**75%**	**77%**	**74%**	**73%**	**74%**	73%
Hetero1	Typical	52%	56%	54%	56%	58%	**63%**	**65%**
	Proposed	**61%**	**63%**	**63%**	**57%**	**66%**	56%	57%
Stacking	Typical	**65%**	61%	**66%**	67%	57%	**66%**	62%
	Proposed	60%	**62%**	64%	**70%**	**63%**	58%	**64%**

Bold value indicates the increase in sensitivity

Table 9 Comparison of specificity using the typical Borda count and the modified Borda count for LASSO and the homogeneous ensemble classifiers on the unseen test data using the dynamic ensemble classifiers. Bold value indicates the increase in specificity

Classifier	**Method**	**KNORAE**	**KNORAU**	**META-DES**	**DES-KNN**	**DESP**	**KNOP**	**KLD**
RF	Typical	77%	**77%**	81%	77%	74%	70%	71%
	Proposed	**78%**	74%	**82%**	**80%**	74%	**71%**	71%
BDT	Typical	78%	78%	81%	79%	75%	72%	72%
	Proposed	78%	**80%**	**83%**	**81%**	75%	**73%**	**73%**
ET	Typical	77%	77%	**79%**	77%	76%	67%	68%
	Proposed	**78%**	**78%**	78%	77%	76%	**72%**	**69%**
Adaboost	Typical	77%	75%	78%	**77%**	**77%**	68%	68%
	Proposed	77%	75%	**80%**	74%	76%	**70%**	**70%**
Rot-Forest	Typical	**81%**	76%	80%	78%	76%	68%	**72%**
	Proposed	76%	**80%**	**82%**	78%	**82%**	**77%**	69%
DT	Typical	69%	70%	69%	71%	69%	71%	69%
	Proposed	**70%**	**71%**	**72%**	**73%**	69%	71%	**72%**
SVM's	Typical	71%	70%	**73%**	74%	69%	71%	69%

Table 9 continued

Classifier	Method	KNORAE	KNORAU	META-DES	DES-KNN	DESP	KNOP	KLD
	Proposed	71%	**71%**	72%	**75%**	**70%**	**73%**	**70%**
MLP's	Typical	62%	67%	68%	60%	**74%**	**70%**	**72%**
	Proposed	**72%**	**72%**	**70%**	**66%**	68%	66%	71%
Hetero1	Typical	60%	**56%**	54%	59%	53%	57%	62%
	Proposed	**62%**	55%	**57%**	**62%**	**64%**	**61%**	**68%**
Stacking	Typical	**62%**	58%	55%	60%	56%	57%	58%
	Proposed	58%	**67%**	**64%**	**67%**	**57%**	**59%**	**60%**

Table 10 Comparison of specificity using the typical Borda count and the modified Borda count for XGBoost on the unseen test data using the dynamic ensemble classifiers

Classifier	Method	KNORAE	KNORAU	META-DES	DES-KNN	DESP	KNOP	KLD
RF	Typical	76%	70%	83%	77%	75%	73%	75%
	Proposed	**78%**	**77%**	**84%**	**82%**	**78%**	73%	**76%**
BDT	Typical	77%	78%	82%	79%	71%	72%	76%
	Proposed	**80%**	**80%**	**84%**	**81%**	**78%**	**76%**	**80%**
ET	Typical	**81%**	80%	**82%**	78%	79%	68%	68%
	Proposed	80%	**81%**	81%	**80%**	**80%**	**71%**	**71%**
Adaboost	Typical	76%	77%	80%	79%	**79%**	70%	70%
	Proposed	**80%**	**78%**	**84%**	**83%**	78%	**71%**	70%
Rot-Forest	Typical	**79%**	78%	**80%**	80%	81%	71%	**71%**
	Proposed	78%	**80%**	80%	79%	**82%**	**73%**	70%
DT	Typical	67%	68%	67%	69%	73%	73%	**70%**
	Proposed	**70%**	**70%**	**70%**	**71%**	73%	73%	64%
SVM's	Typical	71%	72%	73%	73%	68%	72%	66%
	Proposed	**72%**	**74%**	**75%**	**77%**	**73%**	**75%**	**71%**
MLP's	Typical	68%	72%	71%	**70%**	70%	68%	69%
	Proposed	**73%**	**74%**	**77%**	67%	**71%**	**73%**	**77%**
Hetero1	Typical	**64%**	54%	**62%**	62%	62%	59%	61%
	Proposed	57%	**57%**	59%	**63%**	62%	**66%**	**67%**
Stacking	Typical	55%	60%	55%	60%	**63%**	**62%**	**66%**
	Proposed	**60%**	**68%**	**58%**	**64%**	57%	52%	63%

