Contents lists available at ScienceDirect

Applied Soft Computing

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

Highlights

Enhancing dementia prediction models: Leveraging temporal patterns and class-balancing methods

Applied Soft Computing xxx (xxxx) xxx

Flavio Luiz Seixas*, Elaine Rangel Seixas, Alex A. Freitas

- We propose two novel types of constructed temporal features derived from longitudinal data for predicting dementia based on monotonicity patterns and decision trees.
- We evaluate the effect of many combinations of the proposed constructed features and various class-balancing methods on the predictive performance of dementia models learned by a machine-learning algorithm.
- We report the most important features in the best predictive model learned in the experiments, i.e. the features which most contribute to predicting whether a patient has dementia.

Contents lists available at ScienceDirect

Applied Soft Computing

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

Enhancing dementia prediction models: Leveraging temporal patterns and class-balancing methods

Flavio Luiz Seixas [a],*, Elaine Rangel Seixas [a], Alex A. Freitas [b]

[a] Institute of Computing, Fluminense Federal University, Av. Gal. Milton Tavares de Souza, s/n, Niteroi, 24210-346, Rio de Janeiro, Brazil
[b] School of Computing, University of Kent, Canterbury, CT2 7FS, Kent, United Kingdom

ARTICLE INFO

Keywords:
Machine learning
Longitudinal data
Feature construction
Dementia prediction modelling

ABSTRACT

Predicting dementia with machine learning (classification) models learned from longitudinal data remains challenging. This paper introduces an innovative approach for learning predictive dementia models that leverage temporal patterns derived from longitudinal data. Specifically, we propose two types of automatically constructed temporal features based on monotonicity patterns of features' values and decision tree-based patterns. The constructed temporal features were added to the original dataset to improve the predictive performance of well-known classifiers, XGBoost and Random Forest. We also investigated using several types of class-balancing methods to cope with the large degree of imbalanced classes in our dataset. We assessed the impact of the constructed temporal features and different types of class-balancing methods (and their combinations) on improving classifiers' predictive performance on a dementia dataset derived from the English Longitudinal Study of Ageing. We also report the most important predictive features in the best dementia prediction models learned in our experiments.

1. Introduction

Population ageing is a worldwide phenomenon that profoundly impacts society, health, and economic systems [1]. One of the main concerns associated with population ageing is the increasing prevalence of dementia [2]. According to estimates, the number of individuals with dementia worldwide is projected to increase from 57.4 (95% uncertainty interval 50.4–65.1) million cases in 2019 to reach approximately 152.8 (130.8–175.9) million cases in 2050 [3]. Before the COVID-19 pandemic, Alzheimer's disease (AD) had already occupied the sixth leading cause of death in the United States. From 2000 to 2019, the recorded deaths attributed to Alzheimer's disease more than doubled and experienced a 145% increase. In contrast, deaths due to heart disease decreased by 7.3% over the same time frame. Moreover, the care for older adults who have Alzheimer's disease falls mainly on unpaid caregivers, who contribute approximately 80% of the support required by this population [4]. Despite its high mortality rate, up to 40% of dementia cases could potentially be mitigated or prevented by addressing 12 significant risk factors. These factors include, for example, low levels of education, smoking, hypertension, obesity, diabetes, and excessive alcohol consumption [5].

Dementia is a mental disorder characterized by progressive deterioration of cognitive functions, and it encompasses many symptoms, typically including memory impairments, language difficulties, challenges in problem-solving, and various other cognitive issues, each linked to distinct alterations in the brain [6,7]. Among the common types of dementia, such as Lewy Body dementia, cerebrovascular dementia and frontotemporal dementia, Alzheimer's disease is the most frequent type. Alzheimer's disease is characterized by specific brain changes, notably the accumulation of abnormal proteins such as beta-amyloid and phosphorylated tau, affecting neuronal functions and leading to progressive brain function loss [7–9].

The development of artificial intelligence and machine learning algorithms has contributed to supporting healthcare areas in the screening, diagnosing and managing cognitive disorders related to dysfunctional brains. Given the rising prevalence of dementia, there is an increasing demand for effective strategies to detect and manage Alzheimer's disease in its early stages [10–13].

This paper proposes and evaluates a machine learning pipeline for learning dementia prediction models, encompassing data acquisition, data preprocessing, feature engineering (constructing new features from the original features), class-balancing methods (to cope with the large degree of class imbalance in our dataset), model development, and evaluation. More specifically, this paper proposes an innovative machine learning approach for learning predictive dementia models that leverage two types of machine learning methods for improving

* Corresponding author.
E-mail addresses: fseixas@ic.uff.br (F.L. Seixas), elaine_rangel@id.uff.br (E.R. Seixas), A.A.Freitas@kent.ac.uk (A.A. Freitas).
URL: https://www.ic.uff.br/~fseixas (F.L. Seixas).

https://doi.org/10.1016/j.asoc.2025.112754
Received 2 June 2024; Received in revised form 5 November 2024; Accepted 6 January 2025

the predictive accuracy of classification models learned from the data: the automatic construction of new types of predictive features, exploring temporal patterns in the data; and the use of several types of class-balancing methods. The proposed machine learning approach is evaluated by using well-known classifiers to learn predictive dementia models from a dataset derived from the English Longitudinal Study of Ageing (ELSA-UK) [14,15]. We used the XGBoost [16] and Random Forest [17] algorithms as classifiers.

The ELSA-UK is a longitudinal study initiated in 2002 with participants aged 50 and above. It has conducted biennial interviews and collected comprehensive data covering diverse aspects of participants' lives, including well-being, relationships, housing, cognitive functions, and social participation. We selected data that probably correlated with dementia, such as participants' demographic, health, and cognition data. In addition, the longitudinal nature of the data – i.e., the data contains repeated measures of variables across time (e.g. cholesterol levels for an individual in several different time points) – was exploited during the above mentioned feature construction process, as described in detail later.

The remainder of this paper is organized into six more sections. Section 2 reports a brief literature review, focusing mainly on using machine learning to predict dementia to put this work into a broader context. Section 3 describes in detail the proposed new types of constructed temporal features designed to leverage the temporal information in the data to improve predictive performance. Section 4 describes the class-balancing methods used in this work to cope with the large degree of imbalanced classes in our dementia dataset. Section 5 describes the components of our machine-learning pipeline. Section 6 presents and discusses the results of our computational experiments. Section 7 presents the conclusions and suggestions for future work.

2. Literature review

Machine Learning (ML) techniques have been widely applied to the prediction of dementia [7,18–20]. Studies predicting dementia aim to further our comprehension of the progression of Alzheimer's disease (AD) to the extent that we can accurately forecast the stages from the onset of the disease to significant outcomes in individual AD patients. [21,22].

Unlike cross-sectional data, which captures a snapshot of an individual's cognitive state at a specific time, longitudinal data tracks changes over an extended period. This temporal dimension provides a dynamic perspective, enabling researchers and clinicians to observe the trajectory and evolution of cognitive decline and its predictive features. Numerous studies applied machine learning or statistical methods to longitudinal data to develop predictive models for identifying dementia trajectories and their main predictors among the elderly [18,22–26]. Some relevant examples are as follows.

Stamate et al. [27] utilized survival analysis with machine learning algorithms to predict dementia onset. They used data from the ELSA database, identifying dementia cases through participants' self-reported physician diagnoses. In cases where the participant was unable to respond to the interview, the Informant Questionnaire on Cognitive Decline in the Elderly (IQCODE) results were employed, with a score above 3.386 indicating the presence of dementia [28,29]. Their approach involved Cox Penalised Regression using Elastic Net and Survival Random Forest techniques [30,31]. Their results outperformed the traditional Cox regression approach.

Kim et al. [32] focused on analysing feature combinations in a longitudinal model to predict dementia. Various features were analysed and optimized to improve the predictive performance of a dementia model learned by a support vector machine algorithm. Their results demonstrated that the personal disease history of a patient has a crucial role in dementia prediction.

Huang et al. [33] identified disease patterns and symptom clusters and their sequences before incident dementia in a longitudinal cohort study. They used multivariable LASSO regression, feature selection, Support Vector Machine, or Random Forest to construct predictive models to predict subsequent dementia. Their analysis indicated that the most common conditions leading to dementia were cerebrovascular disease and hypertension.

Other studies compared the performance of different prediction models for detecting dementia using ELSA [25,27]. For example, Yang and Bath [18] used attributes from different domains that were directly selected from the items in the ELSA questionnaire and grouped into six categories based on their characteristics: (a) demographic and economic factors, (b) social engagement and social network factors, (c) physical health and disability factors, (d) psychological and mental health factors, (e) lifestyle factors, (f) cognition factors. The developed predictive models ranged from ensemble models (Regularized Greedy Forests and Random Survival Forests) and neural networks (Convolutional Neural Networks) to conventional, simpler models (Logistic Regression). However, class imbalance posed a challenge: Positive dementia cases accounted for less than 2% of the total cases. They used resampling techniques (SMOTE, ADASYN) to mitigate this and assigned different weights to minority and majority classes.

The process of developing predictive models includes many steps. Most works focus on modelling algorithms but neglect other critical aspects of the modelling process, such as finding the best subset of predictors for improving model performance [34,35] – although identifying the optimal feature combination in longitudinal models for predicting dementia has been explored in some prior studies [32,33].

Note that although some machine learning or statistical methods are capable of directly handling longitudinal data, such as Long Short-Term Memory (LSTM) Networks [36,37], Generalized Linear Mixed Models (GLMM) [38], and Generalized Estimating Equations (GEE); most machine learning methods cannot directly cope with longitudinal data [39,40]. More precisely, most supervised machine learning (classification or regression) methods cope with longitudinal data by using a data transformation approach where the time indices of different measurements of the same feature are removed. As a result, the measurements of the same feature across time are treated as different (unrelated) features so that a standard, non-longitudinal classification algorithm can be applied [40]. This common approach ignores the temporal relationship between different feature values, a clear limitation.

In this context, to mitigate that limitation, a promising approach to allow standard classification algorithms to access the temporal information in longitudinal datasets consists of automatically constructing temporal features that exploit that type of information and adding such Constructed Temporal Features (CTFs) to the data, so that they are directly available to standard classification algorithms. This approach was used in [41], where simple CTFs involving the difference between feature values across time were proposed, and in [42], where CTFs based on the concept of monotonicity of a feature's values across time were proposed.

This current work extends the research of Ribeiro and Freitas [42] by proposing two new types of CTFs: (a) a more sophisticated and flexible type of monotonicity pattern-based CTF and (b) a new type of decision tree path-based CTF. Section 3 describes these new CTF types. As an additional contribution, this work also extensively investigates the performance of 10 different variants of class-balancing methods (described in Section 4) and how they perform in combination with the above two new types of CTFs.

3. Constructing features from longitudinal data

The dataset created in this work contains data extracted from the ELSA (English Longitudinal Study of Ageing) database. The ELSA study started in 2002 (wave 1), with its participants initially recruited from an annual cross-sectional survey of households. Subsequently, these

participants were subject to follow-up every two years. Each wave (1–9) of ELSA received ethical approval from the National Research Ethics Service, specifically the London Multicenter Research Ethics Committee, and all participants provided informed consent.

The ELSA database contains longitudinal data, i.e. it encompasses measurements of specific features taken at various time points, tracking the same group of individuals. Longitudinal data analysis plays a crucial role in the context of dementia prediction. Hence, we analyse a longitudinal dataset extracted from the ELSA database in this work. This dataset was constructed for the classification task of machine learning, where the class variable indicates whether or not an individual is diagnosed with dementia, and the predictive features contain some basic demographic information about individuals (in particular, gender and age) and many features related to the health of individuals. Notably, our database originates from the Harmonized English Longitudinal Study of Aging (ELSA) initiative, a subset of the ELSA interviews aimed at enhancing its usability. The Harmonized ELSA is part of a larger project supported by the National Institute on Aging, involving various countries such as the United States, Mexico, Israel, South Korea, China, and India [43]. Following the conventions set forth by RAND HRS and Harmonized HRS [44], the Harmonized ELSA maintains standardized variable naming and data structures. Our analysis focuses on data and information derived from the Harmonized ELSA dataset and its codebook version G, published in July 2023 and developed by the Gateway to Global Aging Data [45]. Specifically, our dataset encompasses features related to demographic and health domains with participants from the UK.

