Automatic detection of Alzheimer’s disease progression: An efficient information fusion approach with heterogeneous ensemble classifiers

Shaker El-Sappagh1,2,3†, Farman Ali4†, Tamer ABUHMED3*, Jaiteg Singh5, Jose M. Alonso6*
1 Faculty of Computer Science and Engineering, Galala University, 435611, Suez, Egypt
2 Information Systems Department, Faculty of Computers and Artificial Intelligence, Benha University, 13518, Banha, Egypt
3 College of Computing and Informatics, Sungkyunkwan University, South Korea
4 Department of Software, Sejong University, Seoul, South Korea
5 Chitkara University Institute of Engineering and Technology, Chitkara University, Punjab, India
6 Departamento de Electrónica e Computación, Universidade de Santiago de Compostela, 15782, Santiago de Compostela, SPAIN

*Corresponding author: Tamer ABUHMED (tamer@skku.edu), Jose M. Alonso (josemaria.alonso.moral@usc.es)
†These authors contributed equally to this work and co-first authors.

Abstract
Predicting Alzheimer’s disease (AD) progression is crucial for improving the management of this chronic disease. Usually, data from AD patients are multimodal and time series in nature. This study proposes a novel ensemble learning framework for AD progression incorporating heterogeneous base learners into an integrated model using the stacking technique. This framework is used to build a 4-class ensemble classifier, which predicts AD progression 2.5 years in the future based on the multimodal time-series data. Statistical measures have been extracted from the longitudinal data to be used by the conventional machine learning models. The examined ensemble members include k-nearest neighbor, extreme gradient boosting, support vector machine, random forest, decision tree, and multilayer perceptron. We utilize three time-series modalities and one static non-time series modality of 1371 subjects from the Alzheimer’s disease neuroimaging initiative (ADNI) to validate our model. Several homogeneous and heterogeneous combinations of ensemble members were implemented, and their performance compared. The balance between accuracy and diversity when selecting ensemble members was investigated. We found that both accuracy and diversity are equally critical metrics to obtain an optimal ensemble model. Furthermore, our testing showed that the proposed model achieves outstanding progression prediction performance. The proposed model achieved a high performance without using neuroimaging data, which means that the model could be implemented in low-cost healthcare environments. The proposed model has achieved superior results compared with the state-of-the-art techniques in Alzheimer’s and ensemble classifiers domains. The proposed framework can be used to implement efficient information fusion ensembles for other medical and non-medical problems.

Keywords: Computational Intelligence; data fusion; ensemble classifiers; stacking; data analysis; Alzheimer disease progression detection.

1. Introduction
Alzheimer’s disease (AD) is an irreversible and incurable neurodegenerative disease. It is the most common form of dementia in the elderly [1]. AD has debilitating effects on patients, their families, society and causes immeasurable grief to caregivers. Mild cognitive impairment (MCI) is an intermediate stage between cognitively normal and dementia. Around 10% to 20% of MCI patients progress to AD within one year [2][3], the remaining non-converting subjects either develop other forms of dementia or remain stable. Early identification of patients who will convert to AD before the appearance of advanced symptoms is crucial for proper treatment and remains a significant medical challenge. Progression detection potentially enables early AD diagnosis and allows the medication to be administered to slow down the disease’s progression. However, even discriminating stable MCI (sMCI) from progressive MCI (pMCI) patients is challenging because these patients have similar properties. The combination of these two challenging problems yields a much more complicated 4-class (cognitive normal [CN] vs. sMCI vs. pMCI vs. AD) classification problem for any AD progression detection approach.

Recently, many studies and competitions have been proposed various approaches to solve this problem using a combination of markers and biomarkers, e.g., positron emission tomography (PET) or magnetic resonance imaging
(MRI) data as inputs to machine learning (ML) algorithms [1][4][5][6]. These studies were based on a limited set of features from a few modalities, and many relied especially heavily on neuroimaging data. In this context, Bron et al. [7] proposed the CADDementia grand challenge1 based on an MRI dataset of 384 patients. The challenge involves classification to one of three classes (CN vs. MCI vs. AD), i.e., there is no differentiation between sMCI and pMCI. By achieving an accuracy of 63.0%, Sorensen et al. [8] won this challenge using a linear discriminant analysis (LDA) classifier fed with MRI volumetric, cortical thickness, hippocampal texture, and hippocampal shape features. Recently, Kaggle2 proposed another competition for AD prediction based on a set of MRI modality features plus three demographics. This challenge was presented as a 4-class classification problem (CN vs. pMCI vs. sMCI vs. AD). Again, the number of cases utilized was limited, as the training and test sets consisted of only 240 and 160 subjects, respectively. As expected, the performance of the resulting models was worse than in the CADDementia challenge because increasing the number of classes makes the classification problem harder. For example, Yao et al. [6] proposed a hierarchical ensemble classifier that converts this 4-class classification problem into five binary classification problems, achieving 54.38% accuracy. Dimitriadis et al. [9] built five ensemble classifiers based on the random forest classification technique to achieve an AD diagnosis as the result of a 4-class classification problem. They achieved an accuracy of 61.9% and asserted that this performance was the best recorded in the literature to that date.

Although there have been many AD classification and progression studies that have used a wide variety of data types, including MRI [10][11][12], PET [13], cerebrospinal fluid (CSF) biomarkers [14][15], or fusion of neuro-images [16], there is still a significant gap between research results and being able to apply these systems in daily clinical practice [17][18]. The deployment of ML models for real-time AD diagnoses is still limited because the results of previous studies have been inconclusive. This is mainly due to the complexity of the problem, the inefficient utilization of available data, and the ineffective design of the ML models [19]. There has been an increasing interest in multivariate approaches [20], where information fusion plays a crucial role in transforming heterogeneous patient data into high-quality fused data [21][22]. By taking this approach, we can provide a complete picture of patient status. Most existing studies mainly depend on extracted features from MRI neuroimaging, but these have failed to accurately predict MCI-to-AD progression [20][21]. Donnelly-Kehoe et al. [23] concluded that the maximum accuracy achieved by using MRI features does not reach the standard of using the mini-mental state examination (MMSE) by itself. In addition, because AD progression classification is medically very complex, depending only on MRI is not acceptable to physicians. In healthcare environments, physicians usually integrate many features from different modalities to make their decisions. Hence, it is essential we explore fusing multiple modalities such as cognitive score markers (CSs), MRI, neuropsychological battery markers (NSB), and CSF biomarkers [24][25][26] to improve the accuracy of classification models. However, the effect of multimodal fusion needs further study.

Since AD is a chronic disease, a patient’s condition keeps evolving over time, and this is because patients are experiencing gradual structural and functional changes to their brain. Most previous studies have predicted the progression of MCI based on baseline visit data only [26]. Thus, they fail to fully represent or consider time-variant medical records that could facilitate comprehensive predictive ML models. Traditional time series methods, such as hidden Markov chains, have been used to study the AD progression detection problem [27][28][29][30]. As the maximum number of visits for a patient in our dataset is four, traditional time series methods are not suitable for capturing disease progression patterns [31]. Deep learning models, such as convolutional neural networks (CNN) and recurrent neural networks (RNN), are also commonly used for time series data analysis, but they require a large amount of data for model training [22][21]. Robust hybrid deep learning models have already been successfully applied to AD progression detections [21]. Moreover, in the medical domain, it is not easy to introduce novel ML methods while physicians are asking for methods that are multi-modal with comprehensible recommendations [32]. Unfortunately, neural network-based ML methods act as non-understandable black boxes and behave poorly when considering small or medium-sized datasets outside of image processing. In this work, we avoid using the neural network-based ML methods and only consider the interpretable ML models such as the decision tree, support vector machines, and random forest.

1 CADDementia grand challenge: https://caddementia.grand-challenge.org/
2 KAGGLE competition on MCI prediction: https://www.kaggle.com/c/mci-prediction/
The role of time series data to improve ML models in the AD progression domain deserves more attention. Authors have published several research works in AD domain [18][21][22][32][33]. In [21], we proposed two novel deep learning models for AD progression detection. These models were based on the fusion of multimodal time series data of four-time steps. One model is a hybrid model of Bidirectional LSTM and random forest, and the other model is based on multitask modeling to predict seven cognitive scores. These scores were used to predict AD progression after 2.5 years. In [22], we proposed a novel deep learning model called the stacked CNN-LSTM model. The technique was based on a multitask modeling approach to jointly predict multiple cognitive scores (regression tasks) and concurrently AD progression (multiclass classification task) within 7.5 years. The study mainly concentrated on studying the role of adding multiple time steps to enhance the performance of a deep learning model and the role of multitasking to build a stable and reliable deep learning model. We concentrated mainly on proposing a deep learning model that can learn complex and multimodal time series data of 15-time steps. In [32], we tried to provide a more understandable decision for AD progression detection by providing explainability capabilities to our model. We used the random forest to build a two-layer model, i.e., the first layer was to early diagnose AD patients, and the second layer was to detect possible MCI-to-AD progression. In [18], we tried to build a cost-effective machine learning model for AD progression detection. The model has been based on a set of cheap features that can be easily collected from patient’s electronic health record at a low cost. As far as we know, the application of ML algorithms using multimodal and time-series data has not yet achieved good results [34]. Ritter et al. [1] combined ten modalities, including MRI, PET, and NSB, to predict AD progression within three years. They achieved an accuracy of 73% using support vector machines. The generalization error of ML models can be divided into three parts: \( \text{error} = \text{bias}^2 + \text{variance} + \text{noise} \). An ensemble system, also known as multiple classifier system, combines a pool of intelligent classifiers seeking to exploit the strengths of each classifier in such a way as to reduce the generalization error you may get from any single model [35]. For example, bagging methods use undersampling to minimize the variance of the base models. Boosting methods reduce bias using a weighting strategy. Subspace methods divide the feature vector space into several subspaces, this improves the diversity of the trained models and reduces noise. Based on this data fusion from multiple sources, an ensemble classifier can become more accurate and outperform the best base classifiers [36][37]. Ensemble models have been used for years in different applications, including in AD progression prediction [6][38][39]. In the TADPOLE grand challenge3, Moore et al. [40] applied the random forest technique to predict AD achieving an AUC of 0.82 and a 3-class classification accuracy of 0.73. Most AD studies have been based on small datasets using baseline MRI features. Farhan et al. [38] used only five features from 85 MRI scans to build an ensemble of classifiers. Unfortunately, these models are not trusted in the medical domain due to a lack of understanding of how the conclusions were drawn [1]. Although ensemble models are expected to achieve better performance than base models [35], this optimal performance usually implies an increase in the complexity of the model that makes it hard for physicians to interpret. Moreover, optimal performance is not guaranteed simply using an ensemble model, and an appropriate ensemble architecture must be carefully designed. For example, (1) the most informative and discriminative features must be selected; (2) suitable data and decision fusion strategies must be chosen; and (3) the most accurate and highly diverse, least correlated, as well as mutually complementary base learners must be implemented. However, it is critical to select the best set of base classifiers and then integrating them in an optimal way for creating an accurate and stable ensemble. Yu and Zhao [41] proposed the Bayesian network-based probabilistic ensemble learning strategy to automatically and effectively embed the suitable base classifiers in a Bayesian network in a probabilistic manner. The study highlighted the importance of selecting suitable base classifiers, and they used the Bayesian network to take this decision. It is worth noting that current AD ensemble-based studies tend to utilize a limited amounts of training data, feature sets, and numbers of modalities while ignoring time series data completely [40][42].