Bold value indicates the increase in specificity

- We are planning to conduct a study using various combination of classifiers in the ensemble models. The proposed algorithm only consider the set of all classifiers for the prediction. This drawback will be rectified using an ensemble model that consider various combination of classifiers on the final voting method.

5 Conclusion

This paper proposes a novel and modified version of the Borda count voting method for improving the classification performance of AD, MCI, and HC patients. The ADNI-

TADPOLE dataset is considered for the study. Our results suggest that the proposed method has significantly improved the classification performance in terms of BCA, sensitivity, and specificity for classifying AD, MCI, and HC patients when compared to the typical Borda count method using various ensemble classifiers. The classification results suggest that the application of novel methods in the voting of ensemble classifiers can increase the classifier performance in distinguishing AD, MCI, and HC patients. Thus, this study pinpoints the importance of novel methods in voting algorithms of dynamic ensemble classifiers. It can also be applied to Alzheimer's classification where a slight increase in the classification performance can highly impact the physician's decision-making. Moreover, we are planning toward implementing the ensemble voting methods for feature selection as a future work for distinguishing AD, MCI, and HC patients.

6 Appendix

Appendix contains the hyper-parameter tuning values for homogeneous tree classifiers like RF, BDT, ET, Adaboost, Rot-Forest, SVM, MLP.

Refer Table 11 for the grid search hyper-parameter values for SVM. Refer Table 12 for the grid search hyper-parameter values for RF, BDT, ET, Adaboost, and Rot-Forest. Refer Table 13 for the grid search hyper-parameter values for bagged MLP.

Table 11 BCA using grid search hyper-parameter tuning values for RF, BDT, ET, Adaboost, Rot-Forest

C	Gamma	BCA
0.1	1	66%
1	1	65%
10	1	67%
100	1	66%
0.1	0.1	66%
1	0.1	67%
10	0.1	66%
100	0.1	67%
0.1	0.01	67%
1	0.01	66%
10	0.01	67%
100	0.01	67%
0.1	0.001	66%
1	0.001	66%
10	0.001	66%
100	0.001	68%
0.1	0.0001	69%
1	0.0001	69%
10	0.0001	70%
100	0.0001	**72%**

Table 12 BCA using grid search hyper-parameter tuning values for RF, BDT, ET, Adaboost, Rot-Forest

Number of trees	Max depth	RF	BDT	ET	Adaboost	Rot-Forest
100	3	77%	75%	81%	82%	77%
100	5	76%	76%	81%	82%	77%
100	7	74%	75%	82%	82%	77%
100	9	76%	75%	81%	83%	77%
100	11	77%	76%	82%	84%	78%
1000	3	77%	78%	79%	81%	79%
1000	5	77%	77%	82%	82%	82%
1000	7	78%	78%	81%	84%	84%
1000	9	79%	77%	85%	85%	83%
1000	11	80%	79%	82%	84%	83%
2000	3	80%	81%	83%	84%	82%
2000	5	80%	83%	82%	83%	82%
2000	7	81%	83%	83%	85%	81%
2000	9	86%	**86%**	**86%**	85%	83%
2000	11	78%	83%	83%	85%	83%
3000	3	79%	78%	79%	79%	84%
3000	5	81%	81%	84%	82%	84%
3000	7	82%	81%	85%	83%	**85%**
3000	9	**86%**	82%	84%	84%	83%
3000	11	85%	84%	85%	85%	83%
4000	3	84%	84%	84%	84%	81%
4000	5	85%	83%	86%	84%	78%
4000	7	84%	82%	86%	85%	79%
4000	9	84%	83%	84%	85%	82%
4000	11	82%	83%	86%	84%	80%
5000	3	83%	84%	86%	81%	80%
5000	5	85%	82%	86%	83%	81%
5000	7	84%	83%	85%	84%	79%
5000	9	85%	82%	84%	84%	82%
5000	11	84%	82%	85%	84%	81%
6000	3	83%	79%	84%	85%	79%
6000	5	82%	78%	85%	86%	79%
6000	7	82%	79%	86%	86%	80%
6000	9	83%	80%	86%	**87%**	81%
6000	11	84%	80%	86%	85%	79%