To leverage the dataset's longitudinal information, we propose Constructed Longitudinal Features (CTFs) that represent patterns in the change of the values of a longitudinal feature across time for each individual. Hence, CTFs can potentially improve predictive accuracy, which will be evaluated later in this paper. More precisely, we propose two new types of CTFs: one based on monotonicity patterns of a feature's values across time and the other based on decision trees, as discussed in detail in the following two subsections.

3.1. Constructing monotonicity score features

Monotonicity is a fundamental concept in data analysis and statistics and refers to a specific pattern or trend in data where values consistently increase or decrease over a sequence or period [46]. Longitudinal data is particularly suitable for learning predictive models that capture monotonicity patterns in the data since, in such studies, variables tend to have repeated measurements of their values across time.

Furthermore, clinical symptoms of dementia and Alzheimer's disease are characterized by progressive amnesia, followed by a gradual decline in all cognitive functions faster than usual with ageing [47]. For dementia cases, it is expected that, for some clinical features, a monotonic increase or decrease in the feature's value may be a good predictor of the occurrence of dementia. Hence, evaluating the monotonic pattern of a feature's values can have a critical role in the timely detection of dementia [48].

Hence, one of the new types of Constructed Temporal Feature (CTF) proposed in this work focuses on exploiting monotonicity patterns in the data. To detect such patterns, for each longitudinal feature (i.e. a feature having repeated measurements across time) available in the original data, we create a new feature called monotonicity score, which summarizes the monotonic pattern (if any) observed in the values of a feature across time.

In essence, a monotonicity score feature takes an integer value representing the number of consecutive pairs of time points where the value of a feature has monotonically increased (positive score) or decreased (negative score) across time. This basic idea is an extension of the more straightforward, coarser-grained type of monotonicity feature proposed in [41,42], as follows. In those two works, a monotonicity feature can take only three values, 1 or -1, to indicate that a feature's values monotonically increase or decrease (respectively) across *all* consecutive pairs of time points; or the value 0 otherwise. This is shown in Example 1.

Example 1. Consider a longitudinal feature f measured across 5-time points, denoted $t_1 \ldots t_5$. Suppose f's value increased from t_1 to t_2, but then f's value monotonically decreased across all pairs of consecutive steps from t_2 to t_5, as shown in Fig. 1(B). Then, despite a trend of monotonically decreasing values from t_2 to t_5, the monotonicity score would be 0 because f's value increased from t_1 to t_2.

Example 1 shows that the monotonicity score proposed in [41,42] is not flexible enough to capture the partial monotonic pattern that holds across nearly all consecutive pairs of time points; i.e., that type of monotonicity score captures only an "all-or-nothing" monotonicity pattern.

The new type of monotonicity score feature proposed in this work was designed to be more flexible, representing both complete and partial degrees of monotonic patterns in the values of a longitudinal feature.

In addition, the proposed monotonicity score was designed to capture monotonicity constraints involving the most recent time points rather than the oldest time points since more recent feature values tend to be more relevant for disease diagnosis.

To be precise, to compute the monotonicity score of a longitudinal feature f for each instance, the system tries to identify a "substantially monotonic" trend in the values of f across time. To define such a trend, first of all, let S denote an uninterrupted sequence of $k+1$ consecutive time points, denoted $S = (t_i, t_{i+1}, \ldots, t_{i+k})$, where k is the number of pairs of adjacent times points in S ($k >= 2$). A "substantially monotonic" trend is defined as an uninterrupted sequence (S) of $k+1$ consecutive time points where all k pairs of adjacent time points t_j and t_{j+1} (with j varying in the range $(i, i+1, \ldots, i+k-1)$) exhibit the same type of "substantial change" in f's value between t_j and t_{j+1}. The two possible types of "substantial changes", substantial increase and substantial decrease, are defined as follows. Let $f(t_j)$ and $f(t_{j+1})$ denote the f values at two adjacent time points t_j and t_{j+1}. Then, f's value exhibits a substantial increase if $f(t_{j+1}) \geq 1.1 \cdot f(t_j)$; and f's value exhibits a substantial decrease if $f(t_{j+1}) \leq 0.9 \cdot f(t_j)$; i.e., deeming an increase or decrease of more than 10% in f's value as a substantial change.

Hence, a sequence S of f's values exhibits a substantially monotonically increasing trend if, for all pairs of adjacent time points t_j and t_{j+1} in S (i.e., for j in $(i, i+1, \ldots, i+k-1)$), $f(t_{j+1}) \geq 1.1 \cdot f(t_j)$; and S exhibits a substantially monotonically decreasing trend if, for all pairs of adjacent time points t_j and t_{j+1} in S, $f(t_{j+1}) \leq 0.9 \cdot f(t_j)$.

In the above definition, the condition $k \geq 2$ is a constraint, meaning that the sequence S must include at least two adjacent pairs of consecutive time points (i.e. three successive time points). This is a minimum baseline for a monotonic trend since a sequence with just one pair of consecutive time points would have just one (potentially substantial) increase or decrease in f's value, which would not be a "trend". In addition, the above 10% threshold is a parameter specified to focus on detecting monotonic feature value changes that are more likely to be relevant for disease prediction, as opposed to minor increases or decreases in a feature value. In this work, we have not attempted to optimize this threshold, using 10% as an intuitively reasonable value, so the optimization or tuning of this threshold is left for future research.

In addition, detecting a monotonic trend also has to satisfy a second constraint: the sequence S must include the last pair of consecutive time points (i.e. the trend must end at the last time point of the dataset).

Suppose the above two constraints are satisfied for the values of a given longitudinal feature f in a given instance. In that case, the monotonicity score of f for that instance is calculated based on the value of k, i.e., the count of consecutive pairs of time points in S satisfying the above definition of a "substantially monotonic" trend, as follows: if S

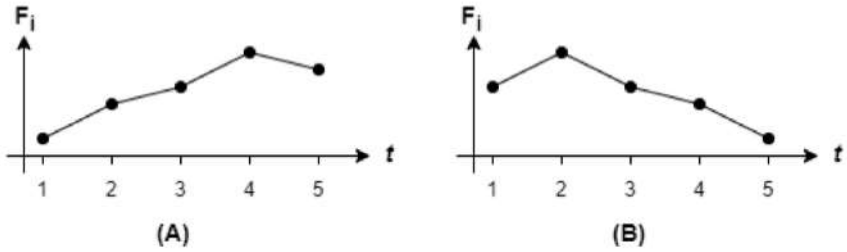

Fig. 1. Examples used to clarify the calculation of the proposed monotonicity score.

Table 1
Examples of monotonicity score calculations for the longitudinal feature "self-report of health" (*shlt*), spanning waves 4 through 9 (denoted by the prefix r4–r9 in the feature's acronym in the ELSA database).

r4shlt	r5shlt	r6shlt	r7shlt	r8shlt	r9shlt	monotonicity score
4	4	4	3	2	1	−2
2	1	1	2	2	4	0
4	4	3	3	4	5	1

exhibits a monotonic *increase* in f's values across the $k+1$ time points in S, the monotonicity score of f is $k-1$; and if S exhibits a monotonic *decrease* in f's values across those time points, the monotonicity score of f is $-(k-1)$. In both cases, the value 1 is subtracted from k because the above definition of substantially monotonic trend has the constraint that $k = 2$ (since a sequence with $k = 1$ would not be any "trend" as mentioned earlier), so the shortest possible trend satisfying the above definition would have the monotonicity score of 1 or -1. Finally, if any of the above two constraints are unsatisfied, the monotonicity score of f is set to 0.

As an example of calculating the proposed monotonicity score, let us revisit the above Example 1 (refer to Fig. 1(B)). As mentioned earlier, in that example, the monotonicity score proposed in [41,42] would be 0, indicating no monotonic trend, despite the occurrence of a monotonically decreasing trend from time point t_2 to t_5, just because this trend did not hold for the oldest pair of time points (from t_1 to t_2). By contrast, in that example, if the decreasing trend from t_2 to t_5 is substantial (as defined earlier), the monotonicity score proposed in this paper would be −2 (calculated as: -(3–1)), recognizing that there is substantial decreasing monotonic trend in f's values, even though the trend is not valid for all pairs of consecutive time points.

Let us now consider an example of the effect of the above second constraint (namely, that the monotonic trend has to finish at the last time point of the dataset) in calculating the proposed monotonicity score.

Example 2. Consider a longitudinal feature f measured across 5 time points, denoted $t_1 \ldots t_5$. Suppose f's value substantially monotonically increased from t_1 to t_4; but f's value decreased from t_4 to t_5, as shown in Fig. 1(A). In this case, despite the increasing trend from t_1 to t_4, the monotonicity score would be 0 because the monotonic trend did not include the last (most recent) pair of time points.

Table 1 illustrates the calculation of the monotonicity score for three instances in the dataset, for the longitudinal feature *shlt* (self-report of health), with feature values from waves 4 through 9 (denoted by the prefix r4–r9 in the feature's acronym in the ELSA database). The feature values highlighted in grey represent substantial increases or decreases used in the calculation of the monotonicity score.

In the first row of Table 1, the feature value monotonically decreased from wave 6 (r6) to wave 9 (r9). For each pair of consecutive waves in this period (i.e., wave pairs r6–r7, r7–r8, r8–r9), the decrease was substantial (more than 10%) – e.g., a decrease from 4 in wave r6 to 3 in wave r7, a decrease of 25%. There are 3 consecutive wave pairs with a substantially monotonic trend, so $k = 3$. This monotonic trend also satisfies the constraint of ending at the last (most recent) wave in the dataset, r9. Recall that the monotonicity score was defined as $-(k-1)$ for substantially monotonically *decreasing* sequences, so the score is −2 in this case.

In the second row of Table 1, the feature value substantially increased from 2 in wave r8 to 4 in wave r9, but it did not increase from wave r7 to wave r8. Hence, in this case the sequence of monotonically increasing feature values ending at wave r9 has just one pair of adjacent waves (r8–r9), less than the minimum of two required to have a monotonicity score different from 0. So, the monotonicity score is 0.

In the third row of Table 1, the feature value monotonically increased from wave r7 to wave r9. For each pair of consecutive waves in this period (i.e., r7–r8 and r8–r9), the increase was substantial (more than 10% increase) – e.g. an increase from 3 in wave r7 to 4 in wave r8, an increase of 33%. There are 2 consecutive wave pairs with a substantially monotonic trend, so $k = 2$. This monotonic trend also satisfies the constraint of ending at the last (most recent) wave in the dataset, r9. For substantially monotonically *increasing* trends, the monotonicity score was defined as $k-1$, so the score is 1 in this case.

3.2. Constructing decision tree path-based features

The second new type of constructed feature proposed in this article is defined as follows for each longitudinal feature f in the dataset. First, the system learns a decision tree from the training set but uses as input variables only the set of temporal variations of feature f (i.e., $f_{t1}, \ldots, f_{tk}$) – where k is the number of time points – and the class variable. Then, each path in that learned decision tree, from the root node to a leaf node, is transformed into a constructed binary feature, taking the true or false value to indicate whether or not an instance satisfies all the logical conditions (feature tests) along that path. The motivation for this is that each path tends to capture non-linear interactions among temporal variations of feature f. Each path has some relevant predictive power by design since the decision tree is learned to predict the class variable's value for an instance based on the values of the features in the decision tree's paths.

