Biomedical Signal Processing and Control 68 (2021) 102729

Contents lists available at ScienceDirect

Biomedical Signal Processing and Control

journal homepage: www.elsevier.com/locate/bspc

Alzheimer's classification using dynamic ensemble of classifiers selection algorithms: A performance analysis

Muhammed Niyas K. P., Thiyagarajan P. *

Department of Computer Science, Central University of Tamilnadu, Thiruvarur, India

ARTICLE INFO

Keywords:
Alzheimer's
Dynamic Ensemble Selection of Classifiers
Mild Cognitive Impairment
Medical imaging
Cerebro-spinal fluid

ABSTRACT

Alzheimer's is a type of severe cognitive impairment where an individual cannot do their daily day-to-day activities. It is a challenging task to find out the Alzheimer's and Mild Cognitive Impairment patients. This study aims to compare the performance of the state of the art Dynamic Ensemble Selection of Classifier algorithms for classifying healthy, Mild Cognitive Impairment, and Alzheimer's disease participants at the baseline stage itself using multimodal features. The data used in the study is from Alzheimer's Disease Neuroimaging Initiative-TADPOLE dataset. The medical imaging, Cerebro-spinal fluid, cognitive test, and demographics data of the patients at the baseline visits are considered for the prediction purpose. The performance of the state-of-the-art Dynamic Ensemble of Classifier Selection algorithms is compared using these features in terms of Balanced Classification Accuracy, Sensitivity, and Specificity. The most commonly used pool of Machine Learning classifiers is used as the input for Dynamic Ensemble of Classifier Selection algorithms. Moreover, the performance of the pool of Machine Learning classifiers without using the Dynamic Ensemble Selection of Classifiers algorithms are also compared. The performance metrics such as Balanced Classification Accuracy, Sensitivity, and Specificity are increased after using the Dynamic Ensemble of Classifier Selection algorithms on most of the pool of classifiers for classifying healthy, Alzheimer's, and Mild Cognitive Impairment patients is promising.

1. Introduction

Alzheimer's Disease (AD) is the sixth most leading cause of death among the aged population in the United States [1]. It is also the most common dementia among the aged population around the globe [1,2]. According to a report by Alzheimer's disease facts and figures 2018, it is estimated that the 7,00,000 people whose age is greater than 65 will have AD when they die [1]. Moreover, the report says that most of the aged people are dying due to the complications made by AD [1]. Thus, a health care practitioner needs to find an AD patient at the baseline visit. Finding the patients who are likely to develop severe AD helps the physician in designing effective treatment strategies for reducing the rate of progression of cognitive destruction [3]. However, it is a challenging task to predict the persons who are likely to have Alzheimer's in their future life or not [4,5]. This is where an evidence-based approach with Machine Learning (ML) helps in finding future AD patients using various quantitative biomarker and cognitive measurements [5]. The quantitative biomarker data of a person captured using various medical imaging techniques such as Magnetic Resonance Imaging (MRI), Positron Emission Tomography (PET), and lab data such as Cerebro-Spinal Fluid (CSF) along with cognitive measurements, age, sex, and education are of great importance in predicting the patients who might have AD in their future life [5,6]. ML techniques are used to predict the AD patients using these multimodal data and assist the physician in appropriate decision making [5].

Cognitive impairment is very common among the aged population around the globe [1]. The intensity of cognitive impairment varies from mild to severe [1,7]. Patients with Mild Cognitive Impairment (MCI) face difficulty in doing complex logical activities. However, MCI patients can do daily life activities without depending on others. But this is not the case for severe cognitive impairment. Severe cognitive impaired patients cannot even do simple daily day-to-day activities like bathing, brushing which makes their lives miserable [1,8]. AD is such a severe impairment that needs special attention. It is to be noted that some MCI patients might convert to AD in the future [8]. However, some MCI patients remain as mild cognitively impaired without converting to AD [9]. Researchers face a challenging task in detecting AD from MCI [10]. This is because there are minute variations that help in distinguishing

* Corresponding author.
E-mail address: thiyagu.phd@gmail.com (T. P.).

https://doi.org/10.1016/j.bspc.2021.102729
Received 21 September 2020; Received in revised form 17 April 2021; Accepted 7 May 2021
Available online 15 May 2021

Table 1
Comparison of previous works on ADNI-TADPOLE.

		[14]	[15]	[16]	[17]	This work
Feature set	Length	6	12	6	13	**161**
	Modalities	MRI, cognitive test, genetic	MRI, genetic cognitive test, age CSF	MRI, PET cognitive test, CSF	genetic, MRI, age, sex	**MRI**, **PET**, **CSF**, **age**, **sex**, **education**,
Feature type	Crosssectional					**Yes**
	Longitudinal	Yes	Yes	Yes	Yes	
Prediction algorithms	ML	Yes	Yes			**Yes**
	DL			Yes	Yes	**Yes**

The bold values in the tables are the results that needs to be highlighted.

AD from MCI [10]. Moreover, the exact reasons for the occurrence of AD are not well-defined [11]. Identifying the AD, MCI, and Healthy Controls (HC) helps the physician to design a proper and effective medication strategies for dealing with them separately [1,12,13]. For instance, identifying an MCI person helps the physician to design the required medication for dealing with that. Similarly, if a physician can identify a severe AD patient, then he or she can design a more well-planned medication strategy for dealing with severe cognitive impairment. In short, the medication strategies can be designed based on their severity of cognitive impairment. This has mainly two benefits, 1. A more precise treatment can be given for an individual based on the severity of their cognitive impairment. For example, the physicians can provide separate treatments for MCI and AD patients, 2. The unnecessary health care costs can be avoided by designing an optimal medication strategy based on the severity of cognitive impairment [1,12]. For example, if the physician can distinguish AD and MCI, then they can design a more focused and precise treatment strategy to counter the progression of severe cognitive impairment like AD which is more expensive than MCI. Likewise, the physician can also design the required and optimal treatment to handle MCI patients that is cheap when compared to AD treatments [12].

1.1. Related works

There were many experiments conducted with multimodal data for classifying AD, MCI, and HC on the Alzheimer's Disease Neuroimaging Initiative-TADPOLE (ADNI-TADPOLE) dataset [14–17]. The study conducted by the researchers in [18] using 6 MRI biomarkers reported with a Multi Area Under the Curve (MAUC) of 0.73. There was an improvement of results using Random Forest (RF) [14,15]. In the study conducted with RF for the classification of AD with longitudinal multimodal features ($n = 19$) reported with a Balanced Classification Accuracy (BCA) and MAUC of 73% and 0.82 respectively [14]. In a similar study using RF, the researchers predicted the AD using 12 multimodal longitudinal features has achieved a BCA of 86% [15]. This slight improvement in the result was achieved using a mixed-effects model for feature selection from the longitudinal data [15]. However, the studies that performed data imputation using Recurrent Neural Networks (RNN) achieved better prediction results [16,17]. The study conducted by the researchers in [16] used SVM with only 6 multimodal features reported with a BCA of 86%. Similarly, the study used RNN for less number of features ($n = 13$) achieved a slightly better improvement of BCA and MAUC of 86% and 0.866 respectively. All the studies were conducted on multimodal features selected from the ADNI-TADPOLE dataset using subsequent longitudinal visit data.

Researchers applied ensemble classifiers for classification tasks in biomedical field with good classification results [19,20]. Antonakakis et al. [19] used an ensemble of Support Vector Machine (SVM) and K Nearest Neighbor (KNN) for classifying mild Traumatic Brain Injury (mTBI) patients [19,20]. These were the first studies that used ensemble classifiers in the biomedical field. Researchers also used Dynamic Ensemble of Classifier Selection (DES) algorithms because it dynamically find a classifier for each test data separately [21]. That is why researchers used DES for finding the patterns from complex domains like biomedical and image segmentation [21,22]. Rather than finding a single classifier for the whole dataset, DES techniques find an ensemble of classifiers for each test data dynamically which makes it more efficient and flexible [21,22]. Consequently, this motivated us to the experimentation with DES algorithms for predicting MCI, AD, and HC using baseline multimodal data. DES techniques reported good performances in UCI datasets such as PIMA, Wisconsin Breast Cancer, Wine, Iris, Yeast, and Image segmentation [21,23]. The flexibility of the DES models in selecting appropriate classifier for each test data using the local performance of the pool of classifiers makes it more efficient in dealing with complex datasets consists of multiple classes [21,24]. Hence, our study focused on experimenting using DES algorithms for improving the classification performance of MCI, AD, and HC participants using baseline multimodal data. The classification improvement of AD, MCI, and HC is important for a physician because even a small improvement in classification performance helps a physician in effective decision making regarding treatment plans that can save and improve the quality of human lives [1].

We conducted the experiments on the popular Alzheimer's database namely ADNI-TADPOLE dataset. The objective of our study is to report the results on the ADNI-TADPOLE dataset by using a larger feature set than the previous works, by experimenting on MRI, PET, CSF, cognitive tests, age, sex, and education features with only considering baseline visit data and check if the performance of the ML classifiers are improved for AD, MCI, and HC classifications using DES algorithms. The experiments are performed using advanced state of the art DES models whose input is the pool of classifiers consists of machine learning and deep learning models. A comparison of experimental settings on ADNI-TADPOLE with related works is given in Table 1.

Consequently, We performed a comparative methodology in which the baseline multimodal data such as MRI, PET, CSF biomarkers, cognitive tests, age, sex, and education data available in the ADNI-TADPOLE dataset for predicting the performance of the AD, MCI, and HC classification using the state of the art 6 DES algorithms such as K-Nearest Oracle Elimination (KNORAE), Meta-Learning for Dynamic Ensemble Selection (META-DES), Dynamic Ensemble Selection Performance (DESP), K-Nearest Oracle Union (KNORAU), Dynamic Ensemble Selection-K Nearest Neighbor (DES-KNN), and Dynamic Ensemble Selection for Multi Imbalanced datasets (DES-MI) whose input is the 8 ensemble pool of classifiers. The 8 pool of classifiers are as follows:

- Homogeneous ensembles: Bagged Decision Tree (BDT), Random Forest (RF), Extra Trees (ET), Adaboost, and Bagging Multi Layer Perceptron (BMLP).

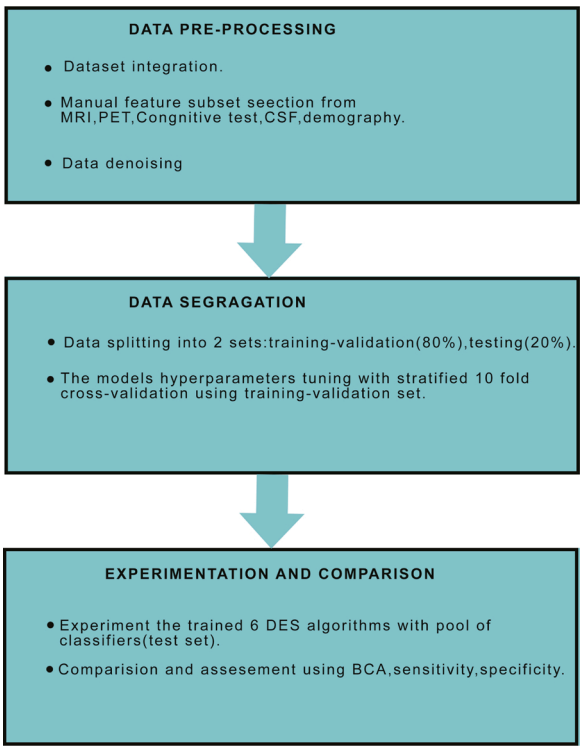


Fig. 1. The overall processing pipeline of the methodology.

- Heterogeneous ensembles: Pooling, bagging, and stacking ensemble of ML classifiers such as SVM, Naive Bayes (NB), Logistic Regression (LR), and K Nearest Neighbor (KNN).

Further, the performance of the ensemble pool of classifiers without using the DES algorithms are also compared. The performance of the models is assessed in terms of BCA, Sensitivity, Specificity. The organization of the paper is as follows: Section 2 contains materials and methods, Section 3 contains experimental results and discussions, Section 4 contains study limitations, and Section 5 contains the conclusions.