Table 13 BCA for grid search hyper-parameter tuning values for MLP

No. of Estimators	Batch size	Epoch size	BCA
200	50	50	55%
200	50	100	56%
200	50	150	57%
200	50	200	55%
200	100	50	56%
200	100	100	58%
200	100	150	58%
200	100	200	58%
200	150	50	57%
200	150	100	56%
200	150	150	58%
200	150	200	58%
200	200	50	59%
200	200	100	57%
200	200	150	56%
200	200	200	57%
200	250	50	57%
200	250	100	57%
200	250	150	56%
200	250	200	58%
200	300	50	58%
200	300	100	57%
200	300	150	58%
200	300	200	57%
200	50	50	55%
200	50	100	56%
200	50	150	57%
200	50	200	58%
200	100	50	57%
200	100	100	57%
200	100	150	58%
200	100	200	58%
400	150	50	59%
400	150	100	59%
400	150	150	58%
400	150	200	58%
400	200	50	58%
400	200	100	57%
400	200	150	58%
400	200	200	58%

Table 13 continued

No. of Estimators	Batch size	Epoch size	BCA
400	250	50	57%
400	250	100	59%
400	250	150	59%
400	250	200	63%
400	300	50	64%
400	300	100	65%
400	300	150	66%
400	300	200	**68%**
600	200	50	65%
600	200	100	65%
600	200	150	66%
600	200	200	66%
600	250	50	65%
600	250	100	66%
600	250	150	66%
600	250	200	65%
600	300	50	66%
600	300	100	66%
600	300	150	65%
600	300	200	66%
800	50	50	65%
800	50	100	65%
800	50	150	66%
800	50	200	66%
800	100	50	65%
800	100	100	65%
800	100	150	64%
800	100	200	66%
800	150	50	66%
800	150	100	67%
800	150	150	67%
800	150	200	66%
800	200	50	65%
800	200	100	65%
800	200	150	65%
800	200	200	65%
800	250	50	66%

Table 13 continued

No. of Estimators	Batch size	Epoch size	BCA
800	250	100	66%
800	250	150	65%
800	250	200	66%
800	300	50	66%
800	300	100	65%
800	300	150	65%
800	300	200	66%
1000	50	50	66%
1000	50	100	67%
1000	50	150	65%
1000	50	200	65%
1000	100	50	66%
1000	100	100	65%
1000	100	150	66%
1000	100	200	65%
1000	150	50	66%
1000	150	100	66%
1000	150	150	65%
1000	150	200	66%
1000	200	50	65%
1000	200	100	67%
1000	200	150	66%
1000	200	200	65%
1000	250	50	66%
1000	250	100	65%
1000	250	150	67%
1000	250	200	67%
1000	300	50	66%
1000	300	100	65%
1000	300	150	66%
1000	300	200	67%

References

1. Fatima M, Pasha M et al (2017) Survey of machine learning algorithms for disease diagnostic. J Intell Learn Syst Appl 9(01):1
2. Wang P, Zheng X, Li J, Zhu B (2020) Prediction of epidemic trends in covid-19 with logistic model and machine learning technics. Chaos Solitons Fractals 139:110058
3. Amar Lamiaa A, Taha Ashraf A, Mohamed Marwa Y (2020) Prediction of the final size for covid-19 epidemic using machine learning: A case study of Egypt. Infect Disease Model 5:622–634
4. Manogaran G, Vijayakumar V, Varatharajan R, Kumar PM, Sundarasekar R, Hsu C-H (2018) Machine learning based big data processing framework for cancer diagnosis using hidden Markov model and gm clustering. Wireless Personal Commun 102(3):2099–2116
5. Jayasuriya D, Chan J, David S (2020) Machine learning for early breast cancer detection, Big data analytics using small datasets