To overcome these limitations that are found in current AD detection approaches, this work proposes a methodology to find the best ensemble classifier by looking at six types of classification models, namely the decision tree (DT), support vector machines (SVM), random forest (RF), k-NN (KNN), multilayer perceptron (MLP), and extreme gradient boosting (XGB) models [40][43][44][45]. This study considers ensemble models that are based on the fusion of heterogeneous yet complementary time-series data from different AD modalities. Unlike existing literature, we used AD patient’s multimodal medical information from various time points to predict the disease’s progression at a

---

3 TADPOLE grand challenge: https://tadpole.grand-challenge.org/
future time point. The study optimized this set of well-known machine learning techniques. The proposed models were optimized using a collection of cost-effective time-series features including patient’s comorbidities, cognitive scores, medication history, and demographics. The current study implemented a different machine leaning technique (ensemble classifiers), explored different feature set to optimize ensemble models, perform critical statistical analysis for feature engineering, and prove the theory of diversity vs. accuracy to build a highly performing ensemble classifier for AD progression detection. We built information fusion enabled ensemble classifiers based on a patient’s first four visits (baseline [BL], month-6 [M06], month-12 [M12], and month-18 [M18]) to predict the patient’s AD state 2.5 years the final visit (i.e., at month-48). It is challenging to select the best set of features from a patient’s highly heterogeneous multimodal data. These features should form a representative, informative, and discriminative list that best describes the temporal patterns at play in the time-series data. In addition, we must integrate these time series features with other baseline static features such as age, number of years in education, etc. It should also be noted that selecting a subset of base classifiers based only on their individual performance does not guarantee optimum performance for the resulting ensemble. Diversity of base learners is crucial for implementing an accurate ensemble [36]; therefore, we studied the accuracy/diversity trade-off and its role in selecting candidate base learners from a pool of classifiers. A single measure of diversity is inadequate [46], so we implemented three pair-wise measures (Q-statistic, correlation, and disagreement) to test the diversity among fused models. We also implemented a list of ensemble frameworks to study the effect of early and late data fusion schemes on the ensembles’ performance. Then, we select the best ensemble model and optimize its parameters. To the best of our knowledge, this is the first study that builds an ensemble model for multiclass AD prediction detection based on multimodal time series data. The main contributions of this study are as follows:

- An ensemble learning framework based on extensive multimodal time series data is proposed for AD progression detection. The proposed framework integrates early data fusion and late decision fusion features to create more stable and accurate ensemble classifiers.
- Statistical analysis is conducted for the feature extraction process using four medically critical modalities (i.e., CSs + MRI + NSB + Static). These modalities are time-series data with 4-time timesteps (i.e., BL + M06 + M12 + M18).
- Information gain and recursive feature elimination strategies are applied to select the most discriminative features from time series and static modalities.
- An actual patient data from 1371 subjects in the Alzheimer’s Disease Neuroimaging Initiative (ADNI)4 is used to optimize the proposed AD progression prediction ensemble model. The model predicts MCI-to-AD progression 2.5 years after the last visit. We explore the role of different AD modality combinations on AD disease progression detection.
- Extensive experiments are performed to evaluate the effectiveness of a set of proposed ensemble models based on complete patient profiles. We also studied the behavior of different ensemble techniques, including voting and stacking. In addition, we compared the behavior of these ensembles with base models such as RF, XGB, SVM, MLP, DT, and KNN.
- We compare the homogeneous and heterogeneous designs of ensemble models and measure the effects of that on the ensemble’s performance.
- We explore the effect of changing the diversity and performance of base classifiers on the overall performance of the ensemble. In other words, we evaluate the tradeoff between diversity and accuracy of based models on the overall performance of ensembles.
- To implement the optimum ensemble model, we consider both the medical relationship among selected features from each modality and the accuracy/diversity trade-off of the base classifiers. Accordingly, we chose a discriminating list of features, the best fusion scheme, and the most accurate and diverse subset of base classifiers.
- The hyperparameters of the selected stacking ensemble framework are optimized using a random search method and real-world data from the ADNI. The experimental results confirm the effectiveness of this stacking model for AD progression detection.

4 http://adni.loni.usc.edu/
The rest of the paper is organized as follows. Section 2 discusses our methodology and introduces the proposed framework. Section 3 discusses the experimental results. Section 4 concludes the paper and provides readers with hints on future directions to extend the proposed models.

2. Methods

2.1 ADNI study

The dataset used in this study is the ADNIG database. The database was accessed on March 18, 2019. We selected 1371 subjects (54.5% male) from ADNI 1, ADNI GO, and ADNI 2 based on the available data. Subjects are categorized into four groups based on the individual’s clinical diagnosis at the baseline and at future-time points, this is illustrated in Fig. 1: (1) CN: 419 subjects diagnosed to be CN at baseline and remained CN over all future time steps; (2) sMCI: 473 subjects diagnosed to have MCI at all time-points; (3) pMCI: 140 subjects that had MCI at baseline+M06+M12+M18 visits but were predicted to progress to AD within 2.5 years from M18 (i.e., by the M48 visit); (4) AD: 339 subjects diagnosed as having AD in all visits. A list of patient IDs used can be found in Supplementary File 1. Any subject that showed improvement in his/her diagnoses during follow-ups was excluded from the study. In other words, those clinically diagnosed as having MCI but reverted to being CN, or those clinically diagnosed as having AD but reverted to having MCI or being CN are considered to have been misdiagnosed as AD is an irreversible form of dementia. Furthermore, cases that where direct conversion from CN to AD occurs are also discarded.

2.2 Selected raw features

Several types of biomarker and neuropsychological test entries are included in the ADNI data and have been individually validated for AD progression. In accordance with results reported in previous research studies in the specialized literature, the following features are considered as potential predictors for AD progression (we have organized these raw features into two main groups) [1][2][6][12][17][31][24][40].

(1) The first group of features are longitudinal and are considered as time series data. These data were collected at baseline and then regularly every six months up to month 18 (i.e., four entries over 1.5 years, see Fig. 1). In the ADNI, time-series data are medically divided into the following three modalities of (1) Cognitive scores CSs (9 features); (2) neuropsychological battery NSB (51 features); and (3) MRI scans (312 features). The CS features include Alzheimer’s disease assessment scales (ADAS 11 and ADAS 13), clinical dementia rating-sum of boxes (CDRSB), global CDR (CDGLOBAL), functional assessment questionnaire (FAQ), geriatric depression scale (GDTOTAL), MMSE, Montreal cognitive assessment (MoCA), and neuropsychiatric inventory score (NPISCORE). Note that we only use sum scores of different neuropsychological tests to cover the important aspects of those tests [1]. The NSB features include Rey's auditory verbal learning test (RAVLT) features, everyday cognition reports, etc. The imaging data used in our experiments are based on a preprocessed set of T1 weighted MRI features from the ADNI database. The data was preprocessed with the standard ADNI pipeline by a team from the University of California at San Francisco (UCSF), who performed cortical reconstruction and volumetric segmentations with the FreeSurfer image analysis suite according to the atlas generated in Desikan

---

5 Alzheimer’s Disease Neuroimaging Initiative (ADNI): http://www.adni.loni.usc.edu
6 http://surfer.nmr.mgh.harvard.edu/
et al. [47]. The MRI features are associated with regional cortical thickness, regional volume, and surface area measures. Details of the analysis procedure are available online and upon request from the authors. Full details about these features can be found at the ADNI website, the ADNI reference repository, and Supplementary File 2. It is worth noting that PET imaging is also available but has some limitations such as (i) the number of scans is small because the patient is exposed to ionizing radiation, (ii) it has a lower spatial resolution in comparison with MRI, and (iii) it is extremely expensive. Therefore, to build a practical model, we ignore the PET data.

(2) The second group of features is static one-time baseline data collected only once (at the baseline visit). These data covers 64 features, including CSF biomarker (3 features of the amyloid-β peptide from 42 amino acids-Aβ1–42 [ABETA], TAU, and phosphorylated TAU [PTAU]); genetic information (one feature from APOe4); family history questionnaire (5 features from family member dementia history); CSF local lab results (4 features from blood cell counts as well as protein and glucose results); symptoms (27 features from occurrences of nausea, crying, falls, etc.); sociodemographic (5 features from age, marriage status, body mass index [BMI], gender, and the number of years education data); and medical history (19 features from psychiatric, neurologic, cardiovascular, musculoskeletal, gastrointestinal, etc.).

ADNI subjects are 55 to 92 years old, fluent in English or Spanish, and had at least six years of education. At baseline, subjects met the following specific inclusion criteria described in Table 1. They met the diagnostic criteria for probable AD by the National Institute of Neurological and Communicative Disorders and Stroke and the Alzheimer's Disease and Related Disorders Association (NINCDS-ADRDA) [48]. Supplementary File 1 has a complete overview of the ADNI subjects, image acquisition protocols and procedures, and related disorders generally associated with AD [49].

### Table 1: Descriptive statistics for ADNI dataset at baseline.

<table>
<thead>
<tr>
<th>Feature</th>
<th>CN (n=419)</th>
<th>sMCI (n=473)</th>
<th>pMCI (n=140)</th>
<th>AD (n=339)</th>
<th>Combined (n=1371)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gender (M/F)</td>
<td>193/226</td>
<td>283/190</td>
<td>86/54</td>
<td>137/252</td>
<td>747/524</td>
</tr>
<tr>
<td>Age (years)</td>
<td>73.84±0.78</td>
<td>72.92±0.76</td>
<td>73.39±0.84</td>
<td>75.01±0.81</td>
<td>73.82±0.78</td>
</tr>
<tr>
<td>Education (years)</td>
<td>16.41±2.70</td>
<td>15.86±2.95</td>
<td>16.13±2.71</td>
<td>15.53±2.96</td>
<td>15.85±0.90</td>
</tr>
<tr>
<td>FAQ</td>
<td>0.19±0.00</td>
<td>0.18±0.01</td>
<td>0.45±0.04</td>
<td>13.32±0.85</td>
<td>04.54±0.65</td>
</tr>
<tr>
<td>MMSE</td>
<td>28.98±1.14</td>
<td>27.63±1.12</td>
<td>26.49±1.04</td>
<td>23.94±0.84</td>
<td>26.59±0.63</td>
</tr>
<tr>
<td>MoCA</td>
<td>23.6±1.07</td>
<td>25.3±1.02</td>
<td>21.78±0.99</td>
<td>17.48±0.54</td>
<td>22.38±0.04</td>
</tr>
<tr>
<td>ADAS-Cog 11</td>
<td>0.23±0.46</td>
<td>0.20±0.60</td>
<td>0.88±0.71</td>
<td>0.85±0.71</td>
<td>0.96±0.67</td>
</tr>
<tr>
<td>ADAS-Cog 13</td>
<td>0.64±0.23</td>
<td>0.73±0.33</td>
<td>1.20±0.33</td>
<td>1.64±0.74</td>
<td>10.97±0.99</td>
</tr>
<tr>
<td>RAVLT immediate</td>
<td>45.80±9.32</td>
<td>46.41±10.08</td>
<td>29.69±7.08</td>
<td>22.64±7.47</td>
<td>35.20±12.80</td>
</tr>
<tr>
<td>RAVLT total</td>
<td>06.03±0.19</td>
<td>04.57±0.29</td>
<td>03.06±0.28</td>
<td>01.83±0.77</td>
<td>04.19±0.74</td>
</tr>
<tr>
<td>RAVLT forgetting</td>
<td>06.62±0.31</td>
<td>04.42±0.21</td>
<td>04.48±0.19</td>
<td>04.48±0.83</td>
<td>04.23±0.44</td>
</tr>
<tr>
<td>RAVLT % forget</td>
<td>53.47±2.89</td>
<td>55.70±3.17</td>
<td>70.73±28.79</td>
<td>89.46±20.87</td>
<td>57.98±14.71</td>
</tr>
<tr>
<td>CDR</td>
<td>00.08±0.10</td>
<td>00.37±0.06</td>
<td>02.07±0.00</td>
<td>05.34±0.21</td>
<td>01.96±0.24</td>
</tr>
</tbody>
</table>

*Data are mean ± standard deviation.

### 2.3 Model design and development

Fig. 2 illustrates the pipeline associated with our model. As introduced previously, we have two types of data (i.e., time-series data, and static data). The whole pipeline is comprised of seven stages.

- **First**, the entire dataset is randomly split, with 80% (1097 patients) going into the model development set (MDS) and 20% (274 cases) going into the model test set (MTS). The split is stratified, and the process is repeated five times, where we split the whole dataset into training and testing sets randomly. The average results from the five repeated splits are reported. Please note that data division has been done from the first beginning of the machine learning pipeline, which prevents the information leakage issue in model testing.
- **Second**, the MDS data is preprocessed, as discussed in Section 2.3.1. The resulting data preprocessing operators are applied separately on the test set without refitting again on the test set.
- **Third**, a feature extraction step is conducted for the time-series data to calculate the statistically relevant features that best describe each time series. The resulting statistical features can be combined with the static features.
- **Fourth**, training the models with the most informative and discriminative list of features improves the model performance, enhances the model generalizability, and reduces the computation cost and complexity.