To learn such a decision tree for feature construction, we specify a constraint in the maximum tree depth size, set to 2 in this work. The motivations for this constraint are twofold. First, this limits the number of constructed features for each longitudinal feature to a relatively small number to avoid giving too many new constructed features to a classification algorithm later. Note that if only two branches are coming out from each internal tree node (as usual when features are numeric), a maximum tree depth of 2 will lead to at most 4 leaves, i.e. at most four constructed features (for each longitudinal feature in the dataset).

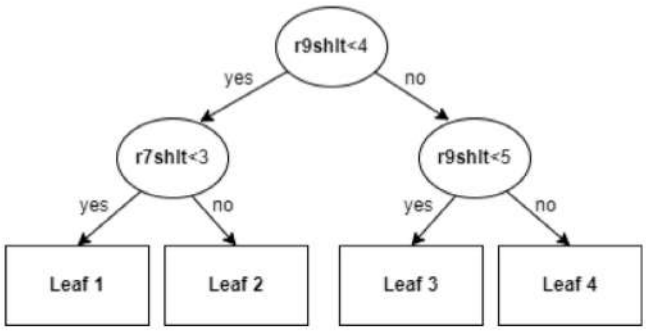

Fig. 2. Decision tree example of feature self-reported general health status.

Second, a maximum tree depth of two helps to mitigate the problem of data fragmentation in decision trees, i.e., the problem that the number of instances falling into each leaf node tends to decrease quickly as the tree depth increases, as a result of the recursive partitioning of the training data performed by the decision tree algorithm [49]. In summary, a maximum tree depth of 2 helps to ensure that not too many features are constructed and increases the chance that each newly constructed feature represents a robust predictive pattern instead of overfitting the training data.

In Fig. 2, we present a decision tree example based on the longitudinal feature **rωshlt**, a feature's name in our dataset where the prefix 'r' is followed by the wave number (denoted by ω) and a mnemonic related to that ELSA variable's meaning (this feature name notation comes from the ELSA database). The prefix 'r' is used when the respondent to the interview is the reference person. In the example decision tree in Fig. 2, the variable 'shlt' represents the respondent's self-reported general health status, rated on a scale from Excellent (code 1) to Poor (code 5), whilst $\omega = 7$ and $\omega = 9$ indicate the ELSA survey's wave number (time point IDs) 7 and 9, i.e. surveys conducted in 2015 and 2019, respectively (each pair of consecutive ELSA waves are two years apart). The four distinct root-to-leaf paths in Fig. 2 are transformed into four binary features, each taking a *true* or $false$ value depending on whether or not all the conditions in that path are satisfied by a given instance (individual). To be precise, the four conjunctions of conditions defining the four newly constructed features are as follows:

Leaf 1: $r9shlt < 4$ AND $r7shlt < 3$
Leaf 2: $r9shlt < 4$ AND $r7shlt \geq 3$
Leaf 3: $4 \leq r9shlt < 5$
Leaf 4: $r9shlt \geq 5$

Note that the newly constructed features are defined only in terms of the logical conditions (feature tests) in their corresponding paths in the tree, i.e., the class label predicted by a leaf node is irrelevant for the constructed features. In other words, each instance in the dataset will take a *true* or $false$ value for each of the four above-constructed features based on the values of the features *r9shlt* and *r7shlt*, regardless of the class label predicted by the leaf node at the end of the corresponding path. It is up to the classification algorithm to decide how to combine the values of the constructed features to perform the best possible class predictions.

Table 2 illustrates the calculation of the values of the 4 decision tree path-based features derived from the 4 root-to-leaf paths shown in Fig. 2. The 4 rows in this table represent 4 instances in the dataset. The first two columns show the values of the longitudinal feature "self-report of health" (*shlt*) in waves r7 and r9. The 4 constructed temporal features (CTFs) are shown in the columns with headings CTF-1 through CTF-4, which are derived from the paths ending in Leaf 1 through Leaf 4 in Fig. 2, respectively. These columns show the CTFs' binary values of 1 or 0, denoting the Boolean values "true" or "false", respectively.

In the first row of Table 2, the *shlt* values in waves r9 and r7 are 1 and 2 respectively, which satisfy the conditions "$r9shlt < 4$" and "$r7shlt < 3$" along the path from the root to Leaf 1 in Fig. 2. Therefore, for this first instance, CTF-1 is set to 1 ("true"), and the other CTFs are set to 0 ("false"), since the combination of *r9shlt* and *r7shlt* values in that instance does not satisfy the conditions in any of the paths leading to Leaf 2, Leaf 3 or Leaf 4 in Fig. 2.

Table 2
Examples of the calculation of values for the 4 constructed temporal features (CTFs) derived from the 4 root-to-leaf paths in the decision tree shown in Fig. 2, involving the values of the longitudinal feature "self-report of health" (*shlt*) in waves 7 and 9 (denoted by the prefix r7 and r9 in the feature's acronym in the ELSA database).

r7shlt	r9shlt	CTF-1	CTF-2	CTF-3	CTF-4
2	1	1	0	0	0
3	1	0	1	0	0
5	4	0	0	1	0
4	5	0	0	0	1

The same logic is used to calculate the CTF values for the other 3 instances in Table 2. For example, the fourth instance in this table satisfies the root-to-leaf path leading to Leaf 4 (since that instance has $r9shlt = 5$), so CTF-4 is set to 1 for this instance; and all other 3 CTFs are set to 0 for this instance (since the combination of *r9shlt* and *r7shlt* values for this instance does not satisfy the conditions in any of the paths leading to Leaf 2, Leaf 3 or Leaf 4).

Note that, for each instance (row) in Table 2, exactly one of the CTFs will take the value 1, and the other 3 CTFs will take the value 0. This is a result of the decision tree property that each instance will always satisfy the conditions along exactly one root-to-leaf path.

4. Handling imbalanced classes

Class imbalance is, in general, a challenge in the classification task, particularly in healthcare applications, where the number of individuals (instances) with a diagnosis for a disease (the positive class) is usually much lower than the number of individuals without that diagnosis (the negative class) [50]. This class imbalance threatens the algorithm's ability to learn models that generalize effectively and make precise predictions for the minority (positive) class [51]. To mitigate this issue, we investigate the effectiveness of three major types of class balancing methods, viz., two types of resampling methods (undersampling and oversampling) and the adjustment of class weights. In more detail, in the experiments, we evaluate four variants of undersampling, four variants of oversampling, and two variants of class-weight adjustment, therefore evaluating 10 different class-balancing methods. The goal is to investigate these methods' impact on the predictive model's performance and identify the method(s) that yield the best results in general.

More precisely, the 10 class-balancing methods evaluated in this work are described next by grouping them into the above three types of methods:

1. **Under-sampling:** These methods sample a subset of instances from the majority (negative) class and delete the remaining negative-class instances from the training set. As a result, the training set is reduced to contain all positive instances and only the sampled negative instances. The main disadvantage of this method is that it deletes a large proportion of negative instances from the training set, losing relevant data. A slight advantage is that the classification algorithm will run faster when applied to the reduced training set. Sampling a subset of negative instances can be done randomly (as usual) or using some more sophisticated clustering or K-nearest-neighbour (kNN) approach. We can also vary the proportion of negative instances removed by specifying different values for the desired class ratio, which is defined as $\alpha_{us} = N_{usM}/N_m$, where N_m is the number of instances in the minority class and N_{usM} is the number of instances in the majority class after under-sampling that class. For example, a

class ratio $\alpha_{us} = 1$ means the training set has the same number of instances in the positive and negative classes after undersampling. However, the optimal value of α_{us} is not necessarily 1, and it is worth considering a somewhat larger class ratio, too (since this involves less data loss).

The four undersampling methods evaluated in this work are:

(a) Random Undersampling **(RU1)**: it randomly samples a subset of the majority (negative)-class instances to get a class ratio $\alpha_{us} = 1$.

(b) Random Undersampling **(RU3)**: it randomly samples a subset of the majority-class instances to get a class ratio $\alpha_{us} = 3$.

(c) Cluster Centroids **(CC)**: This technique replaces the majority class instances in the training set with a smaller set of new instances derived from the centroids computed by the k-Means algorithm [52]. This method creates the desired number N of new majority class instances by running the k-Means algorithm with $k = N$ and using each of the computed N centroids as a new majority class instance. In our experiment, we used $N = 136$, which is the same number of instances in the minority (positive) class, to achieve a balanced training set ($\alpha_{us} = 1$).

(d) Condensed Nearest Neighbour **(CN)**: it creates a condensed set of prototypes that accurately represent the majority class to improve classification performance in imbalanced class distributions. This method uses a 1-nearest neighbour (1-NN) rule to assess whether each instance in the majority class should be retained as a prototype or removed [53–55]. CNN iteratively scans the training data, and for each instance of the majority class, if the 1-NN rule misclassifies it, it is considered for inclusion in the prototype set. This iterative refinement continues until no more prototypes are added. The algorithm prioritizes the selection of majority class instances closer to the decision boundary between classes.

2. **Oversampling:** These methods augment the number of instances in the minority class. The desired class ratio is expressed as $\alpha_{os} = N_M/N_{osm}$ where N_{osm} is the number of instances in the minority class after over-sampling, and N_M is the number of instances in the majority class. The oversampling methods evaluated in this work are:

(a) SMOTE **(OS1)**: it uses the Synthetic Minority Oversampling Technique (SMOTE) method [56] to generate extra instances for the minority class to get a class ratio $\alpha_{os} = 1$. SMOTE's parameter number of nearest neighbours was set to the default value of $k = 5$ in our experiments.

(b) SMOTE **(OS3)**: it uses SMOTE to generate extra instances for the minority class to get a class ratio $\alpha_{os} = 3$. Our implementation uses the same parameter settings as OS1.

(c) Borderline SMOTE **(BS)**: Minority-class instances situated on boundaries between classes are more prone to misclassification and are therefore considered crucial for accurate classification. Hence, this method first identifies instances from the minority class that are borderline (i.e. close to the class boundary) in the feature space. Then, only those borderline minority-class instances are used to generate synthetic instances via SMOTE [57]. Note that, among the two variations of Borderline-SMOTE proposed in [57], we use Borderline-SMOTE-1. The Borderline-SMOTE-1 includes two parameters regarding the numbers of nearest neighbours: the parameter "m" employed in step 1 of the algorithm to select borderline examples, a feature absent in the standard SMOTE, and the parameter "k" utilized in step 4 of the algorithm, which is also present in standard SMOTE. This parameter represents the number of nearest neighbours used for generating synthetic instances. In our experiments, we set $m = 10$ and $k = 5$. After applying this method, the training set contains a fully balanced (50%–50%) class distribution ($\alpha_{os} = 1$).

(d) SVM-SMOTE **(SV)**: In this method, the borderline region is estimated by identifying support vectors following the training of SVM classifiers on the training set. Synthetic data is subsequently generated randomly along the lines connecting each support vector of the minority class with several of its nearest neighbours. This method leverages the support vectors, which essentially represent the pivotal instances near the class boundary, to guide the generation of synthetic minority-class instances [58].

3. **Class-weight Adjustment:** This method increases the weights of training instances belonging to the minority class to increase their importance during training. Let pos_w denote the weight of each training instance of the positive (minority) class and neg_w denote the weight of each training instance of the negative (majority) class. These weights are set as follows: $pos_w = (N_{pos} + N_{neg})/N_{pos}$ and $neg_w = (N_{pos}+N_{neg})/N_{neg}$, where N_{pos} represents the number of positive (minority)-class training instances and N_{neg} represents the number of negative (majority)-class training instances. We implemented two variations of this method, as follows:

(a) **CW1**: set class weights as pos_w for minority-class instances and neg_w for majority-class instances, producing a training set with a fully balanced (50%–50%) class distribution.