6. Sarah S, Saira K, Ur RR, Junaid Q, Tehmina A, Farooq AH (2020) Big data analytics enhanced healthcare systems: a review. J Supercomput 76(3):1754–1799
7. Galetsi P, Katsaliaki K, Kumar S (2020) Big data analytics in health sector: theoretical framework, techniques and prospects. Int J Inf Manage 50:206–216
8. Lynch C (2020) World alzheimer report 2019-attitudes to dementia, global survey. In: 2020 Alzheimer's Association International Conference. ALZ
9. Frank Lori , Shubeck Emily, Schicker Melanie, Webb Teresa, Maslow Katie, Gitlin Laura, Hummel Cynthia Huling, Kaplan Edward K, LeBlanc Brian, Marquez Myriam, et al (2020) Contributions of persons living with dementia to scientific research meetings. Results from the national research summit on care, services, and supports for persons with dementia and their caregivers. Am J Geriatr Psychiatry 28(4):421–430
10. Khachaturian ZS (1985) Diagnosis of Alzheimer's disease. Arch Neurol 42(11):1097–1105
11. Citron M (2010) Alzheimer's disease: strategies for disease modification. Nat Rev Drug Discovery 9(5):387–398
12. Jie D, Davis-Plourde Kendra L, Sanaz S, Tully Phillip J, Wanmei W, Caroline P, Pase Matthew P, Himali Jayandra J, Gwen WB, Michael G et al (2020) Antihypertensive medications and risk for incident dementia and Alzheimer's disease: a meta-analysis of individual participant data from prospective cohort studies. Lancet Neurol 19(1):61–70
13. Rausch C, Hoffmann F (2020) Prescribing medications of questionable benefit prior to death: a retrospective study on older nursing home residents with and without dementia in Germany. Eur J Clin Pharmacol 1–9
14. Castellazzi G, Cuzzoni MG, Cotta Ramusino M, Martinelli D, Denaro F, Ricciardi A, Vitali P, Anzalone N, Bernini S, Palesi F et al (2020) (2020) A machine learning approach for the differential diagnosis of Alzheimer and vascular dementia fed by mri selected features. Front Neuroinform 14:25
15. Mather M (2020) How do cognitively stimulating activities affect cognition and the brain throughout life? Psychol Sci Public Interest 21(1):1–5
16. Livingston G, Huntley J, Sommerlad A, Ames D, Ballard C, Banerjee S, Brayne C, Burns A, Cohen-Mansfield J, Cooper C et al (2020) Dementia prevention, intervention, and care: 2020 report of the lancet commission. Lancet 396(10248):413–446
17. Rostamzadeh A, Jessen F (2020) Early detection of Alzheimer's disease and dementia prediction in patients with mild cognitive impairment: Summary of current recommendations. Der Nervenarzt
18. Association A (2019) 2019 Alzheimer's disease facts and figures. Alzheimer's Dementia 15(3):321–387
19. Bessey Laurel J, Art W (2019) Management of behavioral and psychological symptoms of dementia. Curr Psychiatry Rep 21(8):1–11
20. Osland V-MA, Melvaer GL, Clive B, Dag A (2018) Course of neuropsychiatric symptoms in dementia: 5-year longitudinal study. Int J Geriatr Psychiatry 33(10):1361–1369
21. Sarker Iqbal H (2021) Machine learning: Algorithms, real-world applications and research directions. SN Comput Sci 2(3):1–21
22. Fadnavis R, Dhore K, Gupta D, Waghmare J, Kosankar D (2021) Heart disease prediction using data mining. J Phys: Conf Ser 1913:012099
23. Bari AM, Jamil AHM, Mamtaz M, Monirujjaman KM, Aljahdali S, Kaur M, Singh P, Masud M (2021) A comparative analysis of machine learning algorithms to predict Alzheimer's disease. J Healthc Eng
24. Zheng Y, Weng Y, Yang X, Cai G, Cai G, Song Y (2021) Svm-based gait analysis and classification for patients with Parkinson's disease. In: 2021 15th International Symposium on Medical Information and Communication Technology (ISMICT). IEEE, pp 53–58
25. Cesario E, Folino F, Manco G, Pontieri L (2005) An incremental clustering scheme for duplicate detection in large databases. In: 9th International Database Engineering & Application Symposium (IDEAS'05). IEEE, pp 89–95
26. Gupta A (2021) Artificial intelligence approaches to detect neurodegenerative disease from medical records: a perspective. In: Handbook of Research on Disease Prediction Through Data Analytics and Machine Learning. IGI Global, pp 254–267
27. Çağatay BE, Emre S, Seda K (2021) Neurodegenerative disease detection and severity prediction using deep learning approaches. Biomed Signal Process Control 70:103069
28. Mohamed H, Ibtissam A, Ali I, Juan MC de Gea, José LFA (2019) Reviewing ensemble classification methods in breast cancer. Comput Methods Prog Biomed 177:89–112
29. Zhang F, Li Z, Zhang B, Haishun D, Wang B, Zhang X (2019) Multi-modal deep learning model for auxiliary diagnosis of Alzheimer's disease. Neurocomputing 361:185–195
30. Bi X, Li S, Bin Xiao Yu, Li GW, Ma X (2020) Computer aided Alzheimer's disease diagnosis by an unsupervised deep learning technology. Neurocomputing 392:296–304