---

7http://adni.loni.ucla.edu/methods/mri-tool/mri-analysis/
8https://adni.bitbucket.io/reference/index.html
The feature selection step is implemented prior to the training to identify the most relevant subset of features for our 4-class classification problem. The selected features are marked on the testing set to have the same features as those selected and used after the feature selection process.

- **Fifth**, the data over the four classes is unbalanced, and training with this data is likely to produce biased models [17]. The four classes CN, sMCI, pMCI, and AD, have proportions of 30.56%, 34.50%, 10.21%, and 24.73%, respectively. Namely, the percentage of pMCI is much smaller than CN, sMCI and AD. Therefore, we used the synthetic minority oversampling technique (SMOTE) to deal with the class imbalance in the MDS [49], but SMOTE is not applied to MTS to mimic real-world use. The MDS is further split into training and validation sets using the stratified 10-fold cross-validation (10-fold CV) technique. The resulting sets are used to train ML models to select the best list of features. The literature has confirmed that oversampling by SMOTE is one of the best methods for handling imbalanced datasets [50]. In addition, ML models achieve better performance using balanced datasets [50]. For these reasons, we decided not to train our ML models using imbalanced datasets.

- **Sixth**, MDS is used to train and validate the base ML classifiers based on the stratified 10-fold CV technique. The 10-fold CV ensures unbiased model fitting. The final evaluation of the models is performed using the reserved, unseen MTS. Six of the most popular and diverse ML models (i.e., SVM, RF, DT, XGB, KNN, and MLP) were selected to be trained individually on the MDS sets before all their possible combinations were tested on the MTS, this helped us choose which models were used to build a suitable ensemble classifier[check]. These techniques have been extensively used in the AD domain [17][20]. Models were trained on different modalities before their fusion. The lists of features that achieved the best performance from different modalities or combinations of modalities were used with their candidate classifiers to build the ensemble model.

- **Seventh**, we carried out the aggregation of base learners based on different fusion strategies in this step.
time-series data, we also removed any feature where more than 30% were missing. Any patient cases with missing baseline readings were excluded from the study. Some critical features, such as CSF tau (83%), were missing more than 30% of the time series but were not missing baseline data. We opted to collect these features at baseline and consider them as background data (BG) by combining them with our static data. Moreover, the list of initial potential features was determined according to ADNI recommendations as well as AD diagnosis and progression literature [14]. Then a separate feature selection step was used to further collect significant features from this initial set. For handling non-existing time series values, we followed two sequential strategies according to the intuition of ADNI. First, we filled non-applicable values for every category of data according to ADNI procedures. For example, an ADNI 1 CN patient would not have an MRI scan at visit M18. As a result, we should not consider these types of values to be missing or consider that they are missing not at random. Many lab tests, cognitive tests, and neuroimaging scans are not done for specific diagnoses at specific visits. We followed an accurate procedure to fill these non-applicable values. If the diagnosis had not changed, we used forward filling with the previous values. If the diagnosis had changed, we considered the value as missing. This technique is common in the AD literature [47]. The second step is to determine the missing value from existing data using statistical or ML techniques. We opted for applying a medically intuitive and well-known method. Thus, in the case of numerical data, we used the mean value according to the different classes: CN, sMCI, pMCI, or AD. For categorical features, we used the mode value according to the patient class.

2.3.3 Feature selection

Feature selection is a crucial step to find the most discriminative, informative, and concise subset of features from the available feature set. The selected features are expected to minimize redundancy, avoid overfitting, improve model generality, and reduce computational costs. Relevant features can be selected using filtering, wrapper, or domain expert approaches [52]. Each method has its own advantages and disadvantages [53]. Recommended features by domain experts could enhance model interpretability. Wrapper methods for training and testing classification models are commonly used to assess the importance of different subsets of features in the space of possible feature subsets [54]. These methods typically apply a greedy search algorithm to select the optimum subset of features based on the performance of a specific classification model such as SVM [55], Bagging [56], RF [57], or Naïve Bayes [56]. Filtering methods measure the importance of each feature separately using metrics like information gain or correlation coefficient.
We applied a collection of these methods to select the optimum sets of features from different modalities. More precisely, we utilized the information gain methodology (from the available filtering methods) to select the best features from the Static dataset. From the wrapper methods, we used the recursive feature elimination (RFE) technique combined with RF (RFE-RF) and XGB (RFE-XGB) classifiers to measure feature importance. In the medical domain, studies asserted that RFE-RF is more efficient than other methods such as RFE-SVM to find highly discriminative and small feature sets [17]. We applied RFE-RF and RFE-XGB with 10-CV on the MDS. This 10-fold CV process was repeated 30 times using different stratified and random partitioning of the MDS to avoid any possible bias. RFE searches for the optimum combination of features that maximizes classifier performance using backward feature elimination based on a feature importance metric as the ranking measure. The selected features from all these methods were used to build the ensemble model. In addition, statistical analysis methods were used to analyze the selected features.

### 2.3.4 Ensemble classifier modeling

An ensemble of classifiers is a collection of representative classifiers that are accurate and diverse [35]. All classifiers are trained in tandem, and their decisions are combined according to some strategy. The enhanced performance of the ensemble is due to the ability to combine accurate predictions and correct errors from many diverse classifiers. Thus, getting a good accuracy/diversity trade-off becomes a challenging problem. Homogeneous ensembles (e.g., bagging, boosting, or random forest) utilize a single base classifier and achieve diversity by sampling from training data, assigning different weights to single data instances and/or sampling from the feature set. Heterogeneous ensembles achieve diversity using a pool of distinct base classifiers (e.g., SVM, RF, DT, MLP, etc.). They usually include a form of meta-learning such as techniques called stacking, voting, or ensemble selection. The ensemble selection method includes the following three main steps:

- **Base model generation**: In this step, different models were trained with the data. More precisely, we considered RF, SVM, KNN, XGB, MLP, and DT, as the ML techniques in this study. It is worth noting that even if other ML algorithms could have been used, we chose these ML techniques due to their wide use in the medical domain [58] and to their potentially diverse and complementary predictions.

- **Ensemble pruning**: In this step, a subset of the models previously generated were selected to become members of the ensemble using the accuracy/diversity trade-off as the primary criterion. Maximizing accuracy and diversity produce enhanced stability as a side effect [59]. Diversity should not depend on the classifiers’ outputs but on how they differ and become complementary [46]. General measures like variance and mutual information are not accurate here because they simply measure how members are diverse without employing any information on the correctness of the output [60]. We utilized the so-called “binary oracle method”, which knows the correct answers and evaluates diversity based on whether each classifier is correct or incorrect [61]. Formally, if we take an ensemble \( H = \{c_t, t = 1,2,...C\} \) of \( C \) classifiers to classify \( D = \{(x_1,y_1), (x_2,y_2),...,(x_N,y_N)\} \) of \( N \) samples, \( y_j \in \Omega \) for \( m \) class labels, \( x_j \in \mathbb{R}^n; \) a classifier \( c_t \) assigns a class label from \( \Omega \) to an input example \( x_j \), i.e. \( c_t: \mathbb{R}^n \rightarrow \Omega \), or \( c_t(x_j) \in \Omega, t = 1,2,...C \) is the output of \( t \)th classifier \( c_t \) for the \( j \)th sample \( x_j \), and \( r(x_j) \) is the true label for \( j = 1,2,...N \). For classifiers \( c_1 \) and \( c_2 \), the correctness of classification can be collected in two binary vectors \( v^1 \in \mathbb{R}^N \) and \( v^2 \in \mathbb{R}^N \). For a two-classifier evaluation, \( N^{11} \) denotes the number of times \( c_1 \) and \( c_2 \) are correct, \( N^{00} \) the number of times \( c_1 \) and \( c_2 \) are incorrect, and \( N^{10} \) and \( N^{01} \) the number of times where just one of \( c_1 \) and \( c_2 \) is correct, respectively, and where \( N = N^{11} + N^{10} + N^{01} + N^{00} \). For more than two classifiers, diversity is calculated as the mean of the pairwise values. Single measures of diversity are inadequate [46], so we utilized three measures to test diversity.

The first measure is the **correlation** between the errors calculated by Equation 1, where \( \text{Cov}(\cdot) \) and \( \text{Var}(\cdot) \) are the covariance and variance, respectively. The goal is to minimize the mean pairwise \( \rho \)s.

\[
\rho_{c_1,c_2} = \frac{\text{Cov}(v^1, v^2)}{\sqrt{\text{Var}(v^1)\text{Var}(v^2)}}
\]

(1)

The second measure is the **Q statistic** [62], as defined in Equation 2. The goal is to minimize the mean pairwise \( Q \)s.
Equations 5-9, respectively:

\[ Q_{c_1,c_2} = \frac{N^{11}N^{00} - N^{01}N^{10}}{N^{11}N^{00} + N^{01}N^{10}} \]  

\[ Q_{c_1,c_2} \in [-1, 1] \], and 0 means independent classifiers.

The third measure is the disagreement measure [63] calculated by Equation 3. The goal is to maximize the mean pairwise \( D_{c_1,c_2} \).

\[ D_{c_1,c_2} = \frac{N^{01} + N^{10}}{N} \]  

- **Ensemble integration**: In this step, we pay attention to how to combine single decisions into a unique agreed ensemble decision. The strategy to apply depends on the type of classifier outputs. For each instance \( x_j \), each classifier \( c_i \) outputs discriminant measure \( o_i = c_i(x_j) \), where \( o_i \in \{l_1,l_2...l_m\} \) or \( o_i = [c_{i1}(x_j), c_{i2}(x_j)...c_{im}(x_j)]^T \) and \( c_{ik}(x_j) \) is the predicted probability (or the degree of support) by a classifier \( c_i \) for class \( l_k \) using sample \( x_j \). By combining all classifiers, the decisions are \( O = [o_1,o_2...o_N]^T \), or more generally, as shown in Equation 4.

\[
O = \begin{bmatrix}
    c_{1,1}(x_j) & ... & c_{1,k}(x_j) & ... & c_{1,m}(x_j) \\
    \vdots & \ddots & \vdots & \ddots & \vdots \\
    c_{m,1}(x_j) & ... & c_{m,k}(x_j) & ... & c_{m,m}(x_j)
\end{bmatrix}
\]  

The final decision of the classifier ensemble is a combination of the base learners’ decisions \( H(x_j) = F(c_1(x_j),c_2(x_j),...,c_N(x_j)) = F(O) = F(o_1,o_2...o_N) \). When classifiers return class labels, \( F \) includes: (i) the linear fusion of members’ decisions via majority voting, i.e., \( H(x_j) = \arg\max_k \sum^C_{i=1} c_{i,k}(x_j) \); (ii) weighted majority voting \( H(x_j) = \arg\max_k \sum^C_{i=1} w_i c_{i,k}(x_j) \); (iii) learning based non-linear fusion (e.g., stacking); or (iv) mixture of experts [64]. When classifiers return the degree of certainty, the posterior probability can be calculated as \( P(\cdot|c_i) = o_i \) and \( c_{ik}(x_j) \in o_i \) is defined as before. Without loss of generality, we can let \( o_i \) be normalized, i.e., \( o_i \in [0,1] \); the combination of classifiers is the posterior probability of \( P(l_k|c_1(x_j),c_2(x_j),...,c_N(x_j)) \). Decision rules are defined as follows: (i) product rule, \( \max_{k=1,...,m} \prod^C_{i=1} c_{i,k}(x_j) \); (ii) sum rule, \( \min_{k=1,...,m} \sum^C_{i=1} c_{i,k}(x_j) \); (iii) max rule, \( \max_{k=1,...,m} \max_{l=1,...,C} c_{i,k}(x_j) \); (iv) min rule, \( \max_{k=1,...,m} \min_{l=1,...,C} c_{i,k}(x_j) \); or (v) median rule, \( \max_{k=1,...,m} \frac{1}{C} \sum^C_{i=1} c_{i,k}(x_j) \). In stacking, a meta classifier is used to learn the aggregation function using the base classifiers’ outputs and makes the final decision.