(b) **CW3**: similar to CW1, but modified so that pos_w is divided by 3 to mitigate the effect of increasing positive-class instances' weights by comparison with the more significant increase in the CW1 method (which might over-emphasize the minority class). Hence, this method produces a training set where the sum of the weights for all majority-class instances is three times the sum for all minority-class instances. This leads to a controlled, fair comparison of CW3's results with the results of its counterpart undersampling and oversampling methods, UR3 and OS3, respectively.

The total number of training instances of each class before and after applying each of the undersampling and oversampling methods is shown in Table 3.

Note that each class-balancing method was applied exclusively to the training set; i.e. the class distribution remains imbalanced in the test set (used to measure generalization performance) since the test set has to reflect the original class distribution of the entire dataset. Applying class balancing to the test set would be unfair and lead to an oversimplified prediction problem that would not reflect the challenging reality of imbalanced classes in the real world.

5. Machine learning pipeline

Fig. 3 illustrates the machine-learning pipeline implemented in this work.

We utilized the harmonized ELSA data version dated July 2021, encompassing data collected from 2002 to 2019, corresponding to waves (time points) 1 through 9. The types of features selected for dataset creation (Step 1 in Fig. 3) are as follows:

1. **Individual demographics:** collected and updated details about respondents' age, gender and years of education.

Table 3
Number of instances of each class (denoted by "no" and "yes" for negative and positive classes) in the training set before and after applying each resampling-based class-balancing method.

Strategy	N	Resampler	Before		After	
			no	yes	no	yes
Under-sampling	1	UR1	6850	136	136	136
	2	UR3			408	136
	3	CC			136	136
	4	CN			379	136
Over-sampling	5	OS1			6850	6850
	6	OS3			6850	2283
	7	BS			6850	6850
	8	SV			6850	6850

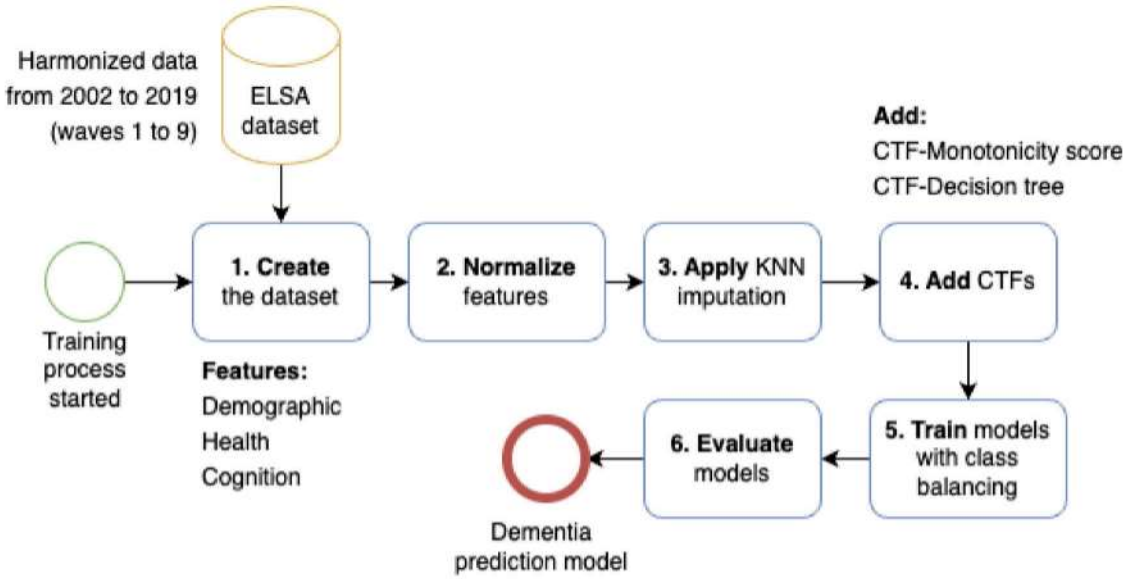

Fig. 3. Machine learning pipeline.

2. **General health:** long-standing illness, diagnoses, symptoms, pain, difficulties with daily activities, mental health, urinary and bowel incontinence, falls and fractures, self-perceived weight and cancer screening.
3. **Cognitive function:** measured different aspects of the respondent's cognitive function, including memory, speed and mental flexibility. Elements included memory and concentration, word list recall, animal naming, backward counting from 20, serial 7s, and naming objects and people.

The dataset contains 38 "base features" without considering the multiple values a longitudinal feature takes across various waves. This set of 38 features includes a few demographic features (e.g. gender) whose values can be represented by a value at just one wave. However, most of the features are longitudinal features, which take values at multiple time points — most of them take values at each of the 9 waves, although some longitudinal features take values at fewer waves. When counting each temporal variation of a longitudinal feature (at each wave) as a separate feature, the total number of features in the dataset is 223. All the 38 base features and the corresponding waves for which they take values are listed in Table A.1 of Appendix.

The dementia class variable has been created by combining the values of the following binary variables from wave 9 of the ELSA database, representing the data collected in 2019.

1. **HEDBDDE:** This variable indicates whether or not (value 1 or 0, respectively) the subject's dementia diagnosis has been confirmed.
2. **HEBIBDE:** This variable indicates whether or not (1 or 0) the respondent has ever reported dementia or memory impairment.
3. **HEDBSDE:** This variable indicates whether or not (1 or 0) the respondent has dementia.

Note that if none of these variables is available for an individual in wave 9, then that individual is not included in the created dataset since the class label cannot be computed. Then, the assignment of a class label to each instance (individual) in the dataset is performed as follows. If any of these three variables have a value of 1 for an individual in ELSA in wave 9, we set the instance's class label as "yes" for dementia. Otherwise, we set the instance's label as "no" for dementia.

We used the k-nearest Neighbour (kNN) method for missing value imputation (Step 3 in Fig. 3). Before applying kNN, each feature's value range was normalized (scaled) to a range between 0 and 1 (Step 2 in Fig. 3) by using Min-Max normalization, i.e. the scaled value x is given by Eq. (1):

$$\frac{x - \min(x)}{\max(x) - \min(x)} \tag{1}$$

where $min(x)$ and $max(x)$ are the minimum and maximum values of the feature in the data. After this normalization, for each feature f, each of its missing values is replaced by f's value in the nearest instance (i.e. $k = 1$ for kNN) among the training instances with a known (non-missing) value for f. The distances between instances were computed using the Gower Distance since it naturally handles instances with missing values [59]. Note that kNN's distance calculation is problematic in datasets with many features because all instances tend to be far away from each other in a very high-dimensional space, i.e. the notion of "nearest neighbour" breaks down. To mitigate this problem, we restrict the features used in kNN's distance calculations to a small set of

Table 4
Effect of different data preparation steps on the number of features in the dementia dataset used in the experiments.

N	Dataset version	Num. of features	Δ
1	Original dataset	222	
2	Dataset after removing features with missing-value rate > 50%	178	−44
3	Dataset after adding monotonicity score-based CTFs	211	+33
4	Dataset after adding decision tree path-based CTFs	343	+132

features, consisting of age, gender, and the values of feature f at other waves (time points), i.e. waves other than the wave where the missing values are currently being inputted, as proposed in [60]. In addition, before using kNN for missing value imputation, we removed features with missing value rates higher than 50% [61].

In Step 4 of the pipeline, we construct two new types of Constructed Temporal Features (CTFs), namely monotonicity score and decision tree path-based CTFs, and add them to the dataset. As detailed in Section 3.1, the monotonicity score ranges from −5 to 5, which was also scaled to the range from 0 to 1 (again, via Min-Max normalization) to be aligned with the scaled range of the other features. The decision tree path-based CTFs are binary features, taking the 1 or 0 to indicate whether or not an instance satisfies the conditions in the corresponding decision tree path, as described in Section 3.2. Table 4 summarizes the impact of each data preparation step, including the removal of features with excessive missing values and the addition of two types of CTF features to the dementia dataset used in the experiments. As shown in this table, the original dataset contained 178 features after removing features with more than 50% of missing values, while the dataset used in the experiments, after adding the CTFs, comprises a total of 343 features.

In Step 5 of the pipeline, we train and evaluate the classifier model using XGBoost, a robust gradient boosting algorithm renowned for its high predictive performance [62], and Random Forest, an ensemble learning method that builds multiple decision trees to improve accuracy and prevent overfitting [17]. In this step, we have also addressed the problem of imbalanced classes using 10 different class-balancing methods involving variants of resampling and class weight-adjustment methods, as discussed in Section 4.

Finally, the predictive accuracies of the models learned by XGBoost and Random Forest are assessed in Step 6 of the pipeline. To assess the models' predictive accuracies, we performed some tuning of the hyperparameter settings for each of these two algorithms, using a nested cross-validation procedure [63] with 5 folds in the outer and the inner loops. For each of the 5 outer loop iterations, an inner loop conducts 5-fold cross-validation on the training set (of the current outer loop iteration) to tune classifier hyperparameter settings. The best classifier hyperparameter settings (i.e., the settings leading to the best F1 score in the inner cross-validation) are then used to learn a model from the entire training set in the current outer loop iteration. The final reported result is, of course, the mean F1 score over the 5 test sets of the outer loop.

Table 5 presents the 8 XGBoost hyperparameters tuned in our experiments and their corresponding candidate values evaluated by the tuning procedure. This procedure consists of 4 sequential runs of a well-known grid search procedure, where each grid search run evaluates all combinations of candidate values for two hyperparameters. In the first run, the grid search evaluates all combinations of the first two hyperparameters in Table 5 (learning rate and *N* estimators), keeping the other 6 hyperparameters fixed at their default values. Then, it selects the best combination of those two hyperparameters' values (with the best F1 score on the inner cross-validation). Then, the second grid search run evaluates all combinations of the third and fourth hyperparameters in Table 5, fixing the first two hyperparameters to the values selected in the first grid search run and using the default values for the remaining hyperparameters (5–8) in Table 5. This same process is used at the third and, finally, the fourth run of the grid search so that, at the end of that fourth run, the 8 hyperparameters have been tuned,

This 4-iteration grid search procedure was used because it is much more computationally efficient (faster) than tuning the 8 hyperparameters via a single-iteration grid search as usual. More precisely, the total number of XGBoost (XGB) configurations evaluated in each of the four grid search runs is as follows:

- 1st run: 40 XGB configurations (4 Learning rates×10 N estimators' values)
- 2nd run: 12 XGB configurations (4 Max depths × 3 min child weights)
- 3rd run: 36 XGB configurations (6 Subsamples × 6 Colsamples)
- 4th run: 49 XGB configurations (7 Gamma values × 7 Lambda values)

Hence, this procedure evaluates 137 XGB configurations. For evaluating each XGB configuration, XGB is executed 5 times across the 5-fold inner cross-validation (CV). Thus, the number of XGB runs in each iteration of the outer CV is $137 \times 5 = 685$. Therefore, the total number of XGB configurations evaluated in the entire nested CV is 3,425 (685 × 5 outer CV folds). In contrast, if we were to assess all hyperparameter combinations in a single-iteration run of the grid search as usual, the total number of evaluated configurations in the nested CV would be $4 \times 10 \times 4 \times 3 \times 6 \times 6 \times 7 \times 7 \times 5 \times 5 = 21,168,000$, which would be an infeasible experiment in practice.

Similarly, Table 6 shows 8 RF hyperparameters, of which 5 are set at a fixed value and 3 are tuned, namely: (a) *min samples leaf*: the minimum number of samples (instances) required in a leaf node; (b) *max features or mtry*: the number of randomly sampled features evaluated for data splitting at each tree node, and (c) *max samples or sample size*: the proportion of total training samples (instances) used for learning each tree in the forest. These hyperparameters' candidate values used by the tuning procedure are also shown in Table 6. For the 5 RF hyperparameters that were kept fixed, 4 were kept at their default value in the scikit-learn library; the only exception was *N estimators* (the number of trees in the forest), which was increased from 100 to 300 to get more robust RF models.