31. Shaker E-S, Hager S, Radhya S, Tamer A, Riazul ISM, Farman A, Amer E (2021) Alzheimer's disease progression detection model based on an early fusion of cost-effective multimodal data. Futur Gener Comput Syst 115:680–699
32. Zhang D, Wang Y, Zhou L, Yuan H, Shen D, Initiative ADN et al (2011) Multimodal classification of Alzheimer's disease and mild cognitive impairment. Neuroimage 55(3):856–867
33. Illakiya T, Karthik R (2023) Automatic detection of Alzheimer's disease using deep learning models and neuro-imaging: current trends and future perspectives. Neuroinformatics 21(2):339–364
34. Fouad Islam A, Labib Fatma El-Zahraa M (2023) Identification of Alzheimer's disease from central lobe eeg signals utilizing machine learning and residual neural network. Biomed Signal Process Control 86:105266
35. Hedayati R, Khedmati M, Taghipour-Gorjikolaie M (2021) Deep feature extraction method based on ensemble of convolutional auto encoders: application to Alzheimer's disease diagnosis. Biomed Signal Process Control 66:102397
36. Danso Samuel O, Zhanhang Z, Graciela M-T, Ritchie Craig W (2021) Developing an explainable machine learning-based personalised dementia risk prediction model: a transfer learning approach with ensemble learning algorithms. Front Big Data 4:21
37. An Ning, Ding Huitong, Yang Jiaoyun Au, Ting Rhoda Ang, FA (2020) Deep ensemble learning for Alzheimer's disease classification. J Biomed Inform 105:103411
38. Jae YC, Bumshik L (2020) Combining of multiple deep networks via ensemble generalization loss, based on mri images, for Alzheimer's disease classification. IEEE Signal Process Lett 27:206–210
39. Fang X, Liu Z, Mingchang X (2020) Ensemble of deep convolutional neural networks based multi-modality images for Alzheimer's disease diagnosis. IET Image Proc 14(2):318–326
40. Kruthika KR, Maheshappa HD, Initiative ADN et al (2019) Multistage classifier-based approach for Alzheimer's disease prediction and retrieval. Inform Med Unlocked 14:34–42
41. Arco Juan E, Ramírez J, Górriz JM, Ruz M (2021) Alzheimer's Disease Neuroimaging Initiative, et al. Data fusion based on searchlight analysis for the prediction of Alzheimer's disease. Expert Syst Appl 185:115549
42. Muhammed Niyas KP, Thiyagarajan P (2021) Alzheimer's classification using dynamic ensemble of classifiers selection algorithms: a performance analysis. Biomed Signal Process Control 68:102729