### 2.3.5 Implementation details and performance metrics

We used Python 3.7, Scikit-Learn 0.21.3, mlxtend 0.17.0, and XGBoost 0.90 to generate base classifiers and to build the ensemble models. The performance of the base classifiers and ensemble models was measured using the standard five metrics of accuracy, precision, recall, F1-score, and balanced accuracy (BA), which are defined as follows in Equations 5-9, respectively:

\[ \text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \]  

\[ \text{Precision} = \frac{TP}{TP + FP} \]  

\[ \text{Recall} = \frac{TP}{TP + FN} \]
3. Results and discussion

Designing an accurate and stable ensemble classifier is a challenging task. In our list of experiments, we gradually test different dimensions and tuning methods, and select the optimum path to reach the best design, which is more accurate and medically relevant, as shown in Fig. 3.

1. First, we select the most critical features using the information gain and recursive feature elimination techniques from every individual modality. This step tried to select the best list of features to make the model more accurate and less expensive medically.

2. Having the list of features from each modality, we need to select the best design for the ensemble model. Note that it is well-known in the literature that ensemble classifiers achieve better results than all its base classifiers. We followed two paths to optimize the ensemble design process:
   a. The first path is to fuse all selected features in step 1 (i.e., early fusion strategy). The main idea here is to explore the possible role of the hidden relationships among features of different modalities to improve the model performance. We used the collected list of features to compare the homogeneous classifiers-based methods like bagging and boosting with the heterogeneous ensemble techniques like voting and stacking. The bagging and boosting methods are based on the most accurate base classifier, so we optimize a set of base classifiers, including KNN, decision tree, SVM, etc., to select the best performing technique. The voting and stacking techniques are based on accuracy/diversity trade-off explorations, so we measure both diversity (using Q-statistic, correlation coefficient, and disagreement) and performance using accuracy, precision, recall, and f1 score. To select the best combination of diverse and accurate models, we test many options. The main reason for combining base classifiers is to choose the best number of base classifiers that are most accurate and most diverse, which achieve the best ensemble performance.
   b. The second path is to build a heterogeneous classifier for each different modality and fuse the decisions of these base classifiers using ensemble techniques like stacking or voting (i.e., late fusion). For each modality, we trained six classifiers and selected the best one or a set of most accurate classifiers. In addition, the possible roles are explored for the different combinations of modalities. The role of 13 combinations of AD modalities is evaluated, and each dataset is tested using six machine learning algorithms. This results in 78...
base classifiers. Here, we explored the role of the medical categorization of features and used each base classifier as an expert in a specific domain. For example, the MRI-based classifier is like an expert in making progression detection decisions based on neuroimages.

c. After selecting the best performing ensemble model based on the accuracy and diversity of the base classifiers, we optimized the hyperparameters of the selected stacking ensemble to further enhance its performance.

d. Finally, we tested the performance of the optimized ensemble to solve the 2-class and 3-class classification tasks for Alzheimer’s disease prediction.

3.1 Feature analysis
To train the base models, we have two choices. First, all data modalities can be fused, and a random subset of the features is selected to train each base classifier. Second, it is possible to benefit from the medical relationships among features of each modality and independently train base models using every single modality. The second choice is called the decision fusion paradigm. In this section, we explain how to select the most discriminative features from each modality. These features are used either in the early or late fusion designs.

3.1.1 Static feature analysis
To select the best set of Static features, we applied the information gain filter-based approach. We selected the top 15 (~24%) discriminative features, including ABETA, AGE, APOE4, BCCRYING, BCDPMOOD, BCENERGY, BCINSOMN, BMI, CTRED, CTWHITE, TAU, FHQMOMAD, MHPYCH, PTAU, and PTEDUCAT (see the Supplementary File 2 for further details). The values of these features are statistically significantly different between the four diagnosis groups ($P=0.05$). Fig. 4 shows the feature importance based on the information gain.

![Figure 4: The importance of static features.](image)

3.1.2 Time series feature analysis
To select the best features from the time-series modalities, we followed a three steps methodology. First, we selected the candidate covariates from each modality according to their perceived clinical importance and to our statistical analysis. Second, we extracted statistical features for each modality, as proposed in Section 2. Third, we used feature selection techniques to automatically select the most informative and discriminative features for AD progression detection. It is worth noting that a detailed description of all the features in this paper is provided in Supplementary File 2.

3.1.2.1 Relevant features collection
Regarding CS features: based on the feature selection step, nine features (CSs) are used in this study. As shown in Fig. 5, these scores let us discriminate the four classes very well at each visit. For example, the AD class has the highest values for ADAS 11, ADAS 13, CDRSB, NPISCORE, CDGLOBAL, and FAQ, followed by pMCI, sMCI, and then CN. On the other hand, the AD class has the lowest MMSE and MOCA, followed by pMCI, sMCI, and then CN. For GDTOTAL, both AD and pMCI have comparable values, but sMCI has higher values than CN. In addition, it is of note that the pMCI class has statistically significantly different values collected in visits at M18 compared to at M48 ($P < 0.05$). Collectively, the box plots in Fig. 6(1) illustrate that the four classes have significantly different
values for cognitive scores in the BL, M06, M12, and M18 data ($P<0.001$). In addition, Fig. 6(2) asserts the role of cognitive scores to clearly separate the four classes ($P<0.05$).

Regarding NSB features: we selected 17 features including RAVLT and ADNI MEM among others (see Supplementary File 2). The selected features are medically well known and extensively used in the literature [1]. Due to space restrictions, Fig. 7 shows box plots for only nine of these features based on the BL, M06, M12, and M18 data. As can be seen, all features are statistically significantly different between the four classes ($P<0.01$). Furthermore, these features discriminate the classes in the same manner at M48.

Regarding MRI features: recent studies have asserted that AD progression is tightly correlated with atrophies in the structures of the medial temporal lobe (MTL) [25][65][66]. The MTL includes a set of anatomical regions such as the hippocampus, amygdala, entorhinal, and parahippocampal cortices. Each of these regions has left and right parts. In addition, cortical thickness has a high predictive value for AD progression detection [67][68]. We collected the volume, surface area, and cortical thickness for each of these parts. Out of the 312 MRI features, we selected 43 features related

---

Figure 5: Cognitive scores temporal changes.

Figure 6: Cognitive scores discriminative features.
to volume and cortical thickness of the left part and suitable structures, including HIPPOCAMPUS, AMYGDALA, PARAHIPPOCAMPAL, ENTORHINAL, VENTRICLES, FUSIFORM, INFERIOR/MIDDLE/SUPERIOR TEMPORAL, and INSULA. These features have the highest information gain values. The weights of these features according to the information gain can be found in Supplementary File 2, Fig. S1. The most important features are the left and right volumes of the hippocampus and the cortical thickness of the left entorhinal. Fig. 8 illustrates the top 10 volumes (left part) and top 10 cortical thicknesses (right part). As can be seen, the selected features can clearly discriminate different classes, especially CN vs. AD \( P<0.05 \). The discriminative power of the selected biomarkers is on par with results from the literature [1][69][70][71][72].

**Figure 7:** Neuropsychological battery discriminative features.

**Figure 8:** Volume and cortical thickness of the most important MRI features.

### 3.1.2.2 Statistical features extraction

In this step, we convert time series data into a suitable format for conventional ML models. As discussed in Section 2, we have two strategies to extract features from time-series data, including baseline data and statistical data. Statistical features are collected from the four readings (BL, M06, M12, and M18) associated with each patient. The calculated features include the maximum, minimum, average, and standard deviation. These feature sets are made up of 36, 68, and 172 features from CSs, NSB, and MRI, respectively.

### 3.1.2.3 Time series feature selection

As discussed in the previous section, we extracted many features from every modality; however, not all of these are discriminative and helpful in AD progression detection. Selecting relevant features improves model performance and simplicity, reduces measurement costs, and enhances model understandability [73]. As a result, a feature selection step is crucial. Feature selection was based on the pre-defined model development set (MDS) (80% of the entire dataset). The ten fitted and tested RFE-RF models selected 20 features (55.5% of the total features) for the CS modality, and the RFE-XGB models selected 25 features (69.4% of the entire feature set). We found 18 features recommended by both types of models. The final set of features consists of the union of the sets provided by both methods (i.e., 27 features). Regarding the MRI modality, RFE-RF recommended 64 features, and RFE-XGB recommended 29 features.
We found that RFE-RF also recommended 65.5% of the RFE-XGB features. We used the combination of these features (i.e., 74 features, 43% of the total feature set). For the NSB modality, RFE-RF selected 16 features, and RFE-XGB selected 12 features. We found 11 features recommended by both methods. We combined both sets and obtained 25 features (36.8% of the complete NSB set). The selected features are ranged from maximum, minimum, standard deviation, and average. A full description of these features can be found in Supplementary File 2. We reevaluated the feature weights using the information gain filter method to confirm that the selected features are statistically significant. We found the selected features had the highest weights for all modalities.

3.2 Design of ensemble classifiers

After selecting the best features associated in each modality, we implemented a set of experiments to choose the best design scheme for the AD detection ensemble model. The first set of experiments in Section 3.2.1 fuses all modalities into one source (early fusion) and uses the resulting set to build an ensemble model. In this design, we depend on the hidden relationships among features of different modalities. The second set of experiments, discussed in Section 3.2.2, builds separate classifiers for every modality and their different integrations and fuses the decision of all base classifiers to make the final decision (i.e., late fusion). In the second design method, we independently utilized the medical relationships among features in each data source by building a separate base classifier for each data type.

3.2.1 Early fusion-based ensemble modeling

The fused data set has 141 features from the four modalities. The first experiment compared the homogeneous methods of bagging and boosting with the heterogeneous ensemble techniques of voting and stacking. On the one hand, the bagging and boosting methods are based on the most accurate base classifier. On the other hand, the accuracy/diversity trade-off is examined for the voting and stacking methods (see Fig. 9). In other words, we built the models with the most accurate base classifiers and with the most diverse base classifiers to assess the effect of both measures (i.e., accuracy and diversity) on the ensemble performance. The individual rate is the minimum error rate of a classifier, as shown in equation 5, this is used as a performance metric in addition to the other metrics of precision, recall, and F1-score.

$$\text{Minimum error rate} = \min_k \left( \frac{1}{m} \sum_{j=1}^{m} e_k(j) \right)$$

where $e_k(j)$ is the classification error of classifier $k$ for data $j$. We examined various diversity measures. This experiment was aimed at finding the optimum set of classifiers with the highest performance and diversity.

Base classifiers were fitted using the MDS with the stratified 10-CV technique. To get more stable results, each experiment was repeated 30 times and the averaged results are reported here. Then, the performance of the model was measured using the unseen MTS test set, as shown in Table 2. For a fairer comparison, the base classifiers were trained using the default parameters of Scikit-learn [74]. It is of note that the RF classifier achieved the highest results (Accuracy=84.95%, Precision=84.88%, Recall=84.95%, F1-score=84.87%, and BA=79.38%), while KNN had the worst results (Accuracy=74.51%, Precision=83.77%, Recall=74.51%, F1-score=74.48%, and BA=78.43%). Both XGB and RF are ensemble models, and, as expected, they achieved the best two results.

<table>
<thead>
<tr>
<th>Classifier index</th>
<th>Base classifier</th>
<th>Performance measurement (%)</th>
<th>Error rate</th>
<th>Classifier rank</th>
<th>Diversity measurement</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RF</td>
<td>84.95</td>
<td>84.88</td>
<td>84.95</td>
<td>84.87</td>
</tr>
<tr>
<td>2</td>
<td>SVM</td>
<td>83.98</td>
<td>84.08</td>
<td>83.98</td>
<td>84.08</td>
</tr>
<tr>
<td>3</td>
<td>KNN</td>
<td>74.51</td>
<td>83.77</td>
<td>74.51</td>
<td>74.48</td>
</tr>
<tr>
<td>4</td>
<td>XGB</td>
<td>84.46</td>
<td>84.07</td>
<td>84.47</td>
<td>84.25</td>
</tr>
<tr>
<td>5</td>
<td>DT</td>
<td>77.43</td>
<td>78.21</td>
<td>77.43</td>
<td>77.66</td>
</tr>
<tr>
<td>6</td>
<td>MLP</td>
<td>82.28</td>
<td>82.69</td>
<td>82.28</td>
<td>82.57</td>
</tr>
</tbody>
</table>

Figure 9: Early fusion-based ensemble selection.
Table 2 also shows the average of the pairwise diversities for each base classifier. We calculated diversity using three popular techniques: Q-statistic, correlation coefficient, and disagreement. It should be noted that all these diversity measures sorted the base classifiers in the same order. KNN is the most diverse classifier, but it has the worst performance while SVM has the lowest diversity. To study the effect of the diversity/accuracy trade-off on classifier subset selection, we performed experiments using voting and stacking ensemble techniques. We used soft voting because it achieves better results than hard voting. As selecting the optimum subset of classifiers requires an exhaustive search among $2^C + 1$ possible combinations, we used a simple greedy method to select the optimum (i.e., the combination with the best balance between diversity and accuracy) subset of base classifiers that work well together, we started with an ensemble size of $n = 2$. The results were compared with the bagging and AdaBoost techniques using RF (i.e., the most accurate classifier) as the base estimator. Bagging and AdaBoost were expected to generate diverse classifiers [46].