2. Materials and methods

A comparative methodology is used in the paper by analyzing the performance of the state of the art 6 DES algorithms with baseline multimodality features such as MRI, PET, CSF, cognitive tests, and demography in terms of BCA, Sensitivity, and Specificity. Fig. 1 contains the overall processing pipeline of the methodology. The overall processing pipeline of the methodology in the sequential order is as follows.

- Data pre-processing
  a. Dataset integration
  b. Manual feature subset selection from MRI, PET, CSF, cognitive tests, age, sex, and education data.
  c. Data denoising involves the operations of cleaning the data for processing.
- The segregation of data into training-validation and test sets. The fine-tuning and validation of the hyper-parameters of the models are performed using a stratified 10 fold cross-validation strategy on the training-validation set.
- Experimenting the 6 DES algorithms namely KNORAE, META-DES, DESP, KNORAU, DES-KNN, and DES-MI using a baseline feature set for classifying AD, MCI, and HC on the test set. The performance of the above algorithms is compared and analyzed for the pools of classifiers in terms of BCA, Sensitivity, and Specificity. Further, the performance of the pool of classifiers without using the DES algorithms are also compared and analyzed.

2.1. Dataset description

The experiments are conducted on ADNI-TADPOLE, a global challenge dataset initiative for the prediction of AD using multimodal longitudinal data started in the year 2017 [25]. The ADNI is launched in 2003 as a public-private partnership, led by principal investigator Michael W. Weiner for conducting a study on investigating a data-driven approach for finding the early onset of AD with MRI, PET, CSF, genetic biomarkers, and other clinical assessment data [26]. The ADNI-TADPOLE dataset is selected because it is one of the most benchmarking datasets for ML researchers consists of diverse modality data that are medically relevant in explaining the progression of AD [26]. Moreover, it is the largest openly available dataset for AD researchers that consists of 1737 individuals data in it [26]. The dataset consists of 1737 individual patient's follow-up data collected by ADNI studies including ADNI1, ADNI2, and ADNI3. The associated MRI, PET, CSF, age, sex, and education data with each patient are collected over various time points. There are 3 standard datasets in ADNI-TADPOLE for AD diagnosis predictions:

- **D1-TADPOLE standard training dataset:** This is a training dataset consists of the follow-up data of every individual who have at least two separate visits to the ADNI study. The MRI, PET, CSF, cognitive tests, demographics, and other clinical assessment data in each of the visits are collected using ADNI's standard data-processing pipelines [25]. The standard training dataset is created by merging the ADNIMERGE spreadsheet (consists of cognitive tests, age, sex, and demographics information) and the MRI, PET, CSF spreadsheets of the ADNI [26].
- **D2-TADPOLE standard prediction dataset:** The TADPOLE standard prediction set contains the whole longitudinal data (including all the follow-up time point data) of the previous ADNI participants as in the D1 dataset. The data processing pipelines and features collected for the individuals are the same as in the D1 dataset [25].
- **D3-Cross sectional prediction dataset:** This is a cross-sectional data consists of demographics, derived MRI volumes, and cognitive tests of the final visit data of the participants as in D2 [25].

The whole data is available on the ADNI-TADPOLE website[1] and the full explanation is given in the paper [25].

2.2. Data pre-processing

This section contains the data pre-processing performed on the ADNI-TADPOLE dataset. The data pre-processing is performed sequentially in the following order: dataset integration, feature subset selection, and data denoising.

2.2.1. Dataset integration

The integration of various biomarker quantitative measurements collected through MRI, PET, CSF, cognitive tests, and demographics data is the initial step. For this, we utilize the data from TADPOLE D1_D2 (which is created after merging the data from D1 and D2 datasets) [25]. The TADPOLE D1_D2 consists of MRI, PET, CSF, cognitive tests, and demographics data of the study participants. These are collected from various ADNI spreadsheets [26]. This dataset contains the biomarker measurements of 1737 participants that are routinely collected over various visiting time points. Thus, by using the ADNI-TADPOLE D1_D2

[1] https://ida.loni.usc.edu/pages/access/studyData.jsp?categoryId=43&subCategoryId=94.

Table 2
Summary statistics of the ADNI-TADPOLE D1_D2 dataset.

Labels	Data	Value
HC	# Participants	523
	# Male	253
	# Female	270
	# Age<55 (years)	NIL
	# Age>55 and Age<65 (years)	15
	# Age>65 and Age<75 (years)	279
	# Age>75 and Age<85 (years)	209
	# Age>85 and Age<95 (years)	20
	Age range of males (years)	[59.9,90.1]
	Age range of females (years)	[56.2,89.6]
MCI	# Participants	872
	# Male	515
	# Female	357
	# Age<55 (years)	14
	# Age>55 and Age<65 (years)	148
	# Age>65 and Age<75 (years)	353
	# Age>75 and Age<85 (years)	317
	# Age>85 and Age<95 (years)	40
	Age range of males (years)	[54.4,91.4]
	Age range of females (years)	[55.0,88.4]
AD	# Participants	342
	# Male	189
	# Female	153
	# Age<55 (years)	NIL
	# Age>55 and Age<65 (years)	44
	# Age>65 and Age<75 (years)	113
	# Age>75 and Age<85 (years)	153
	# Age>85 and Age<95 (years)	32
	Age range of males (years)	[55.9,90.3]
	Age range of females (years)	[55.1,90.9]

dataset in the initial step, we integrated the longitudinal quantitative biomarker measurement data from medical imaging modalities (MRI, PET), CSF, cognitive tests, and demographics of the study participants. However, the ADNI-TADPOLE D3 dataset is excluded from our study because it has only the final visit data of the study participant's and the focus is on the baseline visits of the study participant's [25].

It is observed that the count of male participants is greater than that of the females for HC in both the D1_D2 dataset. Moreover, the majority of participants in the MCI and AD category come under the age group between 65 and 85 years in both the D1_D2 dataset for every labels (see Table 2). Table 2 contains the demography statistics of the ADNI-TADPOLE D1_D2 dataset.

2.2.2. Feature subset selection

The next step after dataset integration is to manually select and segregate features that belong to medical imaging modalities, CSF, cognitive tests, and demographics. The advanced progression of AD can be assessed using the quantification of biomarkers that are captured using various medical imaging modalities such as MRI and PET. The various types of data such as cognitive tests (measuring the cognitive memory), CSF, age, sex, and education [25] are also vital in detecting AD [25]. The main criterion for manually selecting and segregating into the different feature categories are due to the different data acquisition techniques that are responsible for acquiring various types of information for the AD.[2] The MRI biomarkers are derived from the Free Surfer longitudinally processed Region of Interest's (ROI). The PET biomarkers used in the dataset are Fluorodeoxyglucose (FDG), Florbetpair or AV45, and Cerebral Metabolic Rate for Glucose (CMRgl). The CSF biomarkers used in the dataset are: Amyloid-beta, Tau, and P-Tau [25]. The type of information captured for AD by the various data acquisition techniques are as follows:

- **MRI**: MRI imaging modality is used to capture the structural and physiological changes in the brain [25]. Appendix A.1 contains the MRI features used in our study.
- **PET**: PET imaging modality is used to understand cell metabolism inside the brain [25]. Appendix A.2 contains the PET features used in our study.
- **Cognitive tests**: The Cognitive tests are useful in understanding the cognitive behaviors of an individual [25]. Appendix A.3 contains the cognitive tests used in our study.
- **CSF**: CSF is a fluid that is seen in the brain and spinal cord of a person [25]. Appendix A.4 contains the CSF features used in our study.
- **Demographics**: Age, sex. and education are used in our study.

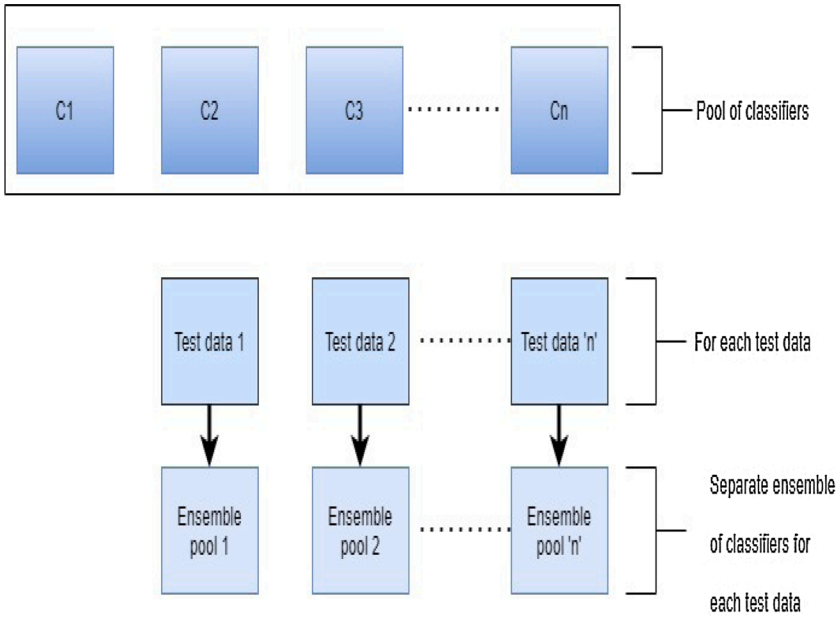


Fig. 2. General DES algorithm.

[2] https://tadpole.grand-challenge.org/Data/.

2.2.3. Data denoising

Following data denoising operations are performed:

- **Baseline visit data:** The baseline visit data are only considered for our study. The main reason for choosing baseline visit data is to create the prediction models by using only the first visit data and assist the physician in decision making without considering the next consequent visits of the patient.
- **Handling of missing values**: The features with more than 50% of the missing data are eliminated for the study since the same approach is used in the previous works [27,28], and [29]. MissForest data imputation algorithm is used for substituting the missing values of the features, implemented with the Python MissForest package [30–32]. There are mainly three reasons for choosing the MissForest technique for data imputation: 1. It does not require any explicit hyperparameter tuning, 2. Handles both categorical and numerical data easily, 3. Figuring out the non-linear correlation interaction between the features [30], [31].

After performing the data denoising, the following data preparation operations are performed before processing data with the ML classifiers.

- **Changing to appropriate labels:** For our study, Stable Mild Cognitive Impairment (SMCI), Cognitively Normal (CN) are considered as HC and Early Mild Cognitive Impairment (EMCI), Late Mild Cognitive Impairment (LMCI) are considered as MCI respectively [25].
- **Changing to a unified data type format:** The values of the feature sets are converted to an unified numeric data type using the Pandas library of Python [33].

2.3. DES algorithms

DES algorithms dynamically find out the combination of classifiers for each test data. The input for the DES algorithm consists of a pool of classifiers and a region of competence defined for each test data. The region of competence is the k-nearest neighbor training data of the test data. Then, the prediction for each training data in the region of competence is performed using all the combinations of classifiers. If the prediction ability of any of the combination of classifiers is satisfied in the region of competence, then the resultant set of classifiers are assigned to predict the given test data [34]. Fig. 2 illustrate the general definition of the DES algorithm.

The 6 states of the art DES algorithms are used for analysis namely, KNORAE, META-DES, DESP, KNORAU, DES-KNN, and DES-MI.

KNORAE This DES algorithm finds those set of classifiers from the pool of classifiers that correctly classifies all of the K nearest neighbors for a given test data in the training set. The ensemble of such chosen classifiers is assigned and made eligible for voting for the classification of the test data (the majority voting rule is used for the prediction in KNORAE). In other words, the algorithm eliminates the classifiers that incorrectly classify any one of the data in the neighborhood of the test data [21]. In case, if no such classifier is found, then the algorithm keeps on reducing the value of nearest neighbors and start the search for at least one classifier that classify all the training samples in the neighborhood of test data [21].