43. Mueller Susanne G, Weiner Michael W, Thal Leon J, Ronald CP, Clifford J, William J, John QT, Arthur WT, Laurel B (2005) The Alzheimer's disease neuroimaging initiative. Neuroimaging Clinics 15(4):869–877
44. Marinescu Razvan V, Oxtoby Neil P, Young Alexandra L, Bron Esther E, Toga Arthur W, Weiner Michael W, Barkhof Frederik, Fox Nick C, Klein Stefan, Alexander Daniel C, et al (2018) Tadpole challenge: prediction of longitudinal evolution in Alzheimer's disease. arXiv:1805.03909
45. Marinescu Răzvan V, Oxtoby Neil P, Young Alexandra L, Bron Esther E , Toga Arthur W,Weiner Michael W, Barkhof Frederik, Fox Nick C, Golland Polina, Klein Stefan, et al (2019) Tadpole challenge: accurate Alzheimer's disease prediction through crowdsourced forecasting of future data. In: International Workshop on PRedictive Intelligence In MEdicine. Springer, pp 1–10
46. Jamal KJ, Foo Simon Y (2021) A comparison of machine learning algorithms for diabetes prediction. ICT Express 7(4):432–439
47. Wibowo FW, Wihayati W (2021). Multi-classification of alcohols using quartz crystal microbalance sensors based-on artificial neural network single layer perceptron. In: 2021 International Conference on Artificial Intelligence and Computer Science Technology (ICAICST). IEEE, pp 37–41
48. Hall Mark A (2000) Correlation-based feature selection of discrete and numeric class machine learning
49. Hans C (2009) Bayesian lasso regression. Biometrika 96(4):835–845
50. Chen T, He T, Benesty M, Khotilovich V, Tang Y, Cho H, Chen K, et al (2015) Xgboost: extreme gradient boosting. R package version 0.4-2 1(4):1–4
51. Chen T, Guestrin C (2016) Xgboost: a scalable tree boosting system. In: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, pp 785–794 (2016)
52. Jain A, Nandakumar K, Ross A (2005) Score normalization in multimodal biometric systems. Pattern Recogn 38(12):2270–2285
53. Patro S, Sahu KK (2015) Normalization: a preprocessing stage (2015). arXiv:1503.06462
54. Ng Andrew Y, et al (1997) Preventing" overfitting" of cross-validation data. In: ICML, volume 97. Citeseer, pp 245–253
55. Ghojogh B, Crowley M (2019) The theory behind overfitting, cross validation, regularization, bagging, and boosting: tutorial. arXiv:1905.12787
56. Li X, Zhu D, Levy P (2020) Predicting clinical outcomes with patient stratification via deep mixture neural networks. AMIA Summits Transl Sci Proc 2020:367