Regarding the voting technique, Table 3 shows how just putting together all the base classifiers does not achieve the best performance. These results are most likely because of the insufficient diversity and poor performance in the learner pool. RF and XGB are the two most accurate classifiers, and together they achieve an accuracy of 85.44%, precision of 85.00%, recall of 85.44%, F1-score of 85.19%, and BA of 78.92%. However, using the two most diverse techniques (i.e., KNN and DT), we achieved an accuracy of 78.16%, precision of 81.55%, recall of 78.16%, F1-score of 85.19%, and BA of 78.92%. As confirmed by the literature, paying attention to only diversity is likely to produce poorer performance [75] [76].

![Table 3: The effect of diversity and performance on voting and stacking ensembles.](image)

Accordingly, diversity must be balanced with accuracy to create a well-performing ensemble [46]. Using the most accurate three classifiers (i.e., RF, XGB, and SVM), we achieved an accuracy of 85.19%, precision of 80.99%, recall of 80.99%, F1-score of 80.99%, and BA of 78.74%. However, this ensemble does not improve on the performance of RF and XGB. This is because SVM has the lowest diversity level and so did not add value to the ensemble. This conclusion is confirmed when using RF and XGB with KNN. Although KNN is less accurate than SVM, it adds more accuracy to the ensemble. This is because KNN has the highest diversity level. The combination of RF, XGB, and KNN achieved the best results (i.e., Accuracy=86.17%, Precision=88.10%, Recall=86.17%, F1-score=86.67%, and BA=84.41%). By repeating the most accurate classifier (i.e., RF) twice, the ensemble achieved better results (i.e., Accuracy=86.65 %, Precision=88.16%, Recall=86.65%, F1-score=87.06, and BA=84.81). By repeating the RF
classifier experiment, we added more weight to this classifier. However, repeating RF more than twice had no effect on the performance. On the other hand, repeating the KNN classifier did not enhance the classifier performance. We checked the different combinations, and none achieved better results. Our vote ensembles give the same weight to all base classifiers. Nevertheless, some base classifiers can be more important than others. Alternatively, the stacking ensemble uses a meta classifier to learn the optimum combination of base classifiers.

Regarding the stacking technique: we selected logistic regression (LR) to play the role of the meta classifier. The utilized decisions of the base classifiers are represented in the form of probability distributions. For example, for a specific training example \( x_j \), in a 3-class setting \([l_1, l_2, l_3]\) with two level-0 classifiers \(c_1, c_2\), these base classifiers might make probability predictions of \(c_1:0.2, 0.5, 0.3\) and \(c_2:0.3, 0.4, 0.4\). The level-1 meta classifier receives \(x_j = [0.2, 0.5, 0.3, 0.3, 0.4, 0.4]\). The most two accurate base classifiers achieve performance of: Accuracy=84.93%, Precision=84.17%, Recall=84.93%, F1-score=84.51%, and BA=77.32%. However, the two most diverse classifiers achieve performance of: Accuracy=79.89%, Precision=80.65%, Recall=79.89%, F1-score=80.19%, and BA=74.23%. The stacked ensemble shows the same behavior as the voting ensemble method. If we concentrate on the accuracy of the models, then removing the worst classifiers does not significantly enhance the performance. The same for diversity, where removing the least diverse classifiers sometimes degrades the performance if the removed classifiers are highly accurate. Thus, the best solution is the one yielding the best balance between accuracy and diversity. This is proved when combining the most accurate classifiers (RF + XGB) and the most diverse classifier (KNN). In this case, the ensemble achieves the best performance: Accuracy=85.51%, Precision=85.13%, Recall=85.51%, F1-score=85.29%, and BA=78.84%. The main difference is that the stacking method achieved better performance than voting when we used all base classifiers (i.e., Accuracy=85.29%, Precision=84.59%, Recall=85.29%, F1-score=84.91%, and BA=77.92%). We note that adding such diversity to the ensemble does not improve performance. This means that diversity enforces regularization for the ensemble because the performance stops improving when there is no more diversity to extract from the pool of base classifiers [75][61].

Both AdaBoost and Bagging ensemble techniques were checked using the most accurate and diverse models. AdaBoost does not achieve good performance using RF and DT as the base classifiers. Bagging on the other hand achieves higher performance than stacking using RF as the base classifier (i.e., Accuracy=85.75%, Precision=86.24%, Recall=85.75%, F1-score=85.95%, and BA=81.77%). Note that all other parameters for both AdaBoost and Bagging were set to the default Scikit-Learn values. We conclude that the performance of the base classifiers has much more effect on the ensemble performance than diversity.

3.2.2 Late fusion-based ensemble modeling

Building an ensemble based on a set of heterogeneous classifiers trained using distinct feature sets often produces an accurate model because by considering heterogeneous base classifiers, they are likely to have complementary decisions and uncorrelated errors [36]. In this experiment, we exploited the medical relationships among the features of each of our four modalities by training a separate classifier for each one. In addition, we tested different combinations of modalities to check the effect of removing noise in different datasets on classifier performance. We implemented six classifiers for each modality and for their different combinations. The selection criteria for individual classifiers were mainly based on their performance and diversity.

![Figure 10: Ensemble model construction steps.](image-url)
For each dataset, we selected the optimum classifier or set of classifiers. Table 10 illustrates the detailed process for creating a customized ensemble model, where an exhaustive search is used to find the best representative models for each data type. It is worth noting that because the number of options is not significant when checking for the best model, we discarded here the use of heuristic optimization techniques such as multi-objective optimization techniques.

First, we trained the six ML models with every modulation and different combinations of modalities. We used the default hyperparameter settings from Scikit-Learn for all ML models. Second, we selected the optimum set of base classifiers, which are the most accurate and diverse. Third, the final ensemble decision is computed as the fusion of the single decisions made by the selected classifiers. We tested both voting and stacking as fusion strategies.

### Table 4: The performance of the base models for different modalities and their combinations.

<table>
<thead>
<tr>
<th>Modality</th>
<th>#</th>
<th>Model</th>
<th>Performance (%)</th>
<th>Overall Diversity</th>
<th>Diversity of each data set</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSB</td>
<td>1</td>
<td>RF</td>
<td>79.96</td>
<td>0.3175</td>
<td>0.3190</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>SVM</td>
<td>71.84</td>
<td>0.2066</td>
<td>0.1333</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>MLP</td>
<td>77.88</td>
<td>0.3915</td>
<td>0.3766</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>KNN</td>
<td>68.93</td>
<td>0.2628</td>
<td>0.2409</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>DT</td>
<td>72.49</td>
<td>0.2947</td>
<td>0.2187</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>MLP</td>
<td>73.99</td>
<td>0.2846</td>
<td>0.1286</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>SVM</td>
<td>76.94</td>
<td>0.3706</td>
<td>0.1591</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>XGB</td>
<td>80.83</td>
<td>0.3227</td>
<td>0.1947</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>GBC</td>
<td>85.89</td>
<td>0.3420</td>
<td>0.1612</td>
</tr>
<tr>
<td></td>
<td>10</td>
<td>KNN</td>
<td>74.83</td>
<td>0.3402</td>
<td>0.1393</td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>DT</td>
<td>77.64</td>
<td>0.3722</td>
<td>0.2147</td>
</tr>
<tr>
<td></td>
<td>12</td>
<td>RF</td>
<td>77.62</td>
<td>0.3355</td>
<td>0.1551</td>
</tr>
<tr>
<td></td>
<td>13</td>
<td>MLP</td>
<td>75.57</td>
<td>0.2842</td>
<td>0.1466</td>
</tr>
<tr>
<td></td>
<td>14</td>
<td>SVM</td>
<td>70.76</td>
<td>0.2906</td>
<td>0.1616</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>XGB</td>
<td>83.48</td>
<td>0.3236</td>
<td>0.1805</td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>KNN</td>
<td>70.48</td>
<td>0.2528</td>
<td>0.1500</td>
</tr>
<tr>
<td></td>
<td>17</td>
<td>DT</td>
<td>67.09</td>
<td>0.1497</td>
<td>0.0873</td>
</tr>
<tr>
<td></td>
<td>18</td>
<td>MLP</td>
<td>48.01</td>
<td>0.1959</td>
<td>0.0840</td>
</tr>
<tr>
<td></td>
<td>19</td>
<td>Stree</td>
<td>43.54</td>
<td>0.1258</td>
<td>0.0708</td>
</tr>
<tr>
<td></td>
<td>20</td>
<td>SVM</td>
<td>52.67</td>
<td>0.2970</td>
<td>0.1335</td>
</tr>
<tr>
<td></td>
<td>21</td>
<td>XGB</td>
<td>75.25</td>
<td>0.2941</td>
<td>0.1413</td>
</tr>
<tr>
<td></td>
<td>22</td>
<td>KNN</td>
<td>47.74</td>
<td>0.2941</td>
<td>0.1413</td>
</tr>
<tr>
<td></td>
<td>23</td>
<td>DT</td>
<td>48.33</td>
<td>0.3068</td>
<td>0.1508</td>
</tr>
<tr>
<td></td>
<td>24</td>
<td>MLP</td>
<td>48.36</td>
<td>0.3559</td>
<td>0.1990</td>
</tr>
<tr>
<td></td>
<td>25</td>
<td>Stree</td>
<td>53.45</td>
<td>0.3819</td>
<td>0.2225</td>
</tr>
<tr>
<td></td>
<td>26</td>
<td>SVM</td>
<td>52.67</td>
<td>0.4356</td>
<td>0.2322</td>
</tr>
<tr>
<td></td>
<td>27</td>
<td>XGB</td>
<td>68.64</td>
<td>0.2584</td>
<td>0.1375</td>
</tr>
<tr>
<td></td>
<td>28</td>
<td>KNN</td>
<td>54.37</td>
<td>0.3563</td>
<td>0.2137</td>
</tr>
<tr>
<td></td>
<td>29</td>
<td>DT</td>
<td>60.87</td>
<td>0.4687</td>
<td>0.2922</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>MLP</td>
<td>49.17</td>
<td>0.5122</td>
<td>0.3006</td>
</tr>
<tr>
<td></td>
<td>31</td>
<td>Stree</td>
<td>79.12</td>
<td>0.5806</td>
<td>0.3411</td>
</tr>
<tr>
<td></td>
<td>32</td>
<td>SVM</td>
<td>56.07</td>
<td>0.4102</td>
<td>0.2323</td>
</tr>
<tr>
<td></td>
<td>33</td>
<td>XGB</td>
<td>81.05</td>
<td>0.5242</td>
<td>0.3043</td>
</tr>
<tr>
<td></td>
<td>34</td>
<td>KNN</td>
<td>50.00</td>
<td>0.3730</td>
<td>0.2230</td>
</tr>
<tr>
<td></td>
<td>35</td>
<td>DT</td>
<td>74.30</td>
<td>0.4643</td>
<td>0.2332</td>
</tr>
<tr>
<td></td>
<td>36</td>
<td>MLP</td>
<td>70.99</td>
<td>0.5177</td>
<td>0.3024</td>
</tr>
<tr>
<td></td>
<td>37</td>
<td>Stree</td>
<td>79.63</td>
<td>0.5180</td>
<td>0.3073</td>
</tr>
<tr>
<td></td>
<td>38</td>
<td>SVM</td>
<td>52.69</td>
<td>0.4436</td>
<td>0.2376</td>
</tr>
<tr>
<td></td>
<td>39</td>
<td>XGB</td>
<td>82.52</td>
<td>0.4950</td>
<td>0.3021</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>KNN</td>
<td>48.06</td>
<td>0.3384</td>
<td>0.2040</td>
</tr>
<tr>
<td></td>
<td>41</td>
<td>DT</td>
<td>77.64</td>
<td>0.4871</td>
<td>0.2822</td>
</tr>
<tr>
<td></td>
<td>42</td>
<td>MLP</td>
<td>66.16</td>
<td>0.4298</td>
<td>0.2498</td>
</tr>
<tr>
<td></td>
<td>43</td>
<td>Stree</td>
<td>75.42</td>
<td>0.4672</td>
<td>0.2949</td>
</tr>
<tr>
<td></td>
<td>44</td>
<td>SVM</td>
<td>55.95</td>
<td>0.4240</td>
<td>0.2497</td>
</tr>
<tr>
<td></td>
<td>45</td>
<td>XGB</td>
<td>63.83</td>
<td>0.4611</td>
<td>0.2976</td>
</tr>
<tr>
<td></td>
<td>46</td>
<td>KNN</td>
<td>48.11</td>
<td>0.4972</td>
<td>0.2976</td>
</tr>
<tr>
<td></td>
<td>47</td>
<td>DT</td>
<td>63.09</td>
<td>0.4139</td>
<td>0.2537</td>
</tr>
<tr>
<td></td>
<td>48</td>
<td>MLP</td>
<td>67.22</td>
<td>0.4100</td>
<td>0.2537</td>
</tr>
<tr>
<td></td>
<td>49</td>
<td>Stree</td>
<td>50.47</td>
<td>0.3282</td>
<td>0.2105</td>
</tr>
<tr>
<td></td>
<td>50</td>
<td>SVM</td>
<td>54.04</td>
<td>0.3083</td>
<td>0.1975</td>
</tr>
<tr>
<td></td>
<td>51</td>
<td>XGB</td>
<td>65.05</td>
<td>0.4162</td>
<td>0.2983</td>
</tr>
<tr>
<td></td>
<td>52</td>
<td>KNN</td>
<td>49.53</td>
<td>0.3699</td>
<td>0.2491</td>
</tr>
<tr>
<td></td>
<td>53</td>
<td>DT</td>
<td>51.22</td>
<td>0.3348</td>
<td>0.2243</td>
</tr>
<tr>
<td></td>
<td>54</td>
<td>MLP</td>
<td>49.88</td>
<td>0.1996</td>
<td>0.1047</td>
</tr>
<tr>
<td></td>
<td>55</td>
<td>Stree</td>
<td>43.25</td>
<td>0.2450</td>
<td>0.1447</td>
</tr>
<tr>
<td></td>
<td>56</td>
<td>SVM</td>
<td>55.95</td>
<td>0.2422</td>
<td>0.1447</td>
</tr>
<tr>
<td></td>
<td>57</td>
<td>XGB</td>
<td>63.83</td>
<td>0.2422</td>
<td>0.1447</td>
</tr>
<tr>
<td></td>
<td>58</td>
<td>KNN</td>
<td>48.94</td>
<td>0.2422</td>
<td>0.1447</td>
</tr>
<tr>
<td></td>
<td>59</td>
<td>DT</td>
<td>51.22</td>
<td>0.2422</td>
<td>0.1447</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>MLP</td>
<td>49.88</td>
<td>0.2422</td>
<td>0.1447</td>
</tr>
<tr>
<td></td>
<td>61</td>
<td>Stree</td>
<td>43.25</td>
<td>0.2422</td>
<td>0.1447</td>
</tr>
<tr>
<td></td>
<td>62</td>
<td>SVM</td>
<td>55.95</td>
<td>0.2422</td>
<td>0.1447</td>
</tr>
<tr>
<td></td>
<td>63</td>
<td>XGB</td>
<td>63.83</td>
<td>0.2422</td>
<td>0.1447</td>
</tr>
</tbody>
</table>