META-DES This DES algorithm approaches the dynamic classification as a meta-problem [35]. The meta-problem for this algorithm is to decide whether the given classifier from the pool of classifiers is competent enough to classify the given test [35]. There are mainly two steps involved in solving this meta-problem: 1. Finding the meta-features for all the classifiers in the pool: There are 4 types of meta-features namely, (a) posterior probability for each label (the probability that the training data in the region of competence belong to the output label), (b) The overall Local Accuracy of the classifier in the region of competence, (c) Neighbors Hard Classification (a vector of size 'n' is created, where n is the number of training samples in the region of competence and if the classifier correctly the sample in the region of competence, if the vector is set to 1, otherwise 0. Thus, a vector of size 'n' is returned.), (d) Classifier's Confidence (the perpendicular distance between the input sample and the decision boundary of the classifier), 2. Meta classifiers predict whether the given classifier is capable of giving the correct prediction for the given test data using the meta-features [35]. Hence, all the classifiers that are chosen by the meta classifiers are selected for the ensemble of classifier set for the given test data [35].

DESP This DES algorithm eliminates the incompetent classifiers in the pool of classifiers by comparing the performance of the classifiers with a random classifier [36]. The performance of the random classifier is $1/M$, where M is the number of classes in the dataset (see the explanation in [36]). The dynamic selection of the classifiers is performed for each test data by comparing the performance of the classifiers with the random classifier in the defined neighborhood of the test data [36]. If the performance of the classifier is greater than that of the random classifier, then it is eligible for the selection in the ensemble of classifiers for the given test data. If no classifier from the pool is selected, then the whole pool of classifiers are chosen for the given test data [23].

KNORAU This algorithm finds all the set of classifiers from the pool of classifiers that correctly classifies any one of the K nearest neighbors of a given test data in the region of competence [21]. In other words, all those classifiers that correctly identifies the label of at least any one of the training data in the K nearest neighborhood of the test data are combined to form an ensemble for the given test data (the majority voting rule is used for the prediction in KNORAU). However, the number of votes that the classifiers in the ensemble depend on the number of correctly classified training set samples in the K neighborhood [21].

DES-KNN This proposed DES algorithm in [37] use both diversity and Accuracy of the classifiers as the metric for choosing ensembles. Initially, the top 'n' accurate classifiers in the region of competence of a test data are found out. Then, the most 'm' diverse among the 'n' accurate classifiers are found out [37] and chosen as the ensemble of classifiers for the given test data. The diversity among the classifiers is found out by the metric double-fault measure, which counts the common misclassified cases of the classifier [37]. If the double fault measure is large, then the diversity of the classifiers are large [37]. For our study, the percentage of classifiers selected from the pool for Accuracy and diversity are 50% and 30% respectively. These values of Accuracy and diversity are selected based on better performance results from previous studies [37,38]. The study conducted by the researchers in [37,38] used the most diverse 30% classifiers among the high accurate 50% classifiers with better results has motivated for choosing these values for our study.

DES-MI This DES algorithm finds the appropriate classifiers for each test data when the dataset is imbalanced [39]. The algorithm finds the proportion of the samples of each label in the region of competence and then effectively design a weightage for the samples of each label separately [39]. In other words, different weightage is given to each label based on the proportion of each label in the region of competence. The higher weightage is given to the label with low proportion and vice versa. Thus, the algorithm balances the under-representation of the minority classes in the region of competence [39].

The explanation for the implemented algorithms along with the reference papers are given in the link.[3]

2.3.1. Pool of classifiers

The following pool of classifiers are used for the DES algorithms: homogeneous ensembles such as Bagged Decision Tree (BDT), Random Forest (RF), Extra Tree (ET), Adaboost, Bagging Multi-Layer Perceptron (BMLP) and heterogeneous ensembles consist of pooling, bagging, and stacking ensemble of ML classifiers constituting Naive Bayes (NB), Support Vector Machine (SVM), K Nearest Neighbor (KNN), Logistic

[3] https://github.com/scikit-learn-contrib/DESlib.

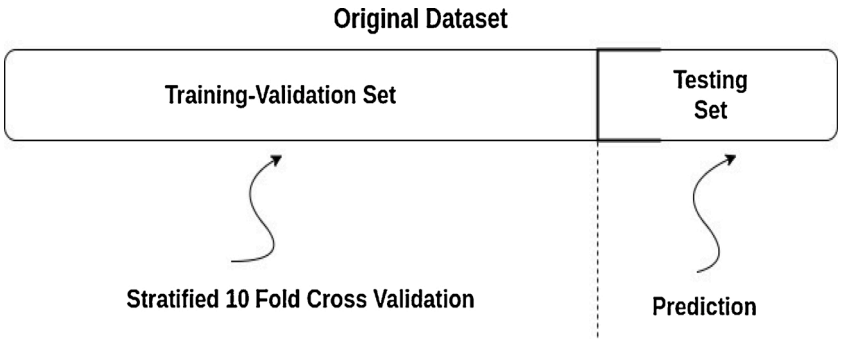


Fig. 3. Data segregation performed for the experiments.

Regression (LR). Appendix B contains detailed information about the pool of classifiers.

2.4. Evaluation criteria

The BCA, Sensitivity, and Specificity are used to evaluate the performance for classifiying MCI, AD, and HC. The performance metrics are evaluated for all the diagnosis labels separately. The overall value of the performance metrics is the average of the performance metrics of the individual diagnostic labels.

The explanation of the terms used in all the performance metrics for Alzheimer's diagnosis label[4] is as follows:

TP (True Positive): It is the count of the total number of predictions correctly identifying the Alzheimer's instances as Alzheimer's.

TN (True Negative): It is the count of the total number of predictions that correctly identifying non-Alzheimer's instances (MCI/HC) as non-Alzheimer's (MCI/HC).

FP (False Positive): It is the count of the total number of predictions that incorrectly predicted the non-Alzheimer's (MCI/HC) instances as Alzheimer's.

FN (False Negative): It is the count of the total number of predictions that incorrectly predicted the Alzheimer's instances as non-Alzheimer's instances (MCI/HC).

The explanation of the evaluation metrics is as follows:

- BCA:

BCA is a metric used for measuring the Accuracy of imbalanced datasets. It is the average of the sum of the Sensitivity and Specificity of that label. The BCA of class 'i' is given in Eq. (1).

$$\mathrm{BCA}_i = 1/2 * [(\mathrm{TP}/\mathrm{TP} + \mathrm{FN}) + (\mathrm{TN}/\mathrm{TN} + \mathrm{FP})] \tag{1}$$

Then, the overall BCA for all the classes is the mean of the individual BCA of each class. Eq. (2) contains the overall BCA.

$$\mathrm{BCA} = 1/L * \sum_{i=1}^{L} \mathrm{BCA}_i \tag{2}$$

Where TP is the True Positives, FN is the False Negatives, TN is the True Negatives, FP is the False Positives, and L is the number of classes in Eqs. (1) and (2).

- Sensitivity

Sensitivity for class 'i' is given in Eq. (3):

$$\mathrm{Sensitivity}_i = \mathrm{TP}/(\mathrm{TP} + \mathrm{FN}) \tag{3}$$

Eq. (8) calculates the Sensitivity for a single class by finding the ratio between the True Positives to the sum of the True Positives and False Negatives in the confusion matrix for that class.

The overall Sensitivity is the mean of individual Sensitivity of each classes. It is given in Eq. (4):

$$\mathrm{Sensitivity} = 1/L * \sum_{i=1}^{L} \mathrm{sensitivity}_i \tag{4}$$

Where TP is the True Positives, FN is the False Negatives, L is the number of classes in Eqs. (3) and (4).

- Specificity:

Specificity for class 'i' is given in Eq. (5):

$$\mathrm{Specificity}_i = \mathrm{TN}/(\mathrm{TN} + \mathrm{FP}) \tag{5}$$

Eq. (5) calculates the Specificity for a single class by finding the ratio between the True Negatives to the sum of the True Negatives and False Positives in the confusion matrix for that class.

The overall Specificity is the mean of individual Specificity of each classes. It is given in Eq. (6):

$$\mathrm{Specificity} = 1/L * \sum_{i=1}^{L} \mathrm{Specificity}_i \tag{6}$$

Where TN is the True Negative, FP is the False Positive, L is the number of classes in Eqs. (5) and (6).

3. Experimental results and discussions

This section contains a detailed explanation of the experimental implementation, results, and discussions.

3.1. Implementation details

The dataset of 1737 study participants has split into mainly 2 sets namely training-validation and test set. 80% of the data is used for training-validation and 20% of the data is used for testing purposes. The training-validation set consists of 273, 697, 418 numbers of AD, MCI, and HC respectively. The testing set consists of 69, 175, and 105 numbers of AD, MCI, and HC respectively. Hence, an equal proportion of the classes for training-validation and testing sets are maintained for the experiments. Fig. 3 illustrates the data segregation used for experiments.

3.1.1. Hyperparameter tuning with Grid Search-Stratified 10 Fold Cross-Validation

Stratified 10 Fold Cross-Validation is performed on the training-validation set. The entire training-validation set is divided into 10 equal-sized folds. Each fold maintains an equal proportion of every label using stratified sampling. The reason for selecting the Stratified 10 Fold Cross-Validation strategy is because it maintains an equal proportion of

[4] Similarly, the explanation can be given for the diagnostic labels **MCI** and **HC**.

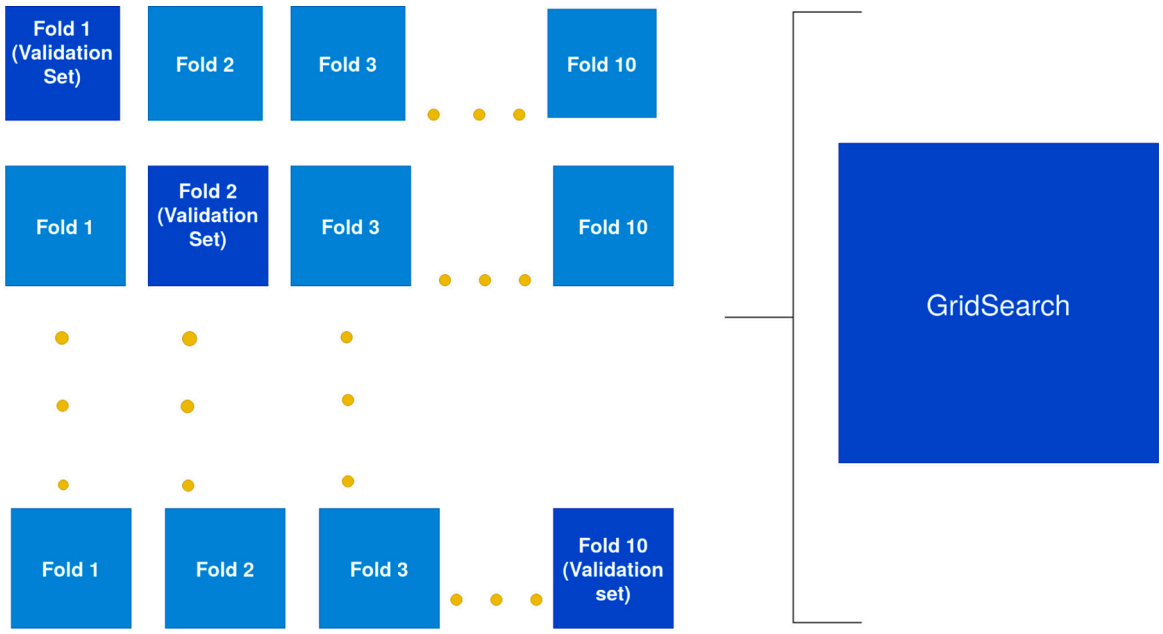


Fig. 4. The diagramatic representation of the Grid Search-Stratified 10 Fold Cross-Validation.

representation of each label which is a good strategy for imbalanced datasets [40–42]. Initially, the model is trained with the nine folds and its performance is validated with the remaining one fold in terms of BCA. This process is repeated 10 times until all the remaining folds (those are part of the training set in the first fold) become a validation set. Then, the BCA of every 10 folds is averaged.

The optimal hyperparameters for the models are found out using the Grid Search technique. The Grid Search technique is used to find out an optimal set of hyperparameters from a list of already pre-defined hyperparameters of a model based on its performance during the cross-validation stage [43,44]. It is also one of the most common hyperparameter tuning technique used for ML tasks in the health domain [44,45]. The possible combination of all the pre-defined hyperparameters is found out and the combination of hyper-parameters with good performance after cross-validation is selected for the model using the Grid Search technique. For our experiments, the combination of hyperparameters that maximizes the overall BCA (average BCA of all the folds) in the Stratified 10 Fold Cross-Validation is selected for the model. The reason for using BCA as the performance indicator is because it combines both Sensitivity and Specificity of the diagnostic labels. Hence, the metric is appropriate for our experiment that also has an imbalanced dataset [46]. Fig. 4 contains the framework for Grid Search – Stratified 10 Fold Cross-Validation.