We tuned only 3 RF hyperparameters because, in general, the RF algorithm is more robust to hyperparameter setting variations than XGBoost, and these 3 tuned hyperparameters are, in general, the most important hyperparameters of the RF algorithm [64]. Since only 3 RF hyperparameters are tuned, the grid search method evaluates all possible combinations of candidate values for those hyperparameters, yielding $5 \times 5 \times 5 = 125$ RF configurations. This approach provides an evaluated number of configurations comparable to that of XGBoost, i.e. roughly the same "computational budget" is used to tune the hyperparameters of RF and XGBoost.

It is important to emphasize that, for both XGBoost and RF, this hyperparameter tuning process was performed solely using the training data, with the test data reserved exclusively for measuring generalization performance, as usual.

The final dataset had 8,732 instances (participants in the ELSA study), out of which only 169 (1.9%) have the positive class label (dementia), and the other 8,563 (98.1%) instances have the negative class label (non-dementia). This highlights the importance of using class-balancing methods, as investigated in our experiments. Table 3 (Section 4) shows the number of instances in the training set separately for the negative class (non-dementia) and the positive class (dementia) before and after applying each of the 8 variants of resampling-based class-balancing methods used in the experiments.

Table 5
XGBoost hyperparameters and their candidate values evaluated by the sequential grid search procedure.

N	Parameter	Description	Values
1	Learning rate	Determines how fast the XGBoost model learns. It controls the magnitude of modification of each additional tree to the overall model during boosting.	0.0001, 0.001, 0.01, 0.1
2	N estimators	Specifies the number of decision trees to be built and boosted. Setting N estimators equal to 1 results in a single decision tree without boosting.	50, 100, 150, 200, 250, 300, 350, 400, 450, 500
3	Max depth	Specifies the maximum depth of a tree; used to mitigate overfitting.	3, 5, 7, 9
4	Min child weight	Sets the minimum sum of weights of all instances required in a child node. Further partitioning stops if the sum of instance weights in a leaf node is less than min_child_weight.	1, 3, 5
5	Subsample	Specifies the fraction of instances randomly selected by XGBoost for constructing each tree. For instance, 0.5 means half of the instances are randomly sampled to grow each tree.	0.4, 0.5, 0.6, 0.7, 0.8, 0.9
6	Colsample by tree	Determines the fraction of columns (features) randomly sampled by XGBoost for constructing each tree.	0.4, 0.5, 0.6, 0.7, 0.8, 0.9
7	Gamma	Specifies the minimum loss reduction necessary to split a node. This parameter serves as pseudo-regularization in gradient boosting, where higher values induce stronger regularization, making the algorithm more conservative.	0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6
8	Lambda	Represents the L2 regularization term applied to weights, akin to Ridge regression. This constant is added to the loss function's second derivative (Hessian) during gain and weight calculations.	0, 0.5, 1, 1.5, 2, 3, 4.5

Table 6
RF hyperparameters and their candidate values evaluated by grid search procedure.

N	Parameter	Description	Values
1	N estimators	Specifies the number of trees in the forest, which contributes to model stability and accuracy.	300
2	Max depth	Sets the maximum depth of each tree. 'None' indicates trees expand until all leaves are pure or have fewer than the minimum samples required to split.	None
3	Min samples split	Minimum number of samples (instances) required to split an internal node.	2
4	Min samples leaf	Minimum number of samples (instances) required in a leaf node.	1, 2, 3, 5, 7
5	Bootstrap	Determines whether the training instances for each tree are drawn with replacement. True enables each tree to be trained on a unique bootstrap sample, enhancing randomness and robustness.	True
6	Max features (mtry)	The number of randomly sampled features considered for splitting the data at each tree node.	4, 7, 13, 28, 37
7	Max samples (Sample size)	Fraction of total samples (instances) drawn for learning each tree. Smaller samples reduce training time.	1, 0.95, 0.9, 0.85, 0.80
8	Criterion (Splitting rule)	Function to measure the quality of a split at each node. Gini impurity is common for classification.	gini

6. Computational results and discussion

This section is divided into three parts. Section 6.1 discusses the predictive accuracy results of the learned XGBoost and Random Forest models. Section 6.2 discusses the interpretation of the best model learned by each of these two types of algorithms by identifying the most important predictive features in those models. Finally, Section 6.3 discusses some limitations of this work.

6.1. Predictive accuracy results

This subsection presents the results evaluating the predictive accuracy of dementia classification models using various feature sets (including original features from the data and/or the CTFs derived from it) and different class-balancing methods. We trained two types of classification models: XGBoost and Random Forest (RF).

We report results for three performance measures: the F1-score, Precision, and Recall. The F1 score is the harmonic mean of Precision and Recall, calculated as $F1 = (2 \times Precision \times Recall)/(Precision + Recall)$. Precision is calculated as $Precision = TP/(TP + FP)$, where TP is the number of true positives (i.e., positive instances correctly predicted as positives) and FP is the number of false positives (i.e., negative instances wrongly predicted as positives). Recall is calculated as: $Recall = TP/(TP + FN)$, where FN is the number of false negatives (i.e., positive instances wrongly predicted as negatives). The positive and negative classes represent the presence or absence of dementia (respectively) in each individual.

Tables 7–12 show the results for the F1-Score, Precision and Recall measures for XGBoost and RF. In each of these tables, the columns refer to the 10 variants of class-balancing methods described in Section 4 and the baseline approach of not using any class-balancing method, and the rows refer to 7 different subsets of features (specified in more detail below). Hence, each table reports the results for 77 (11×7) combinations of a feature subset and a class balancing method or baseline approach.

The seven feature subsets investigated in the experiments are as follows:

1. **Original (OR):** composed solely of the original features in the created dataset, derived directly from the ELSA database, i.e. not including any CTFs.
2. **Monotonicity-score (MO):** composed solely of the monotonicity-score CTFs, i.e. not including any of the original features in the created dataset.
3. **Decision tree (DT):** composed solely of the decision tree path-based CTFs.
4. **Original and Monotonicity-score (OR+MO):** containing both original features and monotonicity-score CTFs.

Table 7
F1-Scores for the features set and class-balancing combinations in XGBoost.

Approach[1]	RU1[2]	RU3	CC	CN	OS1	OS3	BS	SV	CW1	CW3	NONE	AR
OR	0.241	0.341	0.039	0.355	0.382	0.370	0.411	0.385	0.437	0.461	0.371	3.3
MO	0.092	0.164	0.037	0.145	0.098	0.149	0.108	0.101	0.163	0.218	0.083	7.0
DT	0.249	0.385	0.039	0.369	0.345	0.332	0.363	0.428	0.420	0.419	0.300	3.9
OR+MO	0.237	0.371	0.038	0.376	0.392	0.368	0.422	0.401	0.416	0.451	0.372	3.5
OR+DT	0.253	0.378	0.039	0.383	0.364	0.351	0.387	0.413	0.434	0.437	0.360	3.1
MO+DT	0.247	0.379	0.038	0.386	0.308	0.357	0.384	0.389	0.389	0.446	0.313	4.4
OR+MO+DT	0.248	0.380	0.038	0.394	0.341	0.363	0.365	0.407	0.429	**0.462**	0.388	2.9
AR	9.9	5.9	11.0	5.3	7.1	7.4	4.7	3.5	2.4	1.3	7.6	

Table 8
F1-Scores for the features set and class-balancing combinations in Random Forest.

Approach[1]	RU1[2]	RU3	CC	CN	OS1	OS3	BS	SV	CW1	CW3	NONE	AR
OR	0.229	0.383	0.038	0.459	0.175	0.270	0.200	0.221	0.462	0.487	0.321	4.0
MO	0.054	0.178	0.039	0.115	0.066	0.088	0.061	0.064	0.122	0.223	0.058	6.5
DT	0.259	0.396	0.038	0.352	0.334	0.312	0.277	0.319	0.461	0.453	0.235	2.8
OR+MO	0.232	0.379	0.038	0.448	0.210	0.288	0.170	0.231	0.489	0.489	0.263	3.8
OR+DT	0.250	0.395	0.038	0.436	0.209	0.299	0.200	0.279	0.444	**0.496**	0.234	3.7
MO+DT	0.262	0.394	0.038	0.339	0.303	0.294	0.267	0.286	0.459	0.470	0.251	3.5
OR+MO+DT	0.253	0.394	0.038	0.406	0.233	0.279	0.197	0.285	0.461	0.473	0.259	3.8
AR	8.1	3.4	11.0	3.4	7.6	5.7	9.0	6.7	1.9	1.2	7.9	

Table 9
Precision values for the features set and class-balancing combinations in XGBoost.

Approach[1]	RU1[2]	RU3	CC	CN	OS1	OS3	BS	SV	CW1	CW3	NONE	AR
OR	0.139	0.220	0.020	0.306	**0.565**	0.550	0.547	0.552	0.452	0.456	0.544	2.7
MO	0.051	0.106	0.019	0.260	0.074	0.244	0.096	0.094	0.109	0.201	0.330	6.8
DT	0.145	0.259	0.020	0.419	0.468	0.539	0.504	0.562	0.408	0.425	0.461	3.4
OR+MO	0.137	0.242	0.020	0.000	0.519	0.519	0.550	0.526	0.413	0.450	0.552	3.6
OR+DT	0.147	0.252	0.020	0.376	0.511	0.519	0.541	0.548	0.464	0.441	0.517	3.0
MO+DT	0.144	0.254	0.020	0.437	0.433	0.517	0.518	0.491	0.365	0.419	0.484	4.3
OR+MO+DT	0.144	0.251	0.019	0.000	0.513	0.504	0.501	0.520	0.441	0.450	0.555	4.3
AR	9.7	8.3	10.7	7.4	4.6	3.1	3.4	2.9	6.9	6.0	3.0	

Table 10
Precision values for the features set and class-balancing combinations in Random Forest.

Approach[1]	RU1[2]	RU3	CC	CN	OS1	OS3	BS	SV	CW1	CW3	NONE	AR
OR	0.131	0.249	0.019	0.505	0.499	0.583	0.557	0.560	0.441	0.503	0.668	3.4
MO	0.029	0.124	0.020	0.209	0.056	0.153	0.061	0.073	0.073	0.177	0.172	6.3
DT	0.152	0.270	0.019	0.520	0.557	0.504	0.555	0.530	0.422	0.429	0.571	4.1
OR+MO	0.134	0.248	0.019	0.000	0.688	0.584	0.651	0.577	0.479	0.492	**0.714**	2.7
OR+DT	0.145	0.262	0.019	0.552	0.582	0.570	0.548	0.576	0.422	0.474	0.601	3.5
MO+DT	0.154	0.268	0.019	0.504	0.539	0.515	0.533	0.457	0.432	0.435	0.550	4.4
OR+MO+DT	0.147	0.263	0.019	0.000	0.608	0.550	0.641	0.536	0.438	0.466	0.619	3.5
AR	9.7	8.1	10.7	6.1	3.9	4.0	4.0	4.6	7.5	5.9	1.4	

Table 11
Recall values for the features set and class-balancing combinations in XGBoost.