**Abbreviations:** Acc, accuracy; Pre, precision; Recall, recall; F1, F1-score; Q, Q-statistic; Corr, correlation coefficient; Dis., disagreement.
We evaluated the role of 13 combinations of AD modalities, and each dataset was tested using the six ML algorithms under study. This resulted in 78 base classifiers. Table 4 shows the testing results of these classifiers using the MTS data. It should be noted that the XGB classifier achieves the best performance in all experiments. In addition, the CSs turned out to have the highest performance compared with other single modalities. As a result, its combination with other modalities usually improves the performance. The optimum results were achieved by excluding the MRI data, i.e., the combination of CSs, NSB, and Static (Accuracy=85.92%, Precision=86.95%, Recall=85.92%, F1-score=86.37%, and BA=81.11%). In addition, the fusion of CSs and NSB achieves results not far from those reported by the above CSs, NSB, and Static combination (Accuracy=85.68%, Precision=86.37%, Recall=85.68%, F1-score=86%, and BA=80.17%).

Notice that, the addition of Static modalities improves the performance (Accuracy +0.24%, Precision +0.58%, Recall +0.24%, F1-score +0.37%, and BA +0.94%). The average of pairwise diversities for the base classifiers is shown in Table 4. Diversity was calculated among the classifiers of every dataset and the 78 classifiers. Both performance and diversity of the base classifiers were used to select ensemble members. Note that the same base classifier and/or the same dataset could be selected more than once. We used a greedy method to select the optimal combination. We started by fusing two base models. To build a voting ensemble, we used soft voting because, in most cases, it achieves better results [77]. In addition, every experiment was repeated 30 times to get more stable results. Table 5 shows the testing results of different ensembles. We implemented both voting and stacking ensembles, we start by analyzing the results of the voting models. Note that the implemented customized ensembles are based on the accuracy and diversity of the base classifiers.

First, we analyzed the decision fusion of base classifiers trained using single modalities (i.e., models 1 to 24 in Table 4). These models were trained independently using CSs, NSB, MRI, or Static data. The two most accurate base models were 7 (CSs RF) and 9 (CSs XGB). A voting classifier based on these two models improved their individual performance (Accuracy=85.57%, Precision=86.17%, Recall=85.57%, F1-score=85.71%, and BA=81.02%). However, replacing model 7 by model 3 (NSB XGB), which is less accurate but more diverse, improves the performance of this voting ensemble (Accuracy=87.38%, Precision=87.67%, Recall=87.38%, F1-score=87.43%, and BA=82.61%). Adding model 21 to the 3-9 ensemble achieved better performance than the previous early fusion scheme (Accuracy=87.86%, Precision=87.74%, Recall=87.86%, F1-score=87.73%, and BA=82.98%). We note that replacing model 21 with the most diverse model (model 23) degrades the performance of the previous 3-9-21 ensemble. This is expected because the three models (3, 9, and 21) are trained using different feature sets (i.e., CSs, NSB, and Static). Therefore, they are already diverse and replacing a more accurate model with a less accurate one would surely harm the results. In other words, the DT model (23) did not add to the diversity of the ensemble. Other combinations have been checked, but no ensemble beats the 3-9-21 model.

Table 5: Customized ensemble selection.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Fusion strategy</th>
<th>Classifiers</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Accuracy (%)</td>
</tr>
<tr>
<td>CSs, CSs</td>
<td>Vote: 2 most accurate</td>
<td>9, 22</td>
</tr>
<tr>
<td></td>
<td>Vote: 2 most inaccurate</td>
<td>12, 13</td>
</tr>
<tr>
<td></td>
<td>Stack: 2 most accurate</td>
<td>27, 29</td>
</tr>
<tr>
<td></td>
<td>Stack: 2 most inaccurate</td>
<td>28, 29</td>
</tr>
<tr>
<td></td>
<td>Stack: 2 most accurate + 1 more diverse</td>
<td>29, 30, 31</td>
</tr>
<tr>
<td></td>
<td>Stack: 2 most inaccurate + 1 more diverse</td>
<td>28, 29, 31</td>
</tr>
<tr>
<td></td>
<td>Stack: 2 most accurate + 1 most diverse</td>
<td>28, 29, 31</td>
</tr>
<tr>
<td></td>
<td>Stack: 2 most inaccurate + 1 most diverse</td>
<td>28, 29, 31</td>
</tr>
<tr>
<td></td>
<td>Stack: 2 most accurate + 1 accurate</td>
<td>28, 29, 31</td>
</tr>
<tr>
<td></td>
<td>Stack: 2 most inaccurate + 1 accurate</td>
<td>28, 29, 31</td>
</tr>
<tr>
<td></td>
<td>Stack: 2 most accurate + 1 more diverse and 1 more accurate</td>
<td>29, 30, 31, 32</td>
</tr>
<tr>
<td></td>
<td>Stack: 2 most inaccurate + 1 more diverse and 1 more accurate</td>
<td>28, 29, 31, 32</td>
</tr>
<tr>
<td></td>
<td>Stack: 2 most accurate + 1 most diverse and 1 more accurate</td>
<td>28, 29, 31, 32</td>
</tr>
<tr>
<td></td>
<td>Stack: 2 most inaccurate + 1 most diverse and 1 more accurate</td>
<td>28, 29, 31, 32</td>
</tr>
<tr>
<td></td>
<td>Stack: 2 most accurate + 1 accurate and 1 more accurate</td>
<td>28, 29, 31, 32</td>
</tr>
<tr>
<td></td>
<td>Stack: 2 most inaccurate + 1 accurate and 1 more accurate</td>
<td>28, 29, 31, 32</td>
</tr>
<tr>
<td></td>
<td>Stack: 2 most accurate + 1 more diverse and 1 accurate</td>
<td>29, 30, 31, 32</td>
</tr>
<tr>
<td></td>
<td>Stack: 2 most inaccurate + 1 more diverse and 1 accurate</td>
<td>28, 29, 31, 32</td>
</tr>
<tr>
<td></td>
<td>Stack: 2 most accurate + 1 most diverse and 1 accurate</td>
<td>28, 29, 31, 32</td>
</tr>
<tr>
<td></td>
<td>Stack: 2 most inaccurate + 1 most diverse and 1 accurate</td>
<td>28, 29, 31, 32</td>
</tr>
<tr>
<td></td>
<td>Stack: 2 most accurate + 1 accurate and 1 more accurate</td>
<td>28, 29, 31, 32</td>
</tr>
<tr>
<td></td>
<td>Stack: 2 most inaccurate + 1 accurate and 1 more accurate</td>
<td>28, 29, 31, 32</td>
</tr>
</tbody>
</table>

Example entries:
- [CSs, NSB] + [CSs, Static]: Accuracy=87.14%, Precision=88.72%, Recall=87.41%, F1-score=87.88%, BA=83.07%
- [CSs, NSB] + [CSs, MR], [CSs, Static]: Accuracy=87.14%, Precision=88.72%, Recall=87.41%, F1-score=87.88%, BA=83.07%
- [CSs, NSB] + [CSs, MR], [CSs, Static]: Accuracy=87.14%, Precision=88.72%, Recall=87.41%, F1-score=87.88%, BA=83.07%
Second, we analyze the decision fusion of base classifiers trained using two fused modalities (i.e., we pay attention here to models 25 to 60 in Table 4). The XGB classifiers achieved the best results, especially model 27 (Accuracy=85.68%, Precision=86.37%, Recall=85.68%, F1-score=86.00%, and BA=80.17%). The fusion of this model (27) and the second-most accurate model (39) improves the performance (Accuracy=86.89%, Precision=87.56%, Recall=86.89%, F1-score=87.10%, and BA=83.72%). However, replacing the most accurate model 27 with a less accurate one that is more diverse, like model 51, further improves the performance (Accuracy=88.35%, Precision=88.58%, Recall=88.35%, F1-score=88.46%, and BA=83.55%). In addition, the NSB data are utilized by both models 27 and 51. The role of diversity is also confirmed by replacing model 51 with a very weak model (45), and the resulting ensemble achieved even better performance than model 27 alone (Accuracy=86.65%, Precision=87.17%, Recall=86.65%, F1-score=86.89%, BA=81.69%). As we increase diversity, the resulting vote classifier achieves better results, as shown in Table 5. For example, in model 28, the KNN based classifier only has an accuracy of 54.37% but it has a high diversity level, as explained before, with models 27-51. We checked the fusion of this model and other accurate or diverse models. The diversity has shown a better effect to improve the performance, as in 67-69 models. All other combinations did not achieve better results.