The test set is kept as an unseen set to check the performance of the trained models. The main aim of creating the training-validation and keeping the testing set as a seperate set is to avoid overfitting. The basic idea is to optimize the hyperparameter using the training-validation set and keeping the test set as unseen data that do not involve any learning. Hence, the test set can be used for evaluating the performance of the models.

All the pool of classifiers is implemented using the scikit-learn package of the Python [40]. Hyperparameter tuning using the Grid Search is performed on the pool of classifiers for finding the optimal hyperparameters. For the homogeneous tree-based classifiers like RF, BDT, ET, and Adaboost, the optimal values are found for the

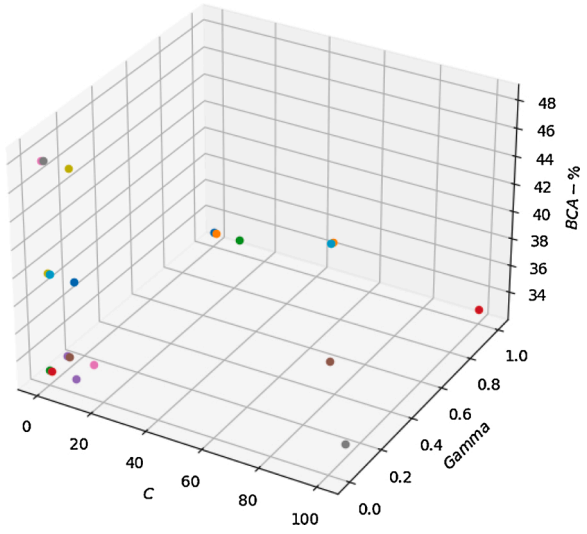


Fig. 5. 3-D visualization of Grid Search hyperparameter tuning of SVM.

hyperparameters such as the number of trees and the maximum depth of a tree and Gini Index is used as the splitting criteria. Gini Index is the most commonly used splitting criteria for RF, BDT, ET, and Adaboost also motivated us to use it for our experiments [47–50]. The range of values considered for the number of trees are [100, 1000, 2000, 3000, 4000, 5000,6000] and maximum depth of a tree are [3, 5, 7, 9, 11, 13, 15] in the Grid Search (see Table 7). Table 7 in the Appendix C contains the Grid Search hyperparameter values for the homogeneous tree-based classifiers such as RF, BDT, ET, and Adaboost. The optimal number of trees and maximum depth are found to be [4000,13], [4000, 13], [4000, 11] for RF, BDT, and ET respectively. It is observed that for BDT, RF, and ET, the BCA starts to reduce when the number of trees becomes 5000,

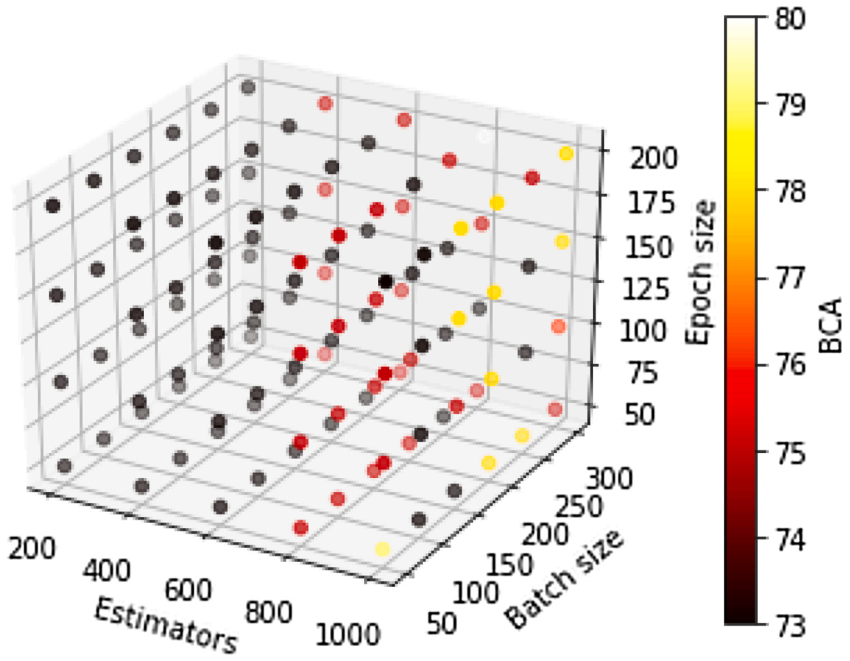


Fig. 6. 4-D visualization of Grid Search hyperparameter tuning of BMLP.

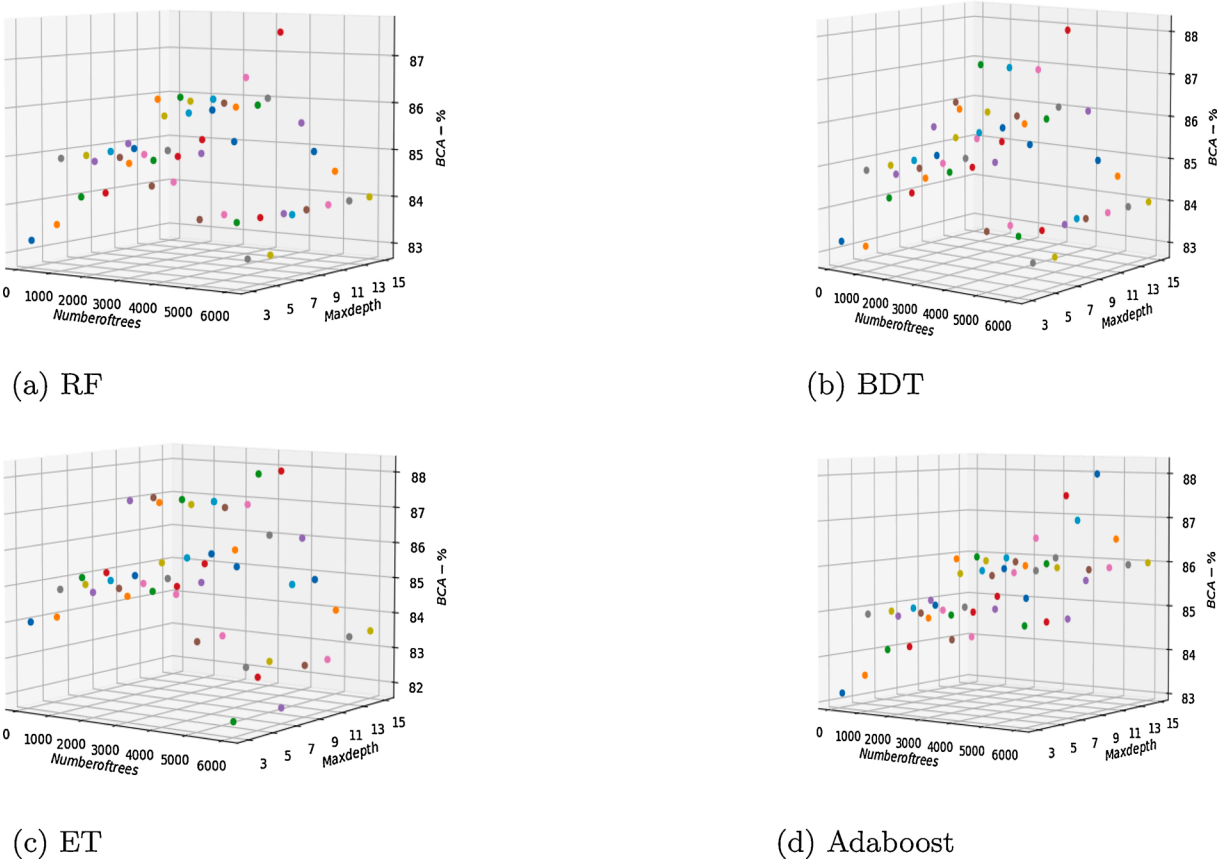


Fig. 7. 3D scatter-plot of the homogeneous tree classifier's hyperparameter tuning using Grid Search.

Table 3
Algorithms used in the DES algorithms with the Python packages.

Algorithm	Python Library
KNORAE	deslib.des.knora_e.KNORAE
META-DES	deslib.des.meta_des.METADES
DES-P	deslib.des.des_p.DESP
KNORAU	deslib.des.knora_u.KNORAU
DES-KNN	deslib.des.des_knn.DESKNN
DES-MI	deslib.des.des_mi.DESMI

Table 4
Algorithms used in the ensemble pool of classifiers with the Python packages.

Algorithm	Sub-algorithms	Python Library
BDT	Bagging	sklearn.ensemble.BaggingClassifier
	DT	sklearn.tree.DecisionTreeClassifier
RF	NIL	sklearn.ensemble.RandomForestClassifier
ET	NIL	sklearn.ensemble.ExtraTreesClassifier
Adaboost	NIL	sklearn.ensemble.Adaboost
BMLP	Bagging	sklearn.ensemble.BaggingClassifier
	MLP	sklearn.neural_network.MLPClassifier
Pooling, Bagging, Stacking ML classifiers	NB	sklearn.naive_bayes.GaussianNB
	SVM	sklearn.svm.svc
	LR	sklearn.linear_model.LogisticRegression
	KNN	sklearn.neighbors.KNeighborsClassifier
	Pooling	NIL (used list concatenation)
	Bagging	sklearn.ensemble.BaggingClassifier
	Stacking	sklearn.ensemble.StackingClassifier

6000 and for all the maximum depth values in the Grid Search (see Table 7). However, the optimal value for the number of trees and a maximum depth is found to be 5000 and 11 respectively for Adaboost. It is also observed that for Adaboost, the BCA starts to reduce when the number of trees becomes 6000 and for the all the maximum depth values in the Grid Search. Fig. 7 illustrates the 3D scatter-plot for the Grid Search hyperparameter values of the RF, BDT, ET, and Adaboost.

The optimal hyperparameter value for the BMLP such as number of MLP's, batch size, and epoch size of each MLP are found out using the Grid Search technique. The most commonly used and efficient Adam optimizer activation function is used for the MLP's [51–53]. The learning rate of the MLP's are selected as 0.01 because it is found as an optimal value for the neural networks in health domain especially for Alzheimer's detection [54–56]. The range of values considered for number of estimators, batch size, and epoch size are [200, 400, 600, 800, 1000], [50, 100, 150, 200, 250, 300], [50, 100, 150, 200] respectively. The optimal hyperparameter for BMLP is found at: number of estimators=800, batch size=300, epoch size=150. Table 8 in Appendix C contains all the possible combination of hyperparameter values using Grid Search for BMLP. Fig. 6 illustrates the 4D scatter plot for the Grid Search hyperparameter values of BMLP.

The RBF kernel is used for the SVM because it is shown to be appropriate for large datasets [57]. Grid Search hyperparameter tuning is performed for the C and Gamma values of the SVM. The range of values considered for C, Gamma values in the Grid Search are [0.1, 1, 10, 100], [1, 0.1, 0.01, 0.001, 0.0001] respectively. The optimal value of C and Gamma hyperparameters are found to be 100 and 0.001 respectively using the Grid Search. Table 9 in Appendix C contains the all the possible combination of hyperparameter values using Grid Search for SVM. Fig. 5 contains the 3D scatter plot for the Grid Search hyperparameter values using SVM. The L2 norm regularization function is used in the LR. The nearest neighbor value 'k' used for the DES algorithms and the KNN classifier in the ML pool of classifiers are the same. The reason for selecting the same 'k' value is based on the approach used by the previous researchers who used KNN classifier in the DES algorithms [21,38,39]. Table 4 contains the Python libraries used for implementing the ML classifiers. Then, these hyperparameter-optimized ML pool of classifiers are applied to the 6 DES algorithms.