Approach[1]	RU1[2]	RU3	CC	CN	OS1	OS3	BS	SV	CW1	CW3	NONE	AR
OR	0.900	0.763	1.000	0.433	0.296	0.285	0.338	0.302	0.450	0.479	0.290	3.0
MO	0.461	0.372	0.940	0.101	0.160	0.113	0.136	0.119	0.343	0.242	0.048	7.0
DT	0.882	0.746	1.000	0.338	0.279	0.255	0.302	0.361	0.450	0.420	0.232	4.6
OR+MO	0.906	0.799	1.000	0.432	0.320	0.291	0.349	0.331	0.432	0.468	0.285	2.6
OR+DT	0.905	0.769	1.000	0.403	0.297	0.273	0.326	0.344	0.444	0.456	0.279	3.5
MO+DT	0.888	0.752	1.000	0.374	0.249	0.285	0.315	0.338	0.432	0.479	0.237	4.5
OR+MO+DT	0.906	0.781	1.000	0.409	0.267	0.290	0.297	0.344	0.444	0.491	0.302	2.8
AR	2.0	3.0	1.0	6.6	9.0	10.0	7.7	7.3	4.8	4.3	10.3	

5. **Original and Decision Tree (OR+DT):** containing both original features and decision tree path-based CTFs.
6. **Monotonicity score and Decision Tree (MO+DT):** containing both monotonicity score CTFs and decision tree path-based CTFs, but not including any original feature.
7. **Original, Monotonicity score and Decision Tree (OR+MO+DT):** containing all available features, including the original features, monotonicity score CTFs, and decision tree path-based CTFs.

The cells in the tables were shaded with a gradient from red (worst) to green (best) to indicate performance levels. For each column (representing a class-balancing method), the underlined value shows which feature subset (row) achieved the best result for that particular method. The highest overall result is highlighted in bold.

In each table (i.e., for each measure), the last column shows each feature set's Average Rank (AR), and the last row shows the AR of each class-balancing method. The AR of each feature set was computed by first computing the rank of each feature subset in each column (i.e. for each class-balancing method), where the feature set with the best (highest) value in a column is assigned rank one and the worst feature set in that column is assigned rank 7. In the case of ties, the ranks are evenly distributed between the tied feature sets, e.g. if two feature sets are tied as the best in a column, each feature set is assigned rank 1.5.

Table 12
Recall values for the features set and class-balancing combinations in Random Forest.

Approach[1]	RU1[2]	RU3	CC	CN	OS1	OS3	BS	SV	CW1	CW3	NONE	AR
OR	0.899	0.823	1.000	0.432	0.107	0.178	0.124	0.142	0.491	0.479	0.219	3.9
MO	0.773	0.343	0.893	0.083	0.089	0.065	0.071	0.065	0.378	0.302	0.036	7.0
DT	0.893	0.757	1.000	0.278	0.249	0.237	0.189	0.237	0.520	0.497	0.154	2.9
OR+MO	0.893	0.805	1.000	0.415	0.125	0.196	0.101	0.148	0.509	0.491	0.171	3.9
OR+DT	0.899	0.805	1.000	0.374	0.130	0.207	0.125	0.189	0.485	0.526	0.154	3.5
MO+DT	0.893	0.745	1.000	0.266	0.219	0.213	0.184	0.213	0.502	0.520	0.166	3.5
OR+MO+DT	0.899	0.787	1.000	0.356	0.148	0.196	0.125	0.201	0.503	0.491	0.172	3.4
AR	2.0	3.1	1.0	6.1	8.7	8.1	10.1	8.5	4.1	4.7	9.4	

Notes for interpreting Tables 7–12:
(1) Features sets: OR: Original, MO: Monotonicity-CTF, DT: Decision-Tree-CTF, OR+MO: Original + Monotonicity-CTF, OR+DT: Original + Decision-Tree-CTF, OR+MO+DT: Original + Monotonicity-CTF + Decision-Tree-CTF
(2) Class balancing methods: UR1: undersampling random with ratio 1, UR3: undersampling random with ratio 3, CC: Cluster Centroid, CN: Condensed Nearest Neighbour, OS1: oversampling SMOTE with ratio 1, OS3: oversampling SMOTE with ratio 3, BS: Borderline SMOTE, SV: SVM SMOTE, CW1: class weight with ratio 1, CW3: class weight with ratio 3
(3) The best result in each column is underlined, indicating the best feature set for the class-balancing method in that column.
(4) The best result in each row is highlighted in light grey colour, indicating the best class-balancing method for the feature set in that row.

Next, the AR of each feature set is simply the arithmetic mean of that feature set's ranks across the 11 columns (class-balancing methods). The AR of each class-balancing method is computed analogously by computing the mean of the ranks of each method across the seven rows (feature sets). Note that, for both feature sets and class-balancing methods, the lower the AR value, the better that feature set or method is.

Tables 7 and 8 present the F1-scores alongside their corresponding Average Ranks (ARs), for XGBoost and RF, respectively. For XGBoost, the highest overall F1 score of 0.462 was achieved by combining the original (OR), CTF-MO, and CTF-DT features with the CW3 class-balancing method. In this method, instance weights are adjusted so that the total weight for majority-class instances is three times that of minority-class instances. Similarly, for RF, the top F1-score of 0.496 was obtained by combining the original (OR) and CTF-DT features, also using the CW3 method. Notably, RF slightly outperformed XGBoost, with both models favouring CW3 and utilizing DT CTFs in addition to the original features.

In addition, on average, across all feature sets, CW3 was the best class-balancing method, with an AR of 1.3 and 1.2 (for XGBoost and RF, respectively), substantially better than the second-best class-balancing method, CW1, with an AR of 2.4 and 1.9 for XGBoost and RF. Interestingly, the two best class-balancing methods were the two variations of the class-weight adjustment approach, which, on average, performed substantially better than all variations of the resampling-based approach. This pattern was consistent for both XGBoost and RF models.

For XGBoost, across all class-balancing methods, the most effective feature set included all feature types (OR + CTF-MO + CTF-DT), achieving an Average Rank (AR) of 2.9, followed closely by OR + DT, with an AR of 3.1. Notably, the feature set OR + MO had an AR of 3.5, slightly worse than the AR of 3.3 for OR alone. This indicates that the MO CTFs improved the F1-score only when combined with both OR and DT CTFs, while the DT CTFs were more successful overall, as previously noted. For RF, the best feature set across all class-balancing methods consisted of DT CTFs, with an AR of 2.8, followed by MO CTFs + DT CTFs, with an AR of 3.5. It is worth noting that feature sets yielding the best results in both XGBoost and RF models included DT CTFs.

Tables 9 and 10 display the precision measures. For XGBoost, the highest precision, 0.565, was achieved by combining SMOTE with a class ratio of $\alpha_{os} = 1$ (OS1) and OR features. The top-ranking feature set was OR, with an Average Rank (AR) of 2.7, while the best class-balancing method was SVM-SMOTE (SV), with an AR of 2.9, closely followed by the baseline of not using any class-balancing method (NONE), with an AR of 3.0. This suggests that class balancing generally tends to decrease, rather than improve, precision. For RF, the highest precision, 0.714, was achieved by combining OR and MO CTFs without any class-balancing method. The best ARs were 2.7 (for feature sets) and 1.4 (for class-balancing methods), achieved by using OR + MO-CTFs and omitting any class-balancing adjustments.

Finally, Tables 11 and 12 present the recall measures. For both, XGBoost and RF models, the best overall recall was 1.0, obtained using CC as the class balancing method and almost any feature set. CC had an AR of 1.0. However, Tables 9 and 10 make it clear that CC was the worst method regarding precision, and this poor trade-off between precision and recall made CC also the worst method regarding F1 score, as shown in Tables 7 and Tables 8.

Note that, among the three measures, the F1 score can be considered the most important one since it considers the trade-off between precision and recall. Hence, in summary, regarding the F1 score, overall, the best class-balancing method was CW3, the best feature set was OR+MO+DT (for XGBoost) and OR+DT (for RF), and the best F1 score was obtained by combining CW3 with one of these two feature sets (depending on the classifier). These two combinations do not particularly excel regarding either precision or recall separately, but these two combinations achieve the best trade-off between precision and recall.

6.2. *Interpreting the best models via feature importance analysis*

We have performed an interpretation of the best dementia classification model (i.e. the model with the highest F1-score) learned by XGBoost in our experiments, which was the model using the combination of original (OR), CTF-MO and CTF-DT constructed features, as well as using CW3 as the class-balancing method. Similarly, we performed an interpretation of the best model for RF in our experiments, which used OR + CTF-DT and CW3 as the class-balancing method.

These two models' interpretation consisted of identifying the most important predictive features in each of these models, based on a feature importance measure suitable for ensembles of decision trees like XGBoost and RF; more precisely, the Gini Index Reduction (GIR) measure [65,66].

The GIR measure essentially works as follows: for each feature, for each node in a decision tree in the ensemble that is labelled by that feature, compute how much the Gini Index (a measure of class impurity) was reduced when the data at that node was partitioned based on the values of that feature. Then, compute the weighted average of this Gini Index reduction for that feature by averaging over all decision tree nodes (in all trees in the ensemble) labelled by that feature, where each Gini Index reduction value's weight is proportional to the number of instances in the corresponding decision tree node. Hence, the larger the GIR value for a feature, the greater the importance of that feature in the ensemble model.

In Table 13, we present the top 10 features with the highest value of the GIR feature importance measure in a model learned by XGBoost using OR+MO+DT as the feature set and CW3 as the class-balancing

Table 13
The 10 most important features (based on the Gini Index Reduction GIR measure of feature importance) in the best-performing XGBoost model for dementia classification in our experiments, which was the model using the combination of original (OR), CTF-MO and CTF-DT features with class-weight adjustment (CW3) as the class-balancing method.

N	Feature[1]	Feat. type[2]	Feat. spec.[3]	Description	GIR[4]
1	**iadltot2**	DT	$r9iadltot2 \geq 0.38$	A summary of 9 IADLS scores: using the phone, managing money, taking medications, shopping for groceries, preparing hot meals, using a map, doing housework, recognizing when in danger, and communicating.	0.174
2	**mo**	DT	$r9mo < 0.5$ AND $r6mo < 0.5$	Cognition measure for cognition date naming-month.	0.093
3	**tr20**	DT	$r9tr20 \geq 0.375$ AND $r8tr20 \geq 0.275$	Summary score for immediate and delayed word recall.	0.072
4	**dlrc**	DT	$r8dlrc \geq 0.25$ AND $r7dlrc \geq 0.45$	Cognition measure for delayed word recall.	0.025
5	**orient**	DT	$r9orient < 0.375$	Cognition measure for the orientation to date, month, year and day of the week.	0.025
6	**mo**	OR	r7mo	*As described in row 2 of this table.*	0.016
7	**sight**	MO	*sight*	The respondent's self-rated eyesight while wearing glasses or corrective lenses as usual.	0.015
8	**tr20**	OR	r9tr20	*As describeded in row 3 of this table.*	0.014
9	**iadltot2**	DT	$r9iadltot2 < 0.167$ AND $r8iadltot2 < 0.055$	*As described in row 1 of this table.*	0.013
10	**vgactx**	DT	$r9vgactx < 0.833$ AND $r4vgactx \geq 0.833$	The frequency of vigorous physical activity.	0.013

Notes for interpreting Tables 13–14:
(1) Feature name, which is a mnemonic assigned to the feature as used in harmonized ELSA documentation. This name does not consider the feature's wave(s), specified in the column "Feat. spec." – see Note (3).
(2) Feature type, where OR represents an original feature (directly extracted from the ELSA database), MO denotes a constructed temporal feature (CTF) representing a monotonicity score, and DT denotes a CTF representing a path in a decision tree model. These CTFs are defined in detail in Sections 3.1 and 3.2.
(3) The feature specification contains the condition(s) in the decision tree path used to create the feature, in the case of DT features, or simply a feature name in the case of OR or MO features. The feature names, in general, consist of the 'r' prefix, which stands for 'respondent', followed by the feature's wave number and the mnemonic of the corresponding variable in the ELSA database. The exception is that the feature type MO does not have a wave number since the monotonic score is computed over all waves.
(4) The weighted average of Gini Index reduction.

method, and learning the model from the entire dataset (no division between training and test sets). This maximizes the amount of data for learning the model, to get more robust feature importance values. Table 13 shows that 8 of the 10 most important features in this best XGBoost model are new types of constructed temporal features (CTFs) proposed in this work. Among those 8 features, 7 (including the top 5) are decision tree path-based features. The other type of CTF proposed in this work, monotonicity score-based features, had less influence on this best XGBoost model, occurring only once among the 10 most important features. Only two of the 10 features in Table 13 are original features, extracted directly from the ELSA database.