Third, we analyzed the decision fusion of base classifiers trained using three fused modalities. As expected, the integrated CSs, NSB, and Static dataset achieved the best results with the XGB classifier (Accuracy=85.92%, Precision=86.95%, Recall=85.92%, F1-score=86.37%, and BA=81.11%). Note that we achieved better results using the late fusion of the same modalities, as explained before, with models 27-51. We checked the fusion of this model and other accurate or diverse models. The diversity has shown a better effect to improve the performance, as in 67-69 models. All other combinations did not achieve better results.

Next, we study the results of stacking models. Based on the base modalities, the stacking of the most accurate three models, 3, 9, and 21, achieved better results than the equivalent voting model (Accuracy=88.11%, Precision=87.55%, Recall=88.11%, F1-score=87.78%, and BA=81.46%). Note that combining more accurate base models (7, 9, and 11) generates a less accurate ensemble because these models have lower diversity than 3, 9, and 21. Like voting, increasing the number of base models increases the computational ensemble time, but it does not improve the results of stacking models. We checked on the role of diversity in enhancing the ensemble performance. The best stacking model is the result of fusing the three models 27, 39, and 51, which are based on [CSs, NSB], [CSs, Static], and [NSB, Static], respectively (Accuracy=88.33%, Precision=88.72%, Recall=88.83%, F1-score=88.77%, and BA=83.59%). It is worth noting that these results are better than all voting models.

These results indicate that the customized stacking model based on the decision fusion of heterogeneous data achieved the most accurate results. The XGB classifiers were used to train three base models. Then, a stacking ensemble model was used to fuse the decisions of these three base models using LR as the meta learner. It can be noticed that CSs and NSB are early fused and used by one XGB classifier. In addition, the Static data are fused with both CSs and NSB in two different datasets and used to train the other two XGB classifiers. As a result, each modality is used twice.

3.2.3 Optimizing selected model

The literature asserts that the performance of base learners must be optimized before using ensemble models [78]. This optimization can be achieved either using the base learners in meta classifiers such as voting or stacking or by hyperparameter optimization [79], but not necessarily using both. In the previous sections, we considered the default settings of the XGB classifiers. In this experiment, we check the effect of optimizing the performance of the base learners on the overall ensemble results. Based on the results of the previous experiments, we optimized an XGB base classifier for each dataset of [CSs, NSB], [CSs, Static], and [NSB, Static]. A randomized search with a stratified 10-CV strategy was used to find the best hyperparameters for every XGB classifier. The classifiers were fitted with the MDS and stratified with 10-CV. Each experiment was repeated 30 times, the average performance of each model was measured using the unseen MTS test set. Next, we combined these optimized models to train the stacking ensemble model. Optimization results are presented in Table 6 and exhaustive lists of the hyperparameters are in Table S2 of Supplementary File 2. As can be seen, optimizing the hyperparameters slightly improves the performance of the base
classifiers. This enhanced performance of the base classifiers has no big effect on the overall performance of the stacking model. This is probably because the diversity of the base classifiers has not been improved.

| Table 6: Optimized performance of base XGB models and stacking model. |
|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|
| Model                   | Hyperparameters | Accuracy (%) | Precision (%) | Recall (%) | F1-score (%) | BA (%) |
| [CSs, NSB] model          | Default          | 85.88        | 86.37         | 85.00      | 86.00         | 88.11 |
|                          | Optimized        | 87.87        | 88.52         | 87.50      | 88.13         | 86.04 |
| [CSs, Static] model       | Default          | 82.52        | 83.21         | 82.52      | 82.91         | 77.16 |
|                          | Optimized        | 83.21        | 83.21         | 83.11      | 83.12         | 77.39 |
| [NSB, Static] model       | Default          | 88.56        | 90.24         | 88.66      | 85.66         | 75.43 |
|                          | Optimized        | 89.15        | 91.11         | 89.16      | 89.87         | 84.44 |

3.2.4 Testing optimized model on 3-class and 2-class problems

The previously trained models were used to solve a 4-class classification problem (i.e., CN vs. sMCI vs. pMCI vs. AD). This is a problematic classification task [6][9]. The main challenge is to identify the pMCI patients. As discussed in Section 3.1 Statistical Analysis, the pMCI cases have similar properties to the AD class at M48. In fact, they will be AD patients at M84, but from BL to M18, they are MCI patients with worse conditions, see Figures 4, 5, 6, and 7. As a result, these cases induce noise to the classification algorithms because they are still MCI, but classifiers have to discriminate them as a different category.

In this experiment, we test our proposal regarding the 3-class classification problem (i.e., CN vs. MCI vs. AD). We consider pMCI and AD to be in the same category, sMCI as just stable MCI, and CN as it is. In addition, we test our approach in a 2-class classification problem (i.e., NOT AD vs. AD). In practice, we check all possible combinations between two classes, namely CN vs. AD (AD is the positive class), CN vs. pMCI (pMCI is the positive class), CN vs. sMCI (sMCI is the positive class), AD vs. sMCI (AD is the positive class), AD vs. pMCI (AD is the positive class), and sMCI vs. pMCI (pMCI is the positive class). Notice that we set the class nearer to AD as the positive class. In addition, we check the binary classification problem when combining both CN and sMCI as class 0 and pMCI and AD as class 1.

As reported in Table 7, both base and ensemble classifiers achieve higher performance compared to the 4-class problem. It is worth noting that the used modalities and base classifiers in this experiment are based on the results of the previous experiment. For the 3-class problem, the stacking classifier achieved an accuracy of 96.43%, precision of 96.47%, recall of 96.43%, F1-score of 96.44%, and BA of 96.38%. Compared to the base learners, it can be seen that the stacking model achieves better performance for all classification tasks. It is worth noting that the best results achieved by the base learners are due to careful data preprocessing, feature engineering, and hyperparameter optimization steps. The results of the sMCI vs. pMCI and [CN/sMCI] vs. [pMCI/AD] binary classification tasks assert that pMCI cases are challenging to identify. However, the proposed stacking ensemble model has the highest performance in all 3-class and 2-class classification tasks.

| Table 7: The 3-class and 2-class results. |
|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|-----------------------------|
| Model                   | No. of classes | Performance (%) | Accuracy (%) | Precision | Recall | F1-score | BA (%) |
| [CSs, NSB] model          | CN vs. AD          | 96.56          | 96.56         | 96.56      | 98.56       | 98.51      |
|                          | CN vs. pMCI        | 97.62          | 97.62         | 97.62      | 97.60       | 96.03      |
|                          | CN vs. sMCI        | 96.11          | 96.13         | 96.11      | 98.11       | 94.03      |
|                          | AD vs. sMCI        | 96.11          | 96.12         | 96.12      | 96.10       | 95.80      |
|                          | AD vs. sMCI        | 95.80          | 95.76         | 95.80      | 93.73       | 91.47      |
|                          | sMCI vs. pMCI      | 82.55          | 82.26         | 82.55      | 82.39       | 74.33      |
|                          | [CNsMCI] vs. [pMCI/AD] | 89.26          | 89.62         | 89.26      | 89.36       | 89.17      |
|                          | CN vs. MCI vs. AD  | 95.76          | 95.81         | 95.76      | 95.67       | 94.68      |
| [CSs, Static] model       | CN vs. AD          | 96.56          | 96.56         | 96.56      | 96.56       | 99.51      |
|                          | CN vs. pMCI        | 97.62          | 97.62         | 97.62      | 97.60       | 96.03      |
|                          | CN vs. sMCI        | 94.45          | 94.63         | 94.45      | 94.44       | 94.23      |
|                          | AD vs. sMCI        | 93.85          | 93.86         | 93.85      | 93.86       | 91.73      |
|                          | AD vs. pMCI        | 83.70          | 84.16         | 83.70      | 83.88       | 81.45      |
|                          | sMCI vs. pMCI      | 81.83          | 81.86         | 81.83      | 81.84       | 74.28      |
|                          | [CNsMCI] vs. [pMCI/AD] | 89.47          | 89.49         | 89.47      | 89.48       | 86.53      |
|                          | CN vs. MCI vs. AD  | 95.67          | 95.66         | 95.67      | 95.65       | 93.66      |
| [NSB, Static] model       | CN vs. AD          | 96.67          | 96.67         | 96.67      | 96.67       | 98.61      |
|                          | CN vs. pMCI        | 96.45          | 96.54         | 96.45      | 96.68       | 96.64      |
|                          | CN vs. sMCI        | 92.85          | 93.11         | 92.85      | 92.82       | 92.58      |
|                          | AD vs. sMCI        | 90.03          | 89.91         | 90.05      | 89.91       | 86.67      |
|                          | AD vs. pMCI        | 89.91          | 89.77         | 89.91      | 89.77       | 86.57      |
|                          | sMCI vs. pMCI      | 82.55          | 82.05         | 82.55      | 82.26       | 74.33      |
|                          | [CNsMCI] vs. [pMCI/AD] | 88.99          | 89.26         | 88.99      | 89.07       | 88.70      |
|                          | CN vs. MCI vs. AD  | 94.97          | 90.16         | 94.97      | 90.02       | 90.23      |
| Stacking model            | CN vs. AD          | 99.56          | 99.56         | 99.56      | 99.56       | 99.51      |
3.2.5 Comparison with the state-of-the-art from previous literature

This section presents a comparison between the best proposed competitive ensemble model in previous literature for AD progression detection, especially ensemble models. Please note that we are considering accuracy as the common metric that is given in all related works for the sake of fairness. Unfortunately, some authors do not publish the data they have used, and it is not possible to reproduce their experiments here. We just trust what was published by other authors to make an illustrative comparison. The proposed model is based on the stacking technique, which is like a voting ensemble except that the decisions of the base classifiers are used by another meta classifier to determine the weight of each learner. We utilize the early fusion of data by testing all possible combinations of the four modalities and selecting the best results. Level-0 in the proposed stacking ensemble has three XGB models as base classifiers with inputs of [CSs, NSB], [CSs, Static], and [NSB, Static]. Then, the decisions of these models are fused and trained using an LR algorithm as the meta classifier for Level-1.

Table 8: A comparison with other ensemble studies in the literature.