The DES algorithms are implemented using the DESLib library package in the Python [58]. Table 3 contains the Python libraries for the 6 DES algorithms. The nearest neighbor value 'k' is required for the region of competence for all the DES algorithms. They are found out using varying values of 'k'. The various values for 'k' such as 3, 5, 7, and 9 have experimented for all the DES algorithms with the pool of classifiers during the cross-validation stage. The 'k' value with the highest overall BCA is selected as the nearest neighbor value for the region of competence for the corresponding experiment. For example, if the BCA using $k = 7$ is found to be the highest for KNORAE with RF as the pool of classifiers during the cross-validation stage, then $k = 5$ is used as the nearest neighbor parameter for the KNORAE-RF experiment during the testing phase. Similarly, the nearest neighbor values are found out for all the experiments involving DES algorithms using this strategy. Fig. 8 contains the BCA for varying values of 'k' for all the experiments involving DES algorithms.[5]

3.2. Results

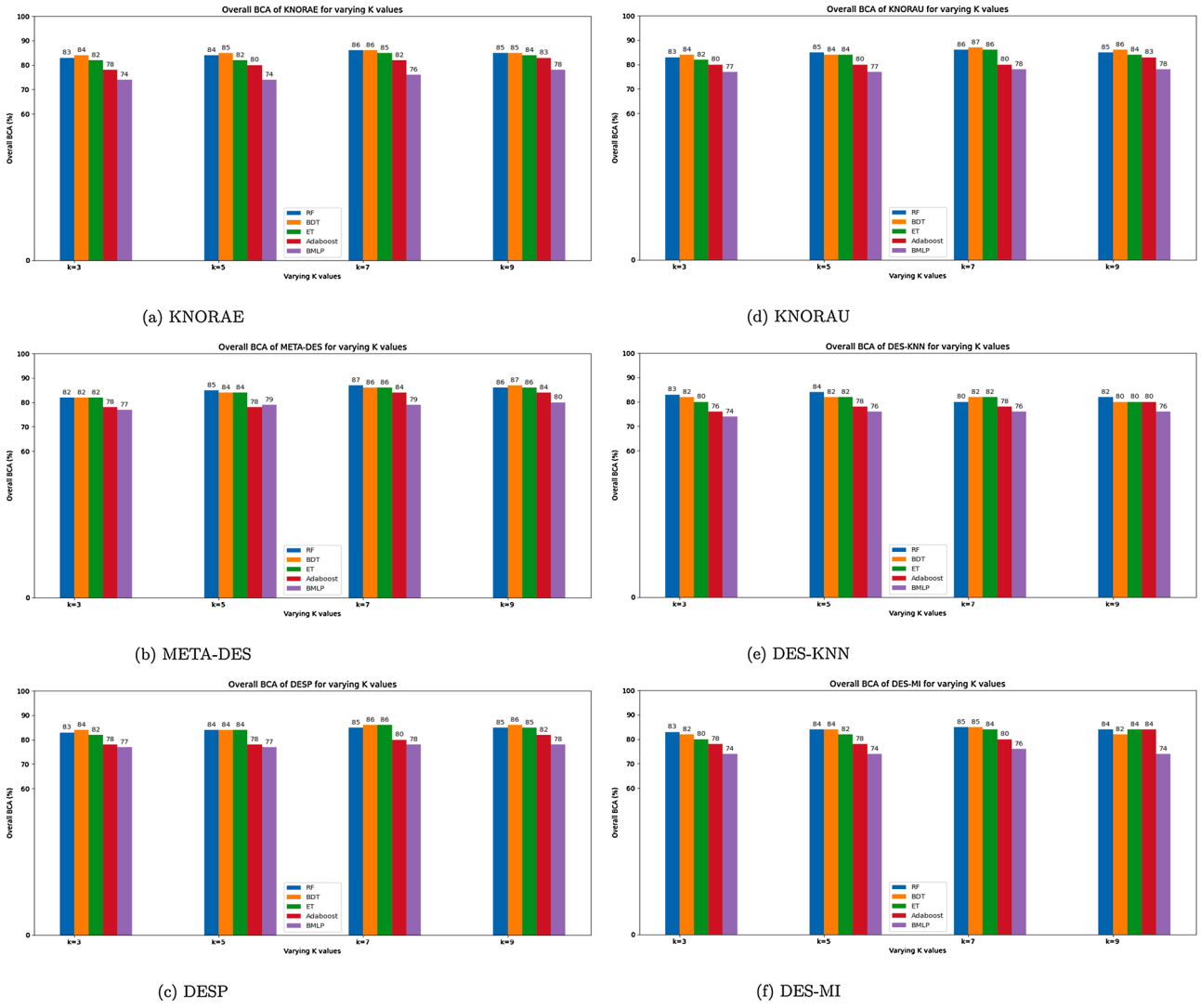
Table 5 contains the BCA reported for all the experiments using stratified 10 fold cross-validation on the training-validation set. It is observed that out of all the DES algorithms, META-DES have reported with highest BCA of 87% with RF and BDT. Moreover, the results after applying DES algorithms to the pool of classifiers are also improved in most of the studies (see Table 5). However, the heterogeneous ML ensembles are excluded for the testing phase due to their low performance in the cross-validation stage (see Table 5).

The following questions are answered through our experiments: a) Which ensemble pool of classifiers using DES algorithms are reported with better performances on the test set? b) Whether the same ensemble pool of classifiers using DES algorithms outperformed without using the DES algorithms on the test set? Table 6 contains a comparison of the pool of classifier's performance using DES and without DES in terms of BCA, Sensitivity, and Specificity. on the test set.

3.2.1. Performance of the ensemble pool of classifiers using the DES algorithms on the test set

Homogeneous DT based classifiers such as RF, BDT, ET has reported better classification results as compared to other ensembles such as BMLP. Among all the DT homogeneous ensembles, ensmeble of RF, BDT with META-DES have achieved a better performance with BCA of 82% and 81.5% respectively. The ensemble of ET with DESP also achieved a better performance with BCA of 82%. As far as Sensitivity is concerned, the ensemble of both RF and BDT with META-DES has outperformed other algorithms with the Sensitivity of 80% using META-DES (see

[5] Results for heterogeneous ensembles are not shown due to poor performances.



(a) KNORAE (d) KNORAU

(b) META-DES (e) DES-KNN

(c) DESP (f) DES-MI

Fig. 8. The overall BCA for varying nearest neighbor values of the DES algorithms.

Table 5
Cross-validation BCA for all the experiments.

Algorithm	BDT	RF	ET	Adaboost	BMLP	Pooling ML	Bagging ML	Stacking ML
KNORAE	86%	86%	85%	83%	78%	33%	33%	44%
META-DES	**87%**	**87%**	**86%**	**84%**	**80%**	30%	22%	33%
DESP	85%	86%	86%	82%	78%	33%	33%	33%
KNORAU	86%	**87%**	**86%**	83%	78%	33%	33%	33%
DES-KNN	84%	82%	82%	80%	76%	27%	33%	33%
DES-MI	85%	85%	84%	**84%**	76%	33%	48%	46%
Without DES	85%	85%	84%	81%	79%	28%	33%	33%

The bold values in the tables are the results that needs to be highlighted.

Table 6
Comparison of Sensitivity, Specificity, and BCA on the test set using DES algorithms.

Algorithm	Metric	RF	BDT	ET	Adaboost	BMLP
KNORAE	Sensitivity	80%	80%	78%	**73%**	65%
	Specificity	82%	80%	80%	**75%**	76%
	BCA	81.5%	80%	79.5%	**74%**	71%
META-DES	Sensitivity	**80%**	**80%**	75%	69%	**68%**
	Specificity	**84%**	**82%**	80%	71%	**74%**
	BCA	**82%**	**81.5%**	77.5%	70%	**72%**
DESP	Sensitivity	76%	76%	**76%**	**70%**	55%
	Specificity	84%	84%	**88%**	**78%**	70%
	BCA	80%	80%	**82%**	**74%**	63%
KNORAU	Sensitivity	78%	76%	78%	66%	56%
	Specificity	78%	80%	74%	72%	70%
	BCA	78%	78%	76%	69%	63%
DES-KNN	Sensitivity	76%	76%	74%	68%	60%
	Specificity	72%	80%	78%	72%	68%
	BCA	74%	78%	76%	70%	64%
DES-MI	Sensitivity	74%	72%	76%	66%	50%
	Specificity	78%	80%	80%	72%	70%
	BCA	76%	76%	78%	69%	60%
Without DES	Sensitivity	72%	72%	76%	60%	62%
	Specificity	80%	80%	76%	72%	70%
	BCA	76%	76%	76%	66%	66%

The bold values in the tables are the results that needs to be highlighted.

Table 6). The BDT and RF using META-DES have reported the highest Sensitivity of 80%. BMLP is reported with a lower BCA, Sensitivity, and Specificity in most of the experiments (see Table 6). Neural network models are observed to perform better in many cases where there is voluminous unstructured data in the form of image, video, and time-series [59,60]. However, there are examples and applications where DT based classifiers outperformed neural networks [61,62]. Although, it is not possible to generalize the classifier that has to be designed for input data. In our study, we have utilized the input baseline data which is non-time series and structured has reported better results for homogeneous DT based ensemble classifiers than the MLP. The homogeneous pool of classifiers reported good results with the META-DES algorithm. The main difference that takes META-DES apart is that it extracts meta-features for each classifier on the defined region of competence.

3.2.2. *Performance of the same ensemble pool of classifiers with and without DES algorithms on the test set*

Most of the pool of classifiers after applying the DES algorithms in the testing set have shown an improved or same BCA. The BCA is decreased for few ensembles such as: RF-DESP, BMLP-KNORAU, BMLP-DES-KNN, BMLP-DES-MI) (see Table 6). The improvement in the BCA is because DES algorithms choose the ensemble of classifiers for each test data based on the performance of the classifiers in the region of competence to where does the test data belong to. A typical single classifier approach focuses on generalizing the entire test data with that classifier. However, flexibility can be achieved in this approach by redistributing a set of multiple classifiers to each test data dynamically. The test data select the appropriate ensemble of classifiers based on its performance in the region of competence (nearest neighbors). This approach achieves two things, 1. It helps in redistribution of the ensemble of classifiers for each test data that prevents over-generalization of an entire test set to a single classifier. 2. The assigning of the ensemble of classifiers for each test data is based on their performances in the neighborhood. This can help in finding the worthy ensemble of classifiers for that test data. Hence, the DES algorithms are capable of increasing the performance of the classifiers.

3.2.3. *Comparison with previous studies*

After thorough observation, we found out that all the related works conducted on ADNI-TADPOLE using longitudinal data (consequent visit data) [14–16], and [17]. But, our study used the baseline visit data of the patients. Consequently, we thought a direct comparison of the results between the studies [14–16], and [17] is unfair (see Table 1).

Our features set is the largest as compared to other related studies, and the best performance using the DES algorithm is reported with a BCA of 84% using META-DES with BDT as the input pool of classifiers. In short, our study contributes to the AD research community by highlighting that better results are also reported using only the baseline multimodal data of ADNI-TADPOLE such as MRI, PET, CSF, cognitive tests, age, sex, and education with classifiers such as BDT, RF, and ET using DES algorithms with them.

4. Study limitations

Our comparative study reports good results using already existing DES algorithms on the ADNI-TADPOLE dataset. Thus, future research pinpoints more on the usage of advanced DES algorithms for the classification of AD. From an ML research point of view, the next focus is to develop an advanced DES algorithm that could solve the complex AD classification problem using a larger multimodal feature set rather than using the already existing black box packages. The other shortcomings of the study are that it used only the baseline data of the patients. The reason for choosing baseline data is to help the doctor by creating a trained baseline model for predicting AD at the baseline visit of a new patient. However, there is much more scope for studying the longitudinal features of the patients, and researching an advanced DES for longitudinal data is a much needed future work.