Among the top-10 features in Table 13, 6 features (namely, the features in rows 2–6 and row 8 of the table) refer to cognition measures involving the recall of words or dates, which are intuitively clearly relevant for dementia classification and are supported by the literature [67]. In addition, two of the features in that table refer to an individual's ability to perform instrumental activities of daily living (IADLs), which is also intuitively relevant for dementia classification and is supported by the literature [68,69]. One feature in the table refers to the frequency of vigorous physical activity (*vgactx*). The results of a meta-analysis have shown that vigorous exercise tends to lower the risk of dementia [70]. Finally, the table also includes the feature *sight* (the respondent's self-rated eyesight), which at first glance seems unexpected as a very relevant feature for a dementia classification problem. However, there is evidence that impairment in eyesight is more common among people with dementia [71]. This might be partly explained by the association of eyesight with IDALs (which is captured by the top feature in Table 13, *iadltot2*), since poor eyesight should reduce the ability to perform instrumental activities of daily living.

Similarly, Table 14 presents the top 10 features with the highest value of the GIR feature importance measure in a model learned by RF using OR+DT features and CW3 as the class-balancing method; and again, learning the model from the entire dataset to maximize the amount of data for learning the model and for computing reliable feature importance values. Regarding feature types, 7 out of the top 10 features in this table are original features, whilst the other 3 features are decision tree path-based features (including two features among the top 5 features). Hence, although decision tree path-based CTFs were, in general, less influential for RF than for XGBoost (where all top-5 features were decision tree path-based features, as shown in Table 13), this feature type still had a substantial influence in the best RF model.

Broadly speaking, the set of top-10 features listed in Table 14 (for RF) is similar to the set of top-10 features reported in Table 13 (for XGBoost). More precisely, there are only two features that are included in Table 13 but not included in Table 14, namely *sight* and *vgactx*; and there is only one feature that is in Table 14 but not in Table 13, namely *verbf*, discussed below. Overall, 9 of the 10 features in Table 14 refer to cognition measures, mainly involving the recall of words or dates (the only exception is the feature in row 2); even more than the 6 features of this type in Table 13. In addition, the features in Table 14 have somewhat more emphasis specifically on word recall, with the feature *dlrc* (delayed word recall, further discussed below) occurring 4 times in this table (with variations shown in the column Feat. Spec.); whilst this feature occurs only once in Table 13. The first feature in Table 14, *mo*, refers to a basic cognition measure involving date naming, specifically the month, tests for orientation, a key cognitive domain often impaired in dementia. Disorientation to time, place, or person is a classic early indicator of dementia, thus justifying its importance in this RF model [67].

The second top feature, *iadltot2*, is related to the IADL or Instrumental Activities of Daily Living score and measures independence in tasks like managing finances, medication, and navigation. Difficulty with IADLs is a strong indicator of cognitive decline and is closely related to dementia, where functional impairment in complex daily tasks is common [68,69].

The feature *dlrc* refers to Delayed Word Recall, a memory function typically among the first cognitive domains affected in dementia. Word recall tests measure the ability to retrieve information after a delay, an ability which is notably reduced in dementia, especially in Alzheimer's disease. The *tr20* (Immediate and Delayed Word Recall) feature measures the ability to recall words both immediately and after a delay and tests both short-term and working memory. Deficits here are typical in dementia, where memory deterioration disrupts day-to-day life. The *orient* feature (Orientation to Time) evaluates the understanding of the date, month, year, and day of the week; and involves multiple cognitive processes: attention, memory, and executive functioning. Orientation

Table 14
The 10 most important features in the best-performing RF model for dementia classification in our experiments, which was the model using the combination of original (OR) and CTF-DT features with class-weight adjustment (CW3) as the class-balancing method.

N	Feature[1]	Feat. type[2]	Feat. spec.[3]	Description	GIR[4]
1	**mo**	OR	*r9mo*	Cognition measure for cognition date naming-month.	0.079
2	**iadltot2**	OR	*r9iadltot2e*	A summary of 9 IADLS scores: using the phone, managing money, taking medications, shopping for groceries, preparing hot meals, using a map, doing housework, recognizing when in danger, and communicating.	0.060
3	**dlrc**	OR	*r9dlrc*	Cognition measure for delayed word recall.	0.050
4	**dlrc**	DT	$r8dlrc < 0.25$ AND $r7dlrc < 0.35$	*As described in row 3 of this table.*	0.045
5	**tr20**	DT	$r8tr20 < 0.28$ AND $r7tr20 < 0.43$	Summary score for immediate and delayed word recall.	0.038
6	**dlrc**	OR	r8dlrc	*As described in row 3 of this table.*	0.035
7	**tr20**	OR	r8tr20	*As described in row 5 of this table.*	0.023
8	**dlrc**	OR	r7dlrc	*As described in row 3 of this table.*	0.021
9	**orient**	DT	$r9orient \geq 0.63$ AND $r8orient \geq 0.63$	Cognition measure for the orientation to date, month, year and day of the week.	0.016
10	**verbf**	OR	*r9verbf*	Verbal fluency score. The respondents were asked to name members of animals within a time span of one minute, up to 100 animals. RwVERBF is the count of the number of acceptable animal names.	0.016

loss is highly associated with moderate to severe dementia stages. Finally, *verbf* (Verbal Fluency) is the verbal fluency test, where subjects name animals and evaluate executive functioning and language. Deficits in verbal fluency are common in various dementia types, particularly in frontotemporal dementia and Alzheimer's, where word-finding and processing speed slow down significantly.

6.3. Limitations

This work has several limitations, as follows. First, the dataset used in our experiments was extracted from the English Longitudinal Study of Ageing (ELSA) database, which contains data about individuals in the UK; hence this dataset may have biases related to demographic, cultural, or socio-economic factors specific to this country, potentially limiting the generalizability of the findings to other countries where the above factors could be very different.

In addition, in our experiments, the machine learning algorithms were evaluated by using only one original dataset, with a single class variable to be predicted, although the experiments considered in total 7 different subsets of predictive features (i.e., 7 variations of the original dataset). The focus on variations of a single dataset (with a single class variable) also limits the robustness of our conclusions about the effectiveness of proposed types of temporal constructed features (CTFs).

Furthermore, the value of the class variable being predicted (whether or not a person has Alzheimer's) was extracted from the ELSA database, where that data was collected mainly via interviews with participants, i.e., the class values are mainly based on self-evaluation by participants, rather than formally recorded clinical diagnoses. This means the reliability of the class values for the instances in the dataset is not ideal, i.e. it has a certain degree of "class noise". This may have affected the results of the machine learning algorithms. However, a certain degree of class noise is quite common in real-world biomedical datasets for classification, due to the difficulty of computing precise class labels.

7. Conclusions and future work

This article proposed a new machine learning approach for learning classification models predicting whether or not an individual has dementia using longitudinal data, which consists of repeated measurements of variables across time. The main contributions of this new approach are two proposed types of constructed temporal features (CTFs) that leverage temporal patterns in longitudinal data. The first type of CTFs is based on detecting monotonicity patterns in longitudinal data, i.e. to what extent a feature's value monotonically increases or decreases with time. The second type of CTFs is based on extracting paths of decision trees that detect non-linear combinations in the temporal values of longitudinal features as a heuristic to identify feature value combinations with good predictive power. In this article, these two types of CTFs were called monotonicity scores and decision tree path-based CTFs. In addition, this article also experimented with 10 different class-balancing methods to reduce the large degree of class imbalance in our dementia dataset to try to improve predictive performance.

To evaluate the proposed machine learning approach, we have used two well-known classification algorithms, namely XGBoost and Random Forest (RF), to learn predictive models for different combinations of feature sets (with and without the proposed types of CTFs) and class-balancing methods from a dementia dataset extracted from the English Longitudinal Study of Ageing (ELSA) database. The predictive accuracy of the classification models was evaluated using three measures: the F1-Score, precision and recall — where the F1-score can be considered the most important measure since it considers the trade-off between precision and recall.

The highest F1 scores for both XGBoost and RF models were achieved using a variant of the class-balancing method that adjusts instance weights in the majority class to enhance class balance. For XGBoost, the best F1 score was achieved by combining all available feature types: original features, monotonicity score CTFs, and decision tree path-based CTFs. In contrast, for RF, the optimal performance was obtained using only the original features and decision tree path-based CTFs.

In addition, we identified the 10 most influential features in the best-performing XGBoost and RF models, as follows. Notably, 7 of the top 10 features (including the top 5) for XGBoost were decision tree path-based CTFs, underscoring the value of this feature type for boosting predictive accuracy. Monotonicity score CTFs were less effective; only one feature of this type occurred among the top 10 features for XGBoost. However, monotonicity score CTFs still contributed to the best XGBoost model. The top-10 features for RF closely matched the top-10 features for XGBoost; only one feature (verbal fluency test score) occurred in the top-10 RF features but not in the top-10 XGBoost features. Two of the top 5 features in the best RF model were decision tree path-based CTFs. Overall, most of the top features across both XGBoost and RF models pertain to memory and other cognitive measures, affirming the consistency and coherence of these two types of models.

Future work might include proposing other types of CTFs and adding them to the dataset features, such as features describing social

Table A.1
Features in the created ELSA dataset (based on the ELSA database).

N	Code	Description	Waves where it is filled							
1	AGEY	Age (years) at ivw	1							
2	RAGENDER	Gender	1							
3	RAEDYRSE	Years of education	1							
4	SHLT	Self-report of health	1	2	3	4	5	6	7	8	9
5	HLTHLM	Health problems limit work		2	3	4	5	6	7	8	9
6	ADLTOTE	Some Diff-ADLS:6-Item /0–6	1	2	3	4	5	6	7	8	9
7	IADLTOT2E	Some Diff-IADLs:9-item /0–9				4	5	6	7	8	9
8	JOINTRE	Ever had any joint replacement	1	2	3	4	5	6	7	8	9
9	SIGHT	Self-rated eyesight	1	2	3	4	5	6	7	8	9
10	HEARING	Self-rated hearing	1	2	3	4	5	6	7	8	9
11	FALLNUM	Number of falls	1	2	3		5	6	7	8	9
12	FALLSLPE	Trouble falling asleep				4		6			8
13	PAINFR	Frequent problems with pain	1	2	3	4	5	6	7	8	9
14	URINAI	Any urinary incontinence	1	2	3	4	5	6	7	8	9
15	BREATHE	Short of breath while walking	1	2	3	4	5	6	7	8	9
16	VGACTXE	Freq vigorous phys activ	1	2	3	4	5	6	7	8	9
17	MDACTXE	Freq moderate phys activ	1	2	3	4	5	6	7	8	9
18	LTACTXE	Freq light phys activ	1	2	3	4	5	6	7	8	9
19	DRINK	Ever drinks any alcohol	1	2	3	4	5	6	7	8	9
20	SMOKEV	Smoke ever	1	2	3	4	5	6	7	8	9
21	SHLTC	Change in self-reported hlth		2		4	5	6	7	8	9
22	ADLC	Change-ADLs /0–5		2	3	4	5	6	7	8	9
23	GROSSC	Chg:Walk1/R,Clim1,Bed,Bath/5		2	3	4	5	6	7	8	9
24	FINEC	Chg:Dime,Eat,Dress /0–3		2	3	4	5	6	7	8	9
25	COGIMP	Factors impaired cognition tests	1	2	3	4	5	6	7	8	9
26	SLFMEM	Self-reported memory	1	2	3	4		6	7	8	9
27	READRC	Word recall list read by	1	2	3	4	5	6	7	8	9
28	IMRC	Immediate word recall	1	2	3	4	5	6	7	8	9
29	DLRC	Delayed word recall	1	2	3	4	5	6	7	8	9
30	T20	Total word recall score	1	2	3	4	5	6	7	8	9
31	MO	Cognition date naming-month	1	2	3	4	5	6	7	8	9
32	ORIENT	Summary date naming	1	2	3	4	5	6	7	8	9
33	VERBF	Verbal fluency score	1	2	3	4	5	6	7	8	9
34	NUMERE	Numeracy score	1			4		6	7	8	9
35	BWC20	Backwards Counting From 20							7	8	9
36	SE	Serial 7s							7	8	9
37	SCIS	Object naming scissors							7	8	9
38	MNRC	Current monarch							7	8	9

relationships, socioeconomic status, and genetic data, as well as evaluating these features' impact on the model's predictive performance. It would also be interesting to evaluate the proposed types of CTFs on other types of biomedical longitudinal datasets involving other age-related diseases, since this article focused only on a dementia dataset.