<table>
<thead>
<tr>
<th>Reference</th>
<th>Population</th>
<th>Modalities</th>
<th>Classes</th>
<th>Time series</th>
<th>ML method</th>
<th>Accuracy</th>
<th>Task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed ensemble</td>
<td>ADNI: 1371 subjects</td>
<td>MRI, EEG, FDG PET, CSF, age, sex, MMSE</td>
<td>4</td>
<td>YES</td>
<td>Stacking with XGB</td>
<td>89.15%</td>
<td>Progression detection</td>
</tr>
<tr>
<td>Farhan et al. [42]</td>
<td>ADNI: 300 subjects</td>
<td>MRI</td>
<td>2</td>
<td>NO</td>
<td>Voting with three neural networks</td>
<td>85.27%</td>
<td>CN AD diagnosis</td>
</tr>
<tr>
<td>Marques [104]</td>
<td>ADNI: 150 subjects</td>
<td>MRI</td>
<td>2</td>
<td>NO</td>
<td>Voting with SVM</td>
<td>83.50%</td>
<td>Progression detection</td>
</tr>
<tr>
<td>Pinho et al. [102]</td>
<td>ADNI: 1737 subjects</td>
<td>MRI</td>
<td>3</td>
<td>YES</td>
<td>Voting with SVM + Gaussian NB + SVM</td>
<td>87.00%</td>
<td>Progression detection</td>
</tr>
<tr>
<td>Liu et al. [98]</td>
<td>ADNI: 116 subjects</td>
<td>MRI</td>
<td>2</td>
<td>NO</td>
<td>Voting with evolutionary weighted SVMs</td>
<td>90.5%</td>
<td>Progression detection</td>
</tr>
<tr>
<td>Nanni et al. [94]</td>
<td>ADNI: 400 subjects</td>
<td>MRI</td>
<td>4</td>
<td>NO</td>
<td>Voting with three 3D Densely Connected CNNs</td>
<td>83.31%</td>
<td>Progression detection</td>
</tr>
<tr>
<td>An et al. [89]</td>
<td>NACC: 23,165 subjects</td>
<td>MRI, EEG, CSF, age, sex, MMSE</td>
<td>2</td>
<td>NO</td>
<td>Stacking with neural network as meta classifier and sparse autoencoders as base classifiers</td>
<td>87%</td>
<td>AD diagnosis</td>
</tr>
<tr>
<td>Pan et al. [83]</td>
<td>ADNI: 509 subjects</td>
<td>MRI</td>
<td>2</td>
<td>NO</td>
<td>Multisite voting with CNN (3D-3DNet) base classifiers</td>
<td>ADNI: 83.4%</td>
<td>Progression detection</td>
</tr>
<tr>
<td>Chou et al. [88]</td>
<td>ADNI: 315 subjects</td>
<td>MRI</td>
<td>3</td>
<td>NO</td>
<td>Voting with CNN (VGG-16, GoogLeNet, and AlexNet)</td>
<td>ADNI: 93.4%</td>
<td>Progression detection</td>
</tr>
<tr>
<td>SYED et al. [89]</td>
<td>Figshare: 333 subjects</td>
<td>MRI, CSF protein biomarkers, age, gender, amyloid proteins, tau, ptau, Aβ</td>
<td>2</td>
<td>NO</td>
<td>Weighted average with two base learners of LIR and Linear SVM</td>
<td>95.52%</td>
<td>CN/MCI diagnosis</td>
</tr>
<tr>
<td>Yao et al. [15]</td>
<td>ADNI: 400 subjects</td>
<td>MRI</td>
<td>2</td>
<td>NO</td>
<td>Hierarchical ensemble of 2 binary classifiers by XGB and SVM</td>
<td>94.38%</td>
<td>Progression detection</td>
</tr>
<tr>
<td>Nanni et al. [24]</td>
<td>ADNI: 400 subjects</td>
<td>MRI</td>
<td>4</td>
<td>NO</td>
<td>Voting with sum rule of 8 classifiers</td>
<td>52.92%</td>
<td>Progression detection</td>
</tr>
<tr>
<td>Barnett et al. [44]</td>
<td>ADNI: 400 subjects</td>
<td>MRI</td>
<td>4</td>
<td>NO</td>
<td>Bagging with SVM</td>
<td>90.10%</td>
<td>Progression detection</td>
</tr>
<tr>
<td>Desikan and Laroia [10]</td>
<td>ADNI: 400 subjects</td>
<td>MRI</td>
<td>4</td>
<td>NO</td>
<td>Voting with RF</td>
<td>61.96%</td>
<td>Progression detection</td>
</tr>
<tr>
<td>Chen et al. [90]</td>
<td>NACC: 105 samples</td>
<td>MRI, MMSE, logical memory (LM)</td>
<td>2</td>
<td>NO</td>
<td>Voting with SVM + SVM + MMSE + LM</td>
<td>90.90%</td>
<td>MC/UCN diagnosis</td>
</tr>
<tr>
<td>Rannsi et al. [32]</td>
<td>ADNI: 200 subjects</td>
<td>MRI</td>
<td>4</td>
<td>NO</td>
<td>Bagging with random forest one vs two classifiers</td>
<td>85.22%</td>
<td>Progression detection</td>
</tr>
<tr>
<td>Jin and Dong [91]</td>
<td>ADNI: 400 subjects</td>
<td>MRI</td>
<td>4</td>
<td>NO</td>
<td>Gradient boosting tree</td>
<td>85.22%</td>
<td>Progression detection</td>
</tr>
<tr>
<td>Zhou et al. [45]</td>
<td>Diffusion Tensor Imaging (DTI)</td>
<td>MRI, age, gender, BMX, CSF</td>
<td>2</td>
<td>YES</td>
<td>Voting with SVM + SVM + BMX + CSF + DTI</td>
<td>ADNI: 85.30%, ADNI: 85.30%</td>
<td>ADNI: 85.30%</td>
</tr>
<tr>
<td>Moon et al. [92]</td>
<td>ADNI: 1737 subjects</td>
<td>MRI</td>
<td>3</td>
<td>YES</td>
<td>Random forest</td>
<td>73.00%</td>
<td>ADNI: 85.30%</td>
</tr>
<tr>
<td>Siddiqui et al. [101]</td>
<td>ADNI: 41</td>
<td>MRI</td>
<td>2</td>
<td>NO</td>
<td>Random linear oracle with SVM</td>
<td>95.00%</td>
<td>ADNI: 85.30%</td>
</tr>
<tr>
<td>Nanni et al. [94]</td>
<td>ADNI: 509 subjects</td>
<td>MRI</td>
<td>2</td>
<td>NO</td>
<td>Voting with weight sum rule and SVM</td>
<td>95.00%</td>
<td>ADNI: 85.30%</td>
</tr>
<tr>
<td>Srirajpara et al. [51]</td>
<td>ADNI: 300 subjects</td>
<td>MRI</td>
<td>2</td>
<td>NO</td>
<td>Voting with C4.5 for feature selection</td>
<td>97.60%</td>
<td>CN/AD diagnosis</td>
</tr>
<tr>
<td>Farhan et al. [42]</td>
<td>ADNI: 300 subjects</td>
<td>MRI</td>
<td>2</td>
<td>NO</td>
<td>Voting with SVM, RT, and MLP</td>
<td>93.75%</td>
<td>ADNI: 85.30%</td>
</tr>
<tr>
<td>Grey et al. [90]</td>
<td>ADNI: 147 subjects</td>
<td>MRI</td>
<td>2</td>
<td>NO</td>
<td>Adaptive boosting with random forest one vs two classifiers</td>
<td>95.90%</td>
<td>ADNI: 85.30%</td>
</tr>
<tr>
<td>Simpson et al. [97]</td>
<td>ADNI: 311</td>
<td>MRI</td>
<td>2</td>
<td>NO</td>
<td>Bagging with SVM</td>
<td>95.90%</td>
<td>ADNI: 85.30%</td>
</tr>
<tr>
<td>Lin et al. [98]</td>
<td>ADNI: 653 subjects</td>
<td>MRI</td>
<td>2</td>
<td>NO</td>
<td>Voting with SVM + SVM + MLP</td>
<td>90.80%</td>
<td>ADNI: 85.30%</td>
</tr>
<tr>
<td>Chen et al. [99]</td>
<td>ADNI: 26</td>
<td>MRI</td>
<td>2</td>
<td>NO</td>
<td>Boosting with Bayesian-network representation</td>
<td>81.00%</td>
<td>ADNI: 85.30%</td>
</tr>
<tr>
<td>Polak et al. [100]</td>
<td>ADNI: 300 subjects</td>
<td>EEG, MRI, PET</td>
<td>2</td>
<td>NO</td>
<td>Voting with SVM + SVM + MLP</td>
<td>85.50%</td>
<td>ADNI: 85.30%</td>
</tr>
<tr>
<td>Hinton and Kingma [101]</td>
<td>ADNI: 308</td>
<td>EEG, MRI, PET</td>
<td>2</td>
<td>NO</td>
<td>Boosting with Convolutional Neural Networks</td>
<td>81.00%</td>
<td>ADNI: 85.30%</td>
</tr>
</tbody>
</table>
Furthermore, the proposed stacking model achieved the best performance because it is based on a carefully selected list of features extracted from different modalities of time series data. Then, our proposed model integrates the capabilities of voting and boosting techniques using XGB as the base learner and stacking as the decision fuser.

As previously noted, this model achieved optimal performance without considering MRI neuroimaging features. This means that the model is expected to be less expensive because it is based on a less expensive list of features. As a result, the model is applicable in a wider variety of medical environments and is especially suitable for use in developing countries. Table 8 shows a comparison of different AD ensemble models. It can be seen that the majority of the studies concentrated on MRI data analysis, yet they achieved worse results than our model. The most relevant properties of our approach are: (1) The size of our dataset is significantly larger than most studies; (2) we have considered the multimodality and time-series nature of the data; (3) we carefully selected the most discriminating features from the used modalities; (4) different ensemble techniques have been tested, and the stacking ensemble model with XGB achieved the best performance; (5) the selected model achieved a balance between accuracy and diversity among the selected base learners; and, finally, (6) the best results were achieved without considering MRI neuroimaging features. This model achieved optimal performance without considering MRI neuroimaging features. This is mainly because the number of base classifiers is few. As a result, an exhaustive search was considered a feasible solution for classifier selection. In a future study, we will consider the fusion of other modalities (e.g., medications, medical history (MH), Hachinski ischemic score (HIS), cerebrovascular disease (CVD), Unified Parkinson's Disease Rating Scale (UPDRS), Neuropsychiatric Inventory Questionnaire (NPIQ), Geriatric Depression Scale (GDS), Functional Activities Questionnaire (FAQ)).

4. Conclusion

This paper proposed a pipeline for building an ensemble for AD progression detection based on multimodal time series data. The advantages of the proposed pipeline are two-fold. First, it provides detailed steps to extract the most discriminating and medically critical features from static and time-series data. Second, it considers the trade-off between accuracy and diversity, which guides the selection of the optimum base classifiers. Based on the selected features from time series and static modalities, we created and compared many ensemble models based on either early data fusion and late decision fusion strategies. We conducted experiments using the ADNI dataset to tackle a 4-class classification problem based on 3-time series modalities and 1-static modality as inputs. The best-performing model...
is based on the stacking scheme with three XGB models at level-0 and the LR classifier as a meta classifier at level-1. The XGB models were fitted with three datasets, [CSs, NSB], [CSs, Static], and [NSB, Static], and the LR model is based on the class probabilities of the first-level classifiers. The experimental results showed that the proposed approach yields superior progression performance compared to all previous ensemble approaches in the literature. In summary, the proposed ensemble evaluates AD progression based on a cheap, easily collectible, simple set of features allowing the prediction of a patient’s condition for 2.5 years in the future. In our future work, we will extend this work to consider other modalities, including the patient’s comorbidities and drugs taken during the observation period. In addition, we will investigate the interpretability of the proposed models to improve the medical expert’s decision-making process.

**Funding:** Jose M. Alonso is Ramon y Cajal Researcher (Ryc-2016-19802). This research is funded by the Spanish Ministry of Science, Innovation, and Universities (grants RTI2018-099646-B-I00 and RED2018-102641-T) and the Galician Ministry of Education, University and Professional Training (grants ED431F2018/02, ED431C2018/29, ED431G/08, and ED431G2019/04). Some of the previous grants were co-funded by the European Regional Development Fund (ERDF/FEDER program). This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ICT Creative Consilience Program (IITP-2021-2020-0-01821) supervised by the IITP (Institute for Information & communications Technology Planning & Evaluation), and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2021R1A2C1011198).

**Conflicts of interest:** No conflicts of interest, financial or otherwise, are declared by the authors.

**Availability of data and material:** The data are publicly available from the ADNI data repository (http://adni.loni.usc.edu/). In addition, the specific patient RIDs used in our study and the full description of used features can be found in the Supplementary Files.

**CRediT author statement:** Shaker El-Sappagh and Farman Ali: Conceptualization, Methodology, Software, Validation, Investigation, Writing - original draft. Tamer ABUHMED: Conceptualization, Data curation, Validation, Writing - review & editing. Jaiteg Singh: Data curation, Validation, Writing - review & editing. Jose M. Alonso: Conceptualization, Methodology, Writing - review & editing, Supervision, Project administration.

**Ethics approval:** Data used in this study were obtained from the ADNI (http://adni.loni.usc.edu/). The Alzheimer’s Disease Neuroimaging Initiative Data and Publications Committee (ADNI DPC) coordinates patient enrollment and ensures standard practice on the uses and distribution of the data as follows: The ADNI data were previously collected across 50 research sites. To participate in the study, each study subject gave written informed consent at the time of enrollment for imaging and genetic sample collection and completed questionnaires approved by each participating sites’ Institutional Review Board (IRB). All procedures performed in studies involving human participants were in accordance with the ethical standards of the institutional and/or national research committee and with the 1964 Helsinki declaration and its later amendments or comparable ethical standards. A complete description of ADNI and up-to-date information is available at http://adni.loni.usc.edu/ and data access requests are to be sent to http://adni.loni.usc.edu/data-samples/access-data/. Detailed inclusion criteria for the diagnostic categories can be found at the ADNI website (http://adni.loni.usc.edu/methods). The ethics committees/institutional review board that approved the ADNI study are listed within Supplementary file (part 4).

**References**


S. Iddi, D. Li, P. S. Aisen, M. S. Rafii, W. K. Thompson, and M. C. Donohue, “Predicting the course of Alzheimer’s progression,” *Brain Informatics*, vol. 6, no. 1, 2019.