57. Santos MS, Soares JP, Abreu PH, Araujo H, Santos J (2018) Cross-validation for imbalanced datasets: avoiding overoptimistic and overfitting approaches [research frontier]. IEEE Comput Intell Magazine 13(4):59–76
58. Ko Albert HR, Sabourin R, Britto AS Jr (2008) From dynamic classifier selection to dynamic ensemble selection. Pattern Recogn 41(5):1718–1731
59. Cruz Rafael MO, Hafemann Luiz G, Robert S, Cavalcanti George DC (2020) Deslib: a dynamic ensemble selection library in python. J Mach Learn Res 21(8):1–5
60. Britto Jr Alceu S, Robert S, Oliveira Luiz ES (2014) Dynamic selection of classifiers-a comprehensive review. Pattern Recogn 47(11):3665–3680
61. Recent Advances and Perspectives, Rafael MOC, Robert S, George DCC (2018) Dynamic classifier selection. Information Fusion 41:195–216
62. Cavalin Paulo R, Robert S, Suen CY (2013) Dynamic selection approaches for multiple classifier systems. Neural Comput Appl 22(3):673–688
63. Beaulieu-Jones Brett K, Moore Jason H, and POOLED RESOURCE OPEN-ACCESS ALS CLINICAL TRIALS CONSORTIUM (2017) Missing data imputation in the electronic health record using deeply learned autoencoders. In: Pacific symposium on biocomputing 2017. World Scientific, pp 207–218
64. Arjun P, Manoj KG (2021) Knowledge discovery from noisy imbalanced and incomplete binary class data. Expert Syst Appl 181:115179
65. Woloszynski T, Kurzynski M, Podsiadlo P, Stachowiak GW (2012) A measure of competence based on random classification for dynamic ensemble selection. Inform Fusion 13(3):207–213
66. Rincy Thomas N, Gupta R (2020) Ensemble learning techniques and its efficiency in machine learning: a survey. In: 2nd international conference on data, engineering and applications (IDEA). IEEE, pp 1–6
67. Dong Y, Zhang H, Wang C, Zhou X (2021) Wind power forecasting based on stacking ensemble model, decomposition and intelligent optimization algorithm. Neurocomputing 462:169–184
68. Berrar D (2018) Bayes' theorem and naive bayes classifier. Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics. Elsevier Science Publisher: Amsterdam, pp 403–412
69. Campagner A, Ciucci D, Cabitza F (2023) Aggregation models in ensemble learning: a large-scale comparison. Inform Fusion 90:241–252
70. Suchithra MS, Pai Maya L (2022) Evaluating the performance of bagging-based k-nearest neighbor ensemble with the voting rule selection method. Multim Tools Appl 81(15):20741–20762
71. Sultan Zia M, Majid H, Arfan Jaffar M (2018) A novel spontaneous facial expression recognition using dynamically weighted majority voting based ensemble classifier. Multim Tools Appl 77(19):25537–25567
72. Madichetty S et al (2020) Identification of medical resource tweets using majority voting-based ensemble during disaster. Soc Netw Anal Min 10(1):1–18
73. Lin W-Y, Tseng M-C, Su J-H (2002) A confidence-lift support specification for interesting associations mining. In: Pacific-Asia conference on knowledge discovery and data mining. Springer, pp 148–158 (2002)
74. Yüksel A (2021) Market basket analysis with association rules. Commun Stat-Theory Methods 50(7):1615–1628
75. Tarmizi OM, Chen Y, Tian L, Fithry SS (2021) Association rule mining for identification of port state control patterns in Malaysian ports. Maritime Policy Manag 48(8):1082–1095
76. Emerson P (2013) The original Borda count and partial voting. Soc Choice Welfare 40(2):353–358
77. Miri M, Dowlatshahi MB, Hashemi A (2022) Evaluation multi label feature selection for text classification using weighted Borda count approach. In: 2022 9th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS). IEEE, pp 1–6
78. Biau G, Scornet E (2016) A random forest guided tour. TEST 25(2):197–227
79. Rigatti Steven J (2017) Random forest. J Insur Med 47(1):31–39
80. Rao JS, Potts WJE (1997) Visualizing bagged decision trees. In: KDD, pp 243–246
81. González S, García S, Del Ser J, Rokach L, Herrera F (2020) A practical tutorial on bagging and boosting based ensembles for machine learning: algorithms, software tools, performance study, practical perspectives and opportunities. Inform Fusion 64:205–237
82. Zhang C-X, Zhang J-S (2008) Rotboost: a technique for combining rotation forest and adaboost. Pattern Recogn Lett 29(10):1524–1536
83. Juan JR, Kuncheva Ludmila I, Alonso Carlos J (2006) Rotation forest: a new classifier ensemble method. IEEE Trans Pattern Anal Mach Intell 28(10):1619–1630
84. Wang L, You Z-H, Xia S-X, Chen X, Yan X, Zhou Y, Liu F (2018) An improved efficient rotation forest algorithm to predict the interactions among proteins. Soft Comput 22(10):3373–3381