CRediT authorship contribution statement

Flavio Luiz Seixas: Writing – review & editing, Writing – original draft. **Elaine Rangel Seixas:** Writing – review & editing, Writing – original draft. **Alex A. Freitas:** Writing – review & editing, Writing – original draft.

Declaration of competing interest

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

Acknowledgements

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) – Brasil – Finance Code 001. ELSA is funded by the National Institute on Aging (R01AG017644) and by UK Government Departments coordinated by the National Institute for Health and Care Research (NIHR).

Appendix. Features in the dataset used in the experiments

See Table A.1.

Data availability

The authors do not have permission to share data.

References

[1] P. Lloyd-Sherlock, M. McKee, S. Ebrahim, M. Gorman, S. Greengross, M. Prince, R. Pruchno, G. Gutman, T. Kirkwood, D. O'Neill, et al., Population ageing and health, Lancet 379 (9823) (2012) 1295–1296.

[2] M.E. Dewey, P. Saz, Dementia, cognitive impairment and mortality in persons aged 65 and over living in the community: a systematic review of the literature, Int. J. Geriatr. Psychiatry 16 (8) (2001) 751–761.

[3] E. Nichols, J.D. Steinmetz, S.E. Vollset, K. Fukutaki, J. Chalek, F. Abd-Allah, A. Abdoli, A. Abualhasan, E. Abu-Gharbieh, T.T. Akram, et al., Estimation of the global prevalence of dementia in 2019 and forecasted prevalence in 2050: an analysis for the Global Burden of Disease Study 2019, Lancet Public Heal. 7 (2) (2022) e105–e125.

[4] Alzheimer's Association, Alzheimer's disease facts and figures, Alzheimer' s Assoc. Rep. 19 (4) (2023) 1598–1695, http://dx.doi.org/10.1002/alz.13016, Received: 8 February 2023.

[5] G. Livingston, J. Huntley, A. Sommerlad, D. Ames, C. Ballard, S. Banerjee, C. Brayne, A. Burns, J. Cohen-Mansfield, C. Cooper, et al., Dementia prevention, intervention, and care: 2020 report of the Lancet Commission, Lancet 396 (10248) (2020) 413–446.

[6] K.M. Fiest, J.I. Roberts, C.J. Maxwell, D.B. Hogan, E.E. Smith, A. Frolkis, A. Cohen, A. Kirk, D. Pearson, T. Pringsheim, et al., The prevalence and incidence of dementia due to Alzheimer's disease: a systematic review and meta-analysis, Can. J. Neurol. Sci. 43 (S1) (2016) S51–S82.

[7] A. Javeed, A.L. Dallora, J.S. Berglund, A. Ali, L. Ali, P. Anderberg, Machine learning for dementia prediction: A systematic review and future research directions, J. Med. Syst. 47 (1) (2023) 17.

[8] R. Matej, A. Tesar, R. Rusina, Alzheimer's disease and other neurodegenerative dementias in comorbidity: a clinical and neuropathological overview, Clin. Biochem. 73 (2019) 26–31.

[9] C. Holmes, J. Amin, Dementia, Medicine 48 (11) (2020) 742–745, http://dx.doi.org/10.1016/j.mpmed.2020.08.014, URL https://www.sciencedirect.com/science/article/pii/S1357303920302073.
[10] X.-H. Hou, L. Feng, C. Zhang, X.-P. Cao, L. Tan, J.-T. Yu, Models for predicting risk of dementia: a systematic review, J. Neurol. Neurosurg. & Psychiatry 90 (4) (2019) 373–379.
[11] E.Y. Tang, C.I. Price, L. Robinson, C. Exley, D.W. Desmond, S. Köhler, J. Staals, B. Yin Ka Lam, A. Wong, V. Mok, et al., Assessing the predictive validity of simple dementia risk models in harmonized stroke cohorts, Stroke 51 (7) (2020) 2095–2102.
[12] R.A. Hackett, A. Steptoe, D. Cadar, D. Fancourt, Social engagement before and after dementia diagnosis in the English Longitudinal Study of Ageing, PLoS One 14 (8) (2019) e0220195.
[13] A. Maharani, N. Pendleton, I. Leroi, Hearing impairment, loneliness, social isolation, and cognitive function: Longitudinal analysis using English longitudinal study on ageing, Am. J. Geriatr. Psychiatry 27 (12) (2019) 1348–1356.
[14] A. Steptoe, E. Breeze, J. Banks, J. Nazroo, Cohort profile: the English longitudinal study of ageing, Int. J. Epidemiol. 42 (6) (2013) 1640–1648.
[15] J. Banks, G.D. Batty, J. Breedvelt, K. Coughlin, R. Crawford, M. Marmot, J. Nazroo, Z. Oldfield, N. Steel, A. Steptoe, et al., English longitudinal study of ageing: waves 0-9, 1998–2019, 2022.
[16] T. Chen, T. He, M. Benesty, V. Khotilovich, Y. Tang, H. Cho, K. Chen, R. Mitchell, I. Cano, T. Zhou, et al., Xgboost: extreme gradient boosting, 1, (4) 2015, pp. 1–4, R package version 0.4-2 1.
[17] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.
[18] H. Yang, P.A. Bath, The use of data mining methods for the prediction of dementia: evidence from the english longitudinal study of aging, IEEE J. Biomed. Health Inf. 24 (2) (2019) 345–353.
[19] G. Mirzaei, H. Adeli, Machine learning techniques for diagnosis of alzheimer disease, mild cognitive disorder, and other types of dementia, Biomed. Signal Process. Control 72 (2022) 103293.
[20] F.L. Seixas, B. Zadrozny, J. Laks, A. Conci, D.C.M. Saade, A Bayesian network decision model for supporting the diagnosis of dementia, Alzheimer's disease and mild cognitive impairment, Comput. Biol. Med. 51 (2014) 140–158.
[21] E. Stallard, A. Kociolek, Z. Jin, H. Ryu, S. Lee, S. Cosentino, C. Zhu, Y. Gu, K. Fernandez, M. Hernandez, et al., Validation of a multivariate prediction model of the clinical progression of Alzheimer's disease in a community-dwelling multiethnic cohort, 2022, MedRxiv 2022-06.
[22] S. Jamalian, M. Dolton, P. Chanu, V. Ramakrishnan, Y. Franco, K. Wildsmith, P. Manser, E. Teng, J.Y. Jin, A. Quartino, et al., Modeling Alzheimer's disease progression utilizing clinical trial and ADNI data to predict longitudinal trajectory of CDR-SB, CPT: Pharmacometrics & Syst. Pharmacol. 12 (7) (2023) 1029–1042.
[23] Y.-C. Fan, S.-F. Lin, C.-C. Chou, C.-H. Bai, Developmental trajectories and predictors of incident dementia among elderly Taiwanese people: A 14-year longitudinal study, Int. J. Environ. Res. Public Health 20 (4) (2023) 3065.
[24] Y. Wei, X. Chen, C.-B. Schönlieb, S.J. Price, C. Li, Predicting conversion of mild cognitive impairment to alzheimer's disease by modelling healthy ageing trajectories, in: 2023 IEEE 20th International Symposium on Biomedical Imaging, ISBI, IEEE, 2023, pp. 1–5.
[25] V. Gkotzamanis, G. Koliopanos, A. Sanchez-Niubo, B. Olaya, F.F. Caballero, J.L. Ayuso-Mateos, S. Chatterji, J.M. Haro, D.B. Panagiotakos, Determinants of verbal fluency trajectories among older adults from the english longitudinal study of aging, Appl. Neuropsychology: Adult 30 (1) (2023) 110–119.
[26] J. Nyholm, A.N. Ghazi, S.N. Ghazi, J. Sanmartin Berglund, Prediction of dementia based on older adults' sleep disturbances using machine learning, Comput. Biol. Med. 171 (2024) 108126, http://dx.doi.org/10.1016/j.compbiomed.2024.108126, URL https://www.sciencedirect.com/science/article/pii/S0010482524002105.
[27] D. Stamate, H. Musto, O. Ajnakina, D. Stahl, Predicting risk of dementia with survival machine learning and statistical methods: Results on the english longitudinal study of ageing cohort, in: IFIP International Conference on Artificial Intelligence Applications and Innovations, Springer, 2022, pp. 436–447.
[28] D. Cadar, C. Lassale, H. Davies, D.J. Llewellyn, G.D. Batty, A. Steptoe, Individual and area-based socioeconomic factors associated with dementia incidence in England: evidence from a 12-year follow-up in the english longitudinal study of ageing, JAMA Psych. 75 (7) (2018) 723–732.
[29] A. Jorm, A short form of the Informant Questionnaire on Cognitive Decline in the Elderly (IQCODE): development and cross-validation, Psychol. Med. 24 (1) (1994) 145–153.
[30] A. Spooner, E. Chen, A. Sowmya, P. Sachdev, N.A. Kochan, J. Trollor, H. Brodaty, A comparison of machine learning methods for survival analysis of high-dimensional clinical data for dementia prediction, Sci. Rep. 10 (1) (2020) 20410.
[31] A.J. Steele, S.C. Denaxas, A.D. Shah, H. Hemingway, N.M. Luscombe, Machine learning models in electronic health records can outperform conventional survival models for predicting patient mortality in coronary artery disease, PLoS One 13 (8) (2018) e0202344.
[32] H. Kim, H.-W. Chun, S. Kim, B.-Y. Coh, O.-J. Kwon, Y.-H. Moon, Longitudinal study-based dementia prediction for public health, Int. J. Environ. Res. Public Health 14 (9) (2017) 983.
[33] S.-T. Huang, F.-Y. Hsiao, T.-H. Tsai, P.-J. Chen, L.-N. Peng, L.-K. Chen, Using hypothesis-led machine learning and hierarchical cluster analysis to identify disease pathways prior to dementia: Longitudinal cohort study, J. Med. Internet Res. 25 (2023) e41858.
[34] M. Kuhn, K. Johnson, Feature Engineering and Selection: a Practical Approach for Predictive Models, Chapman and Hall/CRC, 2019.
[35] H. Ding, A. Mandapati, A.P. Hamel, C. Karjadi, T.F. Ang, W. Xia, R. Au, H. Lin, Multimodal machine learning for 10-year dementia risk prediction: The framingham heart study, J. Alzheimer' s Dis. (Preprint) (2023) 1–10.
[36] H. Cheng, S. Yuan, W. Li, X. Yu, F. Liu, X. Liu, T.T. Bezabih, De-accumulated error collaborative learning framework for predicting Alzheimer's disease progression, Biomed. Signal Process. Control 89 (2024) 105767.
[37] K. Poonam, R. Guha, P.P. Chakrabarti, Accurate prediction of alzheimer's disease progression trajectory via a novel encoder-decoder LSTM architecture, in: 2023 45th Annual