5. Conclusion

In this paper, we present a comparison of 6 DES algorithms such as KNORAE, META-DES, DESP, KNORAU, DES-KNN, and DES-MI for the 8 input pool of classifiers namely BDT, RF, ET, Adaboost, BMLP, Pooling, Bagging, and Stacking of ML classifiers constituting SVM, NB, LR, and KNN for classification of AD, MCI, and HC patients using baseline multimodal data in terms of BCA, Sensitivity, and Specificity. The performance of the pool of classifiers with DES algorithms is also compared without DES. The novelty of this paper lies in the implementation of DES algorithms in classifying MCI, AD, and HC. Our results on the popular ADNI-TADPOLE suggest that the DES algorithms significantly improved the performance of the pool of classifiers in classifying AD, MCI, and HC. The best result was reported using the META-DES on RF with a BCA of 82% on the test set. The classification performance of the most of ensemble pool of classifiers is increased with DES algorithms for AD, MCI, and HC classification on the test set. However, the classification performance of some of the pool of classifiers is reduced after applying DES algorithms such as: RF is reduced after applying DESP, and BMLP is reduced after applying KNORAU, DES-KNN, and DES-MI. As future work, we are planning to experiment with an even larger feature set consists of genetic data for classifying AD, MCI, and HC. We are also focusing on experimenting and developing efficient DES algorithms for the classification of AD, MCI, and HC. Moreover, our study is conducted only on the baseline data. So, we are planning to investigate longitudinal

Table 7
Homogeneous tree classifiers hyperparameter tuning using Grid Search.

Number of trees	Max depth	BCA			
		RF	BDT	ET	Adaboost
100	3	83.25%	83.25%	84%	83.25%
100	5	83.50%	83%	84%	83.5%
100	7	84%	84%	85%	84%
100	9	84%	84%	85%	84%
100	11	85%	85.5%	87%	85%
100	13	84%	86%	87%	84%
100	15	84%	85%	84%	84%
1000	3	85%	85%	85%	85%
1000	5	85%	85%	85%	85%
1000	7	85%	85%	85%	85%
1000	9	85%	85%	85%	85%
1000	11	86%	86%	87%	86%
1000	13	86%	87%	87%	86%
1000	15	85%	85%	85%	85%
2000	3	85%	85%	85%	85%
2000	5	85%	85%	85%	85%
2000	7	85%	85%	85%	85%
2000	9	85%	85%	85%	85%
2000	11	86%	86%	87%	86%
2000	13	86%	87%	87%	86%
2000	15	85%	85%	85%	85%
3000	3	85%	85%	85%	85%
3000	5	85%	85%	85%	85%
3000	7	85%	85%	85%	85%
3000	9	85%	85%	85%	85%
3000	11	86%	86%	87%	86%
3000	13	86.5%	87%	87%	86%
3000	15	86%	86%	86%	85%
4000	3	86%	86%	86%	86%
4000	5	86%	86%	86%	86%
4000	7	86%	86%	86%	86%
4000	9	86%	86%	86%	86%
4000	**11**	86%	86%	**88%**	86%
4000	**13**	**87.5%**	**88%**	**88%**	87.5%
4000	15	85.5%	86%	86%	85.5%
5000	3	84%	84%	84%	86%
5000	5	84%	84%	84%	86%
5000	7	83%	84%	83%	86%
5000	9	83%	84%	83%	86%
5000	11	83.75%	84%	85%	87%
5000	**13**	85%	85%	85%	**88%**
5000	15	84.5%	84.5%	84%	86.5%
6000	3	84%	84%	82%	86.5%
6000	5	84%	84%	83%	86.5%
6000	7	84%	84%	82%	86%
6000	9	84%	84%	83%	85%
6000	11	84%	84%	83%	85%
6000	13	84%	84%	83.5%	85%
6000	15	84%	84%	83.5%	85%

The bold values in the tables are the results that needs to be highlighted.

Table 8
Grid Search hyperparameter tuning values for BMLP

Number of estimators	Batch size	Epoch size	BCA
200	50	50	73%
200	50	100	73%
200	50	150	73%
200	50	200	73%
200	100	50	73%
200	100	100	73%
200	100	150	73%
200	100	200	73%
200	150	50	73%
200	150	100	73%
200	150	150	73%
200	150	200	73%
200	200	50	73%
200	200	100	73%
200	200	150	73%
200	200	200	73%
200	250	50	73%
200	250	100	73%
200	250	150	73%
200	250	200	73%
200	250	50	73%
200	250	100	73%
200	250	150	73%
200	250	200	73%
200	300	50	75%
200	300	100	75%
200	300	150	75%
200	300	200	75%
400	50	50	73%
400	50	100	73%
400	50	150	73%
400	50	200	73%
400	100	50	73%
400	100	100	73%
400	100	150	73%
400	100	200	73%
400	150	50	73%
400	150	100	73%
400	150	150	73%
400	150	200	73%
400	200	50	73%
400	200	100	73%
400	200	150	73%
400	200	200	73%
400	250	50	73%
400	250	100	73%
400	250	150	73%
400	250	200	73%
400	250	50	73%
400	250	100	73%
400	250	150	73%
400	250	200	73%
400	300	50	75%
400	300	100	75%
400	300	150	75%
400	300	200	75%
600	50	50	73%
600	50	100	73%
600	50	150	73%
600	50	200	73%
600	100	50	73%
600	100	100	73%
600	100	150	73%
600	100	200	73%
600	150	50	73%
600	150	100	73%
600	150	150	73%
600	150	200	73%
600	200	50	73%
600	200	100	73%
600	200	150	73%
600	200	200	73%
600	250	50	73%
600	250	100	73%
600	250	150	73%
600	250	200	73%
600	250	50	73%
600	250	100	73%
600	250	150	73%
600	250	200	73%
600	300	50	75%
600	300	100	75%
600	300	150	75%
600	300	200	75%
800	50	50	75%
800	50	100	75%
800	50	150	75%
800	50	200	75%
800	100	50	75%
800	100	100	75%
800	100	150	75%
800	100	200	75%
800	150	50	75%
800	150	100	75%
800	150	150	75%
800	150	200	75%
800	200	50	75%
800	200	100	75%
800	200	150	75%
800	200	200	75%
800	250	50	73%
800	250	100	73%
800	250	150	73%
800	250	200	73%
800	250	50	73%
800	250	150	75%
800	250	200	75%
800	300	50	73%
800	300	100	75%
800	**300**	**150**	**80%**
800	300	200	79%
1000	50	50	75%
1000	50	100	75%
1000	50	150	73%
1000	50	200	73%
1000	100	50	73%
1000	100	100	73%
1000	100	150	73%
1000	100	200	73%
1000	150	50	75%
1000	150	100	78%
1000	150	150	78%
1000	150	200	78%
1000	200	50	78%
1000	200	100	78%
1000	200	150	78%
1000	200	200	78%
1000	250	50	75%
1000	250	100	73%
1000	250	150	73%
1000	250	200	73%
1000	250	50	75%
1000	250	100	75%
1000	250	150	75%
1000	250	200	75%
1000	300	50	75%
1000	300	100	76%
1000	300	150	78%
1000	300	200	78%

The bold values in the tables are the results that needs to be highlighted.

Table 9
Grid Search hyperparameter tuning values for SVM.

C	Gamma	BCA
0.1	1	33%
1	1	33%
10	1	33%
100	1	33%
0.1	0.1	33%
1	0.1	33%
10	0.1	33%
100	0.1	33%
0.1	0.01	40%
1	0.01	40%
10	0.01	45%
100	**0.01**	**49%**
0.1	0.001	48%
1	0.001	48%
10	0.001	48%
100	0.001	47%
0.1	0.0001	47%
1	0.0001	47%
10	0.0001	47%
100	0.0001	47%

The bold values in the tables are the results that needs to be highlighted.

data as future work. We are also planning to implement feature selection algorithms for finding the optimal features.

Acknowledgements

The authors of this study wish to express their gratitude to the ADNI-TADPOLE for providing access to the database; to the Department of Computer Science, Central University of Tamilnadu, Thiruvarur, India for supporting this study.

Declaration of Competing Interest

The authors hereby declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Featureset descriptions

The description of the various types of features such as MRI, PET, CSF, cognitive tests are as follows:

A.1 Description about the MRI modality feature set

The complete set of features used from the MRI modality are as follows:

- The following 6 features are extracted by the University of California, San Francisco (UCSF) Berkely Medical School, and contributed to the ADNI-TADPOLE consortium [25]. They are 1. Ventricles: Ventricles Volume, 2. Hippo Campus: Hippocampus Volume, 3. Whole Brain: Whole-brain Volume, 4. Entorhinal: Entorhinal Volume, 5. Fusiform: Fusiform Volume, 6. Midtemp: Midtemporal Volume, 7. Intra Cranial Volume: Volume of the Intra Cranial region [25].
- Longitudinal Free Surfer 19 Region of Interest (ROI) based features. The 19 ROI's are Right Pallidum, Right Paracentral, Right Parahippocampal, Right Pars Opercularis, Right Pars Orbitalis, Right Pars Triangularis, Right Pericalcarine, Right Postcentral, Right PosteriorCingulate, 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. The structural features such as the volume of White Matter (WM) parcellation, surface area, thickness average, and thickness standard deviation of the 19 ROI's are used in our study [25]. White matter (WM) is the tissue through which the information passes between different areas of Grey Matter within the central nervous system [25]. Thus, the volume of WM help in identifying the amount of information that passes inside the brain [25]. Consequently, The amount of WM Parcellation is, therefore, can be considered as a good indicator in distinguishing AD and MCI [25].

A.2 Description about the PET modality feature set

The complete set of features used from the PET modality are as follows:

- Fluorodeoxyglucose: Fluorodeoxyglucose (FDG) is used to measure the rate of neurodegeneration inside the brain. FDG is a good indicator of finding efficiency in the working of neurons. Therefore, Average FDG of angular, temporal, and posterior cingulate are selected as features [25].
- Cerebral Metabolic Rate for Glucose (CMRgl), is an indicator of how much blood flow exists in the brain. This is a good parameter to understand the information passage in the brain that is ultimately caused by the blood flow in the cells of the brain [25]. AV45 is used to find out the abnormal protein namely Amyloid-Beta exists in the brain cells. If the amount of AV45 is large, then the cells are likely to suffer from cognitive decline [25]. That is why CMRgl, and AV45 are selected as features for 32 ROI's that are selected as features in the PET modality. The left and right portions of the 32 ROI's are manually selected as features: 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 [25].

A.3 Description about the cognitive tests feature set

The set of cognitive tests used in our study are as follows:

- Clinical Dementia Rating-Scale Box: Clinical Dementia Rating-Scale Box (CDR-SB) test is used to assess the six domains of cognitive and functional performance of an individual such as Memory, Orientation, Judgment, Problem Solving, Community Affairs, Hobbies, and Personal Care [63].
- Mini Mental State Examination: Mini-Mental State Examination (MMSE) is used to assess the cognitive functionalities of an individual such as orientation, attention, and memory, and language [64].
- Alzheimer's Disease Assessment Scale 11: Alzheimer's Disease Assessment Scale 11 (ADAS 11) is used to measure the cognitive functionalities such as following an ordered command, naming of real objects and fingers, copying of geometric forms, preparation of letter for mailing, orientation, word recall test, and word recognition test [65].
- Alzheimer's Disease Assessment Scale 13: Alzheimer's Disease Assessment Scale 13 (ADAS 13) is used to assess the cognitive functionalities of an individual. Along with the ADAS 11 tests, two more points are added in the ADAS 13 namely number cancellation task and a delayed free recall task [65].
- Functional Activities Questionnaire: Functional Activities Questionnaire (FAQ) is the neuropsychological test that is used to assess the daily day to day activities of an individual such as preparing meals, washing clothes, etc. This test contains questions capable of assessing the daily day to day activities of a person [66].
- Montreal Cognitive Assessment Test: Montreal Cognitive Assessment (MOCA) test is used to assess the level of cognitive impairment. It involves tests various cognitive domains such as memory, language, executive functions, visuospatial skills, calculation, abstraction, attention, concentration, and orientation [67].

A.4 Description about the CSF feature set

The set of CSF data used for our study are as follows:

- Abeta Amyloid Peptides
- Tau protein

A.5 Description about demographics feature set

The demographic features included in the study are age, sex, and education.

Appendix B. Pool of classifiers

The complete details about the pool of classifiers used for DES algorithms are as follows:

B.1 BDT

A bagging classifier trains individual base classifiers each on the random subsets of the original dataset by considering all the features [68]. For example, if there are 'n' base classifiers, then a random subset of 'n' training data are trained on each base classifier [68]. The main aim objective of bagging is to reduce the variance of the classifiers by training each base classifier on random training subsets and finally making a prediction considering all the subsets [68].