85. Sheikhi S, Kheirabadi MT (2022) An efficient rotation forest-based ensemble approach for predicting severity of Parkinson's disease. J Healthc Eng
86. Noriega L (2005) Multilayer perceptron tutorial. Staffordshire University, School of Computing
87. Ramchoun H, Ghanou Y, Ettaouil M, Mohammed AJI (2016) Architecture optimization and training, multilayer perceptron
88. Noble William S (2006) What is a support vector machine? Nat Biotechnol 24(12):1565–1567
89. Pisner Derek A, Schnyer David M (2020) Support vector machine. In: Machine learning. Elsevier, pp 101–121
90. Zhang M-L, Zhou Z-H (2007) Ml-knn: a lazy learning approach to multi-label learning. Pattern Recogn 40(7):2038–2048
91. Leung KM (2007) Naive bayesian classifier. Polytechnic University Department of Computer Science/Finance and Risk Engineering, pp 123–156
92. Hamerly G, Elkan C (2003) Learning the k in k-means. Adv Neural Inform Process Syst 16
93. Ahmed M, Seraj R, Islam SMS (2020) The k-means algorithm: a comprehensive survey and performance evaluation. Electronics 9(8):1295
94. Chen C, Zhang Q, Yu B, Yu Z, Lawrence PJ, Ma Q, Zhang Y (2020) Improving protein-protein interactions prediction accuracy using xgboost feature selection and stacked ensemble classifier. Comput Biol Med 123:103899
95. LaValley MP (2008) Logistic regression. Circulation 117(18):2395–2399
96. Maren ES, Stephen AD, Farhood F, Eric LG (2019) Developing prediction models for clinical use using logistic regression: an overview. J Thorac Disease 11(Suppl 4):S574
97. Désir C, Petitjean C, Heutte L, Salaun M, Thiberville L (2012) Classification of endomicroscopic images of the lung based on random subwindows and extra-trees. IEEE Trans Biomed Eng 59(9):2677–2683
98. Brodersen KH, Ong CS, Stephan KE, Buhmann JM (2010) The balanced accuracy and its posterior distribution. In: 2010 20th international conference on pattern recognition. IEEE, pp 3121–3124
99. Bekkar M, Djemaa HK, Alitouche TA (2013) Evaluation measures for models assessment over imbalanced data sets. J Inf Eng Appl 3(10)
100. Prusty S, Patnaik S, Dash SK (2022) Skcv: Stratified k-fold cross-validation on ml classifiers for predicting cervical cancer. Front Nanotechnol 4:972421
101. Szeghalmy S, Fazekas A (2023) A comparative study of the use of stratified cross-validation and distribution-balanced stratified cross-validation in imbalanced learning. Sensors 23(4):2333
102. Bischl B, Binder M, Lang M, Pielok T, Richter J, Coors S, Thomas J, Ullmann T, Becker M, Boulesteix A-L et al (2023) Hyperparameter optimization: foundations, algorithms, best practices, and open challenges. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 13(2):e1484
103. Sah S, Surendiran B, Dhanalakshmi R, Yamin M (2023) Covid-19 cases prediction using sarimax model by tuning hyperparameter through grid search cross-validation approach. Expert Syst 40(5):e13086
104. Shengli Zhang F, Zhu QY, Zhu X (2021) Identifying dna-binding proteins based on multi-features and lasso feature selection. Biopolymers 112(2):e23419
105. Ueno D, Kawabe H, Yamasaki S, Demura T, Kato K (2021) Feature selection for rna cleavage efficiency at specific sites using the lasso regression model in *Arabidopsis thaliana*. BMC Bioinform 22(1):1–17
106. Wang D, Yang X, Liu X, Ma L, Li L, Wang W (2021) Photoplethysmography-based blood pressure estimation combining filter-wrapper collaborated feature selection with lasso-lstm model. IEEE Trans Instrum Meas 70:1–14
107. Prabha A, Yadav J, Rani A, Singh V (2021) Design of intelligent diabetes mellitus detection system using hybrid feature selection based xgboost classifier. Comput Biol Med 136:104664
108. Dai H, Huang G, Zeng H, Yang F (2021) Pm2.5 concentration prediction based on spatiotemporal feature selection using xgboost-mscnn-ga-lstm. Sustainability 13(21):12071