Decision Tree (DT) is a rule-based classifier that creates a tree-based decision structure for the training data [69]. In other words, a DT contains each node as a feature value that represents a test and the leaf node as an outcome for the test. Similarly, DT represents a decision-based structure based on feature values [69]. But, DT is more prone to variance as the entire tree is built on the inferences made from a single training set. However, the methods like bagging DT can create various DT's by training on different random subsets and thereby reduce the variance of the training set. Using a BDT, each DT is trained on random subsets of the data and aggregated to form the final DT using a bagging classifier [69]. A DES algorithm chooses the best dynamic ensemble of DT's created by the bagging classifier for each test data.

B.2 RF

An RF is quite a different version of the BDT in the sense that each DT is created by random samples that are taken with replacement and all the features are not considered for creating the DT (features are chosen randomly with replacement) [70]. The main advantage of using an RF is to control the overfitting of the data. For example, if there are 'n' training samples and 'm' features, then for each tree a random subset from both the 'n' training samples and 'm' features are chosen with replacement [70]. A DES algorithm choose the best dynamic ensemble of DTs created by the RF for each test data.

B.3 ET

ET is a modified version of both BDT and RF because it uses the whole training dataset for creating each DT (not a random subset). Also, the splitting points in each tree is chosen randomly (unlike BDT and RF where the splitting is performed by maximizing the information gain or Gini Index) [71]. ET do not uses the best split in each tree rather a splitting point is chosen randomly [71]. A DES algorithm chooses the best dynamic ensemble of DT's created by the ET for each test data.

B.4 Adaboost

Adaboost classifier is an ensemble of DT's. It assigns an initial weightage to every sample in the training set and fit into the first DT. If there are any misclassified samples, then the weights of those misclassified samples are adjusted (increased) and fit into the second DT. Like wise, in each subsequent DT's the weights of misclassified samples are adjusted and the process is repeated until the final DT [72,73]. The only difference from RF is that it learns and corrects from its predecessor DT's and also it uses the entire samples in the training set for each tree [72,73]. DES algorithm finds out those combinations of DT's that maximizes the performance (by adjusting the weights of the incorrect samples in the training set in a subsequent manner) for each test data dynamically.

B.5 BMLP

An MLP is a type of feed-forward neural network which is inspired by the biological human neuron system. Every neural network has an input layer, hidden layers (intermediate layers), and an output layer in its [74]. The feature data is fed to the input layer where a random weight and bias are given to each input. In the subsequent layers, an activation function is used for non-linear transformation of the data and the output from each layer is given as the input layer [74,75]. In the final layer, a loss function is used to calculate the error in the prediction. Hence, an optimization is performed based on the loss function and the bias and weights are re-arranged accordingly for better performance [76]. The transformation function is given in Eq. (1):

$$\mathrm{Transform}_{\mathrm{layer}+1} = \mathrm{Activation}_{\mathrm{layer}}((\mathrm{weight}_{\mathrm{layer}} * \mathrm{input}_{\mathrm{layer}} - \mathrm{bias}_{\mathrm{layer}}) \tag{7}$$

Eq. (1) shows that the input for the next layer is the output of the previous layer obtained after performing transformation using the activation function on the product of the weight and input of the previous layer subtracted by the bias values. The bias and weight values are randomly get updated based on the optimization using the error function.

Bagging is performed by training with random subsets on multiple MLP's. A DES algorithm will choose the better performing MLP's from the bagged ensembles for each test data dynamically.

B.6 Pooling, bagging, and stacking of ML classifiers

The following ML classifiers are used:

- SVM:

SVM is used to find a hyperplane in a *n* dimensional space that is used to separate or create an optimal boundary between the multi-class data points in the dataset. Initially, the algorithm will transform every non-linear input training data into a real-valued function using a Radial Basis Function (RBF) [77]. Then, an optimal hyperplane is found out using the transformed values that can maximize the distance between various classes in the dataset [77,78].

- NB:

The NB classifier is based on the property of the Gaussian function and Baye's theorem [79]. Initially, the probability of the occurrence of a test sample belonging to a label is found out using Baye's theorem.

$$P(X == \mathrm{MCI}/f_1, f_2 \ldots f_n) = P(f_1, f_2 \ldots f_n / X == \mathrm{MCI}) * P(X == \mathrm{MCI}) / P(f_1, f_2 \ldots f_n) \tag{8}$$

For example, in Eq. (2) the probability that a test data belongs to MCI labels given there are *n* features denoted by P(X==MCI/f_1,f_2…f_n) (posterior probability) is found out using the Baye's theorem. Initially, the prior probabilities such as P(X==MCI) denoting the probability of occurrence of MCI multiplied by the P(f_1,f_2…f_n/X==MCI) denoting the probability of the occurrence of the *n* features given the test data belongs to MCI divided by the probability of the occurrence of *n* features denoted by P(f_1,f_2…f_n). Likewise, the posterior probabilities for each of the labels for the given test are found out and the test data is assigned the label for which posterior probability is maximum [79]. The probability is found out using the Gaussian distribution function [79].

- LR:

Logistic Regression forms a hierarchical mode of classification in which the 3-way classification problem is split into two stages. In the first stage, the test data is classified into any of the two classes and in the second stage to any of the third class or the chosen class in the first stage. The classification is performed using the logit function as given in Eq. (3) that considers the non-linearity of the samples by transforming into a log function [80].

$$f(x) = \log(x/1 - x) \tag{9}$$

In Eq. (3), x is the probability of the occurrence of test data belongs to a particular label. The logit function performs a simple log-likelihood transformation on this for easy classification. The labels are assigned to the test data depends upon a threshold value set for logit function value.

For example, if the value is greater than 0, then it is assigned to label A, otherwise, label B [80].

- KNN: The KNN algorithm assigns a label to the test data based on the majority of training sample's labels near to the test data. It is a type of supervised algorithm in which all the training samples in the specified (say nearest 1,2…n neighbors) neighborhood and the majority label of the training samples is assigned to the test data [81].

B.6.1 Pooling ML classifiers

A pool consists of all the above 4 ML classifiers are created and the DES algorithm chooses the dynamic ensemble from the pool for each test data dynamically.

B.6.2 Bagging ML classifiers

The 4 ML classifiers are trained using a random subset of the training data. The random subset of the training samples is individually selected for each classifier and aggregated to form the ensemble. The DES algorithm chooses from the begged ensembles of the pool of classifiers for each test data dynamically [68].

Stacking ML clasifiers: The stacked ensemble model consists of base classifiers and a meta classifier. The predictions made by the base classifiers are fed as input to the meta classifier [82]. In our study, the predictions made by the 4 ML classifiers such as SVM, NB, LR, and KNN are taken as input to the DT, used as meta classifier. A DES algorithm finds out the best combination of base classifiers and meta classifier from the pool for each test data dynamically [82].

Appendix C. Grid search stratified 10 fold cross-validation hyperparameter tuning

Following are the overall BCA values for all possible combination of hyperparameter values of the ML classifiers with Grid Search. Table 7 contains the Grid Search hyperparameter values for homogeneous tree classifiers such as RF BDT, ET, and Adaboost.

Table 8 contains the Grid Search hyperparameter tuning values for BMLP.

Table 9 contains the Grid Search hyperparameter tuning values for SVM.

References

[1] As Association, et al., Alzheimer's disease facts and figures, Alzheimer's Dement. 13 (4) (2017) 325–373.

[2] Q. Cao, C.-C. Tan, W. Xu, H. Hu, X.-P. Cao, Q. Dong, L. Tan, J.-T. Yu, The prevalence of dementia: a systematic review and meta-analysis, J. Alzheimer's Dis. 73 (3) (2020) 1157–1166.

[3] P.J. Nestor, P. Scheltens, J.R. Hodges, Advances in the early detection of Alzheimer's disease, Nat. Med. 10 (7) (2004) S34–S41.

[4] D.M. Holtzman, J.C. Morris, A.M. Goate, Alzheimer's disease: the challenge of the second century, Sci. Transl. Med. 3 (77) (2011) 77sr1.

[5] J. Ye, M. Farnum, E. Yang, R. Verbeeck, V. Lobanov, N. Raghavan, G. Novak, A. DiBernardo, V.A. Narayan, Sparse learning and stability selection for predicting mci to ad conversion using baseline adni data, BMC Neurol. 12 (1) (2012) 46.

[6] S. Lahmiri, A. Shmuel, Performance of machine learning methods applied to structural mri and adas cognitive scores in diagnosing alzheimer's disease, Biomed. Signal Process. Control 52 (2019) 414–419.

[7] T.F. Hughes, J.D. Flatt, B. Fu, C.-C.H. Chang, M. Ganguli, Engagement in social activities and progression from mild to severe cognitive impairment: the myhat study, Int. Psychogeriatr./IPA 25 (4) (2013) 587.

[8] F. Clément, S. Belleville, S. Gauthier, Cognitive complaint in mild cognitive impairment and alzheimer's disease, J. Int. Neuropsychol. Soc. 14 (2) (2008) 222–232.

[9] B.J. Kelley, R.C. Petersen, Alzheimer's disease and mild cognitive impairment, Neurol. Clin. 25 (3) (2007) 577–609.

[10] M.-J. Chiu, T.-F. Chen, P.-K. Yip, M.-S. Hua, L.-Y. Tang, Behavioral and psychologic symptoms in different types of dementia, J. Formos. Med. Assoc. 105 (7) (2006) 556–562.

[11] J.T. Coyle, D.L. Price, M.R. Delong, Alzheimer's disease: a disorder of cortical cholinergic innervation, Science 219 (4589) (1983) 1184–1190.

[12] J.E. Galvin, Prevention of alzheimer's disease: lessons learned and applied, J. Am. Geriatr. Soc. 65 (10) (2017) 2128–2133.

[13] P.D. Meek, E. Kristin McKeithan, G.T. Schumock, Economic considerations in alzheimer's disease, Pharmacother. J. Hum. Pharmacol. Drug Ther. 18 (2P2) (1998) 68–73.

[14] P.J. Moore, T.J. Lyons, John Gallacher, Alzheimer's Disease Neuroimaging Initiative, et al., Random forest prediction of alzheimer's disease using pairwise selection from time series data, PLOS ONE 14 (2) (2019).

[15] S. Iddi, D. Li, P.S. Aisen, M.S. Rafii, W.K. Thompson, M.C. Donohue, Alzheimer's Disease Neuroimaging Initiative, et al., Predicting the course of alzheimer's progression, Brain Inform. 6 (1) (2019) 6.

[16] J. Albright, Alzheimer's Disease Neuroimaging Initiative, et al., Forecasting the progression of alzheimer's disease using neural networks and a novel preprocessing algorithm, Alzheimer's Dement. Transl. Res. Clin. Interv. 5 (2019) 483–491.

[17] M. Nguyen, N. Sun, D.C. Alexander, J. Feng, B.T. Thomas Yeo, Modeling alzheimer's disease progression using deep recurrent neural networks, in: 2018 International Workshop on Pattern Recognition in Neuroimaging (PRNI), IEEE, 2018, pp. 1–4.

[18] M. Mehdipour Ghazi, M. Nielsen, A. Pai, M. Jorge Cardoso, M. Modat, S. Ourselin, L. Sørensen, Training recurrent neural networks robust to incomplete data: application to alzheimer's disease progression modeling, Med. Image Anal. 53 (2019) 39–46.

[19] M. Antonakakis, S.I. Dimitriadis, M. Zervakis, A.C. Papanicolaou, G. Zouridakis, Aberrant whole-brain transitions and dynamics of spontaneous network microstates in mild traumatic brain injury, Front. Comput. Neurosci. 13 (2020) 90.

[20] M. Antonakakis, S.I. Dimitriadis, M. Zervakis, A.C. Papanicolaou, G. Zouridakis, Altered rich-club and frequency-dependent subnetwork organization in mild traumatic brain injury: a meg resting-state study, Front. Hum. Neurosci. 11 (2017) 416.

[21] A.H.R. Ko, R. Sabourin, A. Souza Britto Jr., From dynamic classifier selection to dynamic ensemble selection, Pattern Recognit. 41 (5) (2008) 1718–1731.

[22] B.B. Damodaran, R.R. Nidamanuri, Y. Tarabalka, Dynamic ensemble selection approach for hyperspectral image classification with joint spectral and spatial information, IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens. 8 (6) (2015) 2405–2417.

[23] T. Woloszynsk, M. Kurzynski, A probabilistic model of classifier competence for dynamic ensemble selection, Pattern Recognit. 44 (10–11) (2011) 2656–2668.

[24] A. Nabiha, F. Nadir, New dynamic ensemble of classifiers selection approach based on confusion matrix for arabic handwritten recognition, in: 2012 International Conference on Multimedia Computing and Systems, IEEE, 2012, pp. 308–313.

[25] R.V. Marinescu, N.P. Oxtoby, A.L. Young, E.E. Bron, A.W. Toga, M.W. Weiner, F. Barkhof, N.C. Fox, S. Klein, D.C. Alexander, et al., Tadpole Challenge: Prediction of Longitudinal Evolution in Alzheimer's Disease, 2018 (arXiv preprint), arXiv: 1805.03909.

[26] http://adni.loni.usc.edu//, adni-alzheimer's disease neuroimaging initiative.

[27] Y. Dong, C.-Y. Joanne Peng, Principled missing data methods for researchers, SpringerPlus 2 (1) (2013) 222.

[28] C. Curley, R.M. Krause, R. Feiock, C.V. Hawkins, Dealing with missing data: a comparative exploration of approaches using the integrated city sustainability database, Urban Aff. Rev. 55 (2) (2019) 591–615.

[29] L.L. Brockmeier, J.D. Kromrey, C.V. Hines, Systematically missing data and multiple regression analysis: an empirical comparison of deletion and imputation techniques, Mult. Linear Regres. Viewp. 25 (1998) 20–39.

[30] D.J. Stekhoven, P. Stekhoven, Missforest-non-parametric missing value imputation for mixed-type data, Bioinformatics 28 (1) (2012) 112–118.
[31] D.J. Stekhoven, Missforest: Nonparametric Missing Value Imputation Using Random Forest, ASCL, 2015 pages ascl-1505.
[32] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, R.B. Altman, Missing value estimation methods for dna microarrays, Bioinformatics 17 (6) (2001) 520–525.
[33] W. McKinney, et al., pandas: a foundational python library for data analysis and statistics, Python High Perform. Sci. Comput. 14 (9) (2011).
[34] R.M.O. Cruz, L.G. Hafemann, R. Sabourin, G.D.C. Cavalcanti, Deslib: A Dynamic Ensemble Selection Library in Python, 2018 (arXiv preprint), arXiv:1802.04967.
[35] R.M.O. Cruz, R. Sabourin, G.D.C. Cavalcanti, T.I. Ren, Meta-des: a dynamic ensemble selection framework using meta-learning, Pattern Recognit. 48 (5) (2015) 1925–1935.
[36] T. Woloszynski, M. Kurzynski, P. Podsiadlo, G.W. Stachowiak, A measure of competence based on random classification for dynamic ensemble selection, Inf. Fusion 13 (3) (2012) 207–213.
[37] R.G.F. Soares, A. Santana, A.M.P. Canuto, M. Carlos Pereira de Souto, Using accuracy and diversity to select classifiers to build ensembles, in: The 2006 IEEE International Joint Conference on Neural Network Proceedings, IEEE, 2006, pp. 1310–1316.
[38] A.S. Britto Jr., R. Sabourin, L.E.S. Oliveira, Dynamic selection of classifiers-a comprehensive review, Pattern Recognit. 47 (11) (2014) 3665–3680.
[39] S. García, Z.-L. Zhang, A. Altalhi, S. Alshomrani, F. Herrera, Dynamic ensemble selection for multi-class imbalanced datasets, Inf. Sci. 445 (2018) 22–37.
[40] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: machine learning in Python, J. Mach. Learn. Res. 12 (2011) 2825–2830.
[41] M.S. Santos, J.P. Soares, P.H. Abreu, H. Araujo, J. Santos, Cross-validation for imbalanced datasets: avoiding overoptimistic and overfitting approaches [research frontier], IEEE Comput. Intell. Mag. 13 (4) (2018) 59–76.
[42] S. Jain, E. Kotsampasakou, G.F. Ecker, Comparing the performance of meta-classifiers-a case study on selected imbalanced data sets relevant for prediction of liver toxicity, J. Comput.-Aided Mol. Des. 32 (5) (2018) 583–590.
[43] P. Liashchynskyi, P. Liashchynskyi, Grid Search, Random Search, Genetic Algorithm: A Big Comparison for nas, 2019 (arXiv preprint), arXiv:1912.06059.
[44] S. Mezzatesta, C. Torino, P. De Meo, G. Fiumara, A. Vilasi, A machine learning-based approach for predicting the outbreak of cardiovascular diseases in patients on dialysis, Comput. Methods Programs Biomed. 177 (2019) 9–15.
[45] M. Sato, K. Morimoto, S. Kajihara, R. Tateishi, S. Shiina, K. Koike, Y. Yatomi, Machine-learning approach for the development of a novel predictive model for the diagnosis of hepatocellular carcinoma, Sci. Rep. 9 (1) (2019) 1–7.
[46] A. Tharwat, Classification assessment methods, Appl. Comput. Inform. (2020).
[47] M. Kropf, D. Hayn, G. Schreier, Ecg classification based on time and frequency domain features using random forests, in: 2017 Computing in Cardiology (CinC), IEEE, 2017, pp. 1–4.
[48] K. Açıcı, Ç. Berke Erdaş, T. Aşuroğlu, M.K. Toprak, H. Erdem, H. Oğul, A random forest method to detect parkinson's disease via gait analysis, in: International Conference on Engineering Applications of Neural Networks, Springer, 2017, pp. 609–619.
[49] H. Li, G. Hu, J. Li, M. Zhou, Intelligent fault diagnosis for large-scale rotating machines using binarized deep neural networks and random forests, IEEE Trans. Autom. Sci. Eng. (2021).
[50] M. Azka Putra, N. Akhmad Setiawan, S. Wibirama, Wart treatment method selection using adaboost with random forests as a weak learner, Commun. Sci. Technol. 3 (2) (2018) 52–56.
[51] K. Gopalakrishnan, S.K. Khaitan, A. Choudhary, A. Agrawal, Deep convolutional neural networks with transfer learning for computer vision-based data-driven pavement distress detection, Constr. Build. Mater. 157 (2017) 322–330.
[52] S. Bera, V.K. Shrivastava, Analysis of various optimizers on deep convolutional neural network model in the application of hyperspectral remote sensing image classification, Int. J. Remote Sens. 41 (7) (2020) 2664–2683.
[53] U. Mahadeo Khaire, R. Dhanalakshmi, High-dimensional microarray dataset classification using an improved adam optimizer (iadam), J. Ambient Intell. Hum. Comput. 11 (11) (2020) 5187–5204.
[54] S.-H. Wang, P. Phillips, Y. Sui, B. Liu, M. Yang, H. Cheng, Classification of alzheimer's disease based on eight-layer convolutional neural network with leaky rectified linear unit and max pooling, J. Med. Syst. 42 (5) (2018) 1–11.
[55] Y. Umeda-Kameyama, M. Kameyama, T. Tanaka, B.-K. Son, T. Kojima, M. Fukasawa, T. Iizuka, S. Ogawa, K. Iijima, M. Akishita, Screening of alzheimer's disease by facial complexion using artificial intelligence, Aging (Albany NY) 13 (2) (2021) 1765.
[56] C. Park, J. Ha, S. Park, Prediction of alzheimer's disease based on deep neural network by integrating gene expression and dna methylation dataset, Expert Syst. Appl. 140 (2020) 112873.
[57] Zuzana Majdisova, Vaclav Skala, A Radial Basis Function Approximation for Large Datasets, 2018 (arXiv preprint), arXiv:1806.04243.
[58] R.M.O. Cruz, L.G. Hafemann, R. Sabourin, G.D.C. Cavalcanti, Deslib: a dynamic ensemble selection library in python, J. Mach. Learn. Res. 21 (8) (2020) 1–5.
[59] C. Pelletier, G.I. Webb, F. Petitjean, Temporal convolutional neural network for the classification of satellite image time series, Remote Sens. 11 (5) (2019) 523.
[60] C.-L. Liu, W.-H. Hsaio, Y.-C. Tu, Time series classification with multivariate convolutional neural network, IEEE Trans. Ind. Electron. 66 (6) (2018) 4788–4797.
[61] M. Liu, M. Wang, J. Wang, D. Li, Comparison of random forest, support vector machine and back propagation neural network for electronic tongue data classification: application to the recognition of orange beverage and chinese vinegar, Sens. Actuators B: Chem. 177 (2013) 970–980.
[62] T. Han, D. Jiang, Q. Zhao, L. Wang, K. Yin, Comparison of random forest, artificial neural networks and support vector machine for intelligent diagnosis of rotating machinery, Trans. Inst. Meas. Control 40 (8) (2018) 2681–2693.
[63] M.N. Samtani, N. Raghavan, G. Novak, P. Nandy, V.A. Narayan, Disease progression model for clinical dementia rating-sum of boxes in mild cognitive impairment and alzheimer's subjects from the alzheimer's disease neuroimaging initiative, Neuropsychiatric Dis. Treat. 10 (2014) 929.
[64] L. Kurlowicz, M. Wallace, et al., The mini-mental state examination (mmse), J. Gerontol. Nurs. 25 (5) (1999) 8–9.
[65] J. Skinner, J.O. Carvalho, G.G. Potter, A. Thames, E. Zelinski, P.K. Crane, L. E. Gibbons, The alzheimer's disease assessment scale-cognitive-plus (adas-cog-plus): an expansion of the adas-cog to improve responsiveness in mci, Brain Imaging Behav. 6 (4) (2012) 489–501.
[66] A.M. Mayo, Use of the Functional Activities Questionnaire in Older Adults With Dementia. Try This: Best Practices in Nursing Care to Older Adults with Dementia D, 13, 2012.
[67] P. Julayanont, Z.S. Nasreddine, Montreal cognitive assessment (moca): concept and clinical review. Cognitive Screening Instruments, Springer, 2017, pp. 139–195.
[68] L. Breiman, Bagging predictors, Mach. Learn. 24 (2) (1996) 123–140.
[69] D. Steinberg, P. Colla, Cart: classification and regression trees, Top Ten Algorithms Data Min. 9 (2009) 179.
[70] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.
[71] P. Geurts, D. Ernst, L. Wehenkel, Extremely randomized trees, Mach. Learn. 63 (1) (2006) 3–42.
[72] T. Hastie, S. Rosset, J. Zhu, H. Zou, Multi-class adaboost, Stat. Interface 2 (3) (2009) 349–360.
[73] Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, J. Comput. Syst. Sci. 55 (1) (1997) 119–139.
[74] G.E. Hinton, Connectionist learning procedures. Machine Learning, Elsevier, 1990, pp. 555–610.
[75] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (2010) 249–256.
[76] D.P. Kingma, J. Ba, Adam: A Method for Stochastic Optimization, 2014 (arXiv preprint), arXiv:1412.6980.
[77] R. Debnath, H. Takahashi, Learning capability: classical rbf network vs. svm with gaussian kernel, in: International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems, Springer, 2002, pp. 293–302.
[78] Y. Sun, A. Gilbert, A. Tewari, But how does it work in theory? Linear svm with random features. Advances in Neural Information Processing Systems, 2018, pp. 3379–3388.
[79] D.D. Lewis, Naive (bayes) at forty: the independence assumption in information retrieval, in: European Conference on Machine Learning, Springer, 1998, pp. 4–15.
[80] C.R. Mehta, N.R. Patel, Exact logistic regression: theory and examples, Stat. Med. 14 (19) (1995) 2143–2160.
[81] N. Roussopoulos, S. Kelley, F. Vincent, Nearest neighbor queries, Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data (1995) 71–79.
[82] D.H. Wolpert, Stacked generalization, Neural Netw. 5 (2) (1992) 241–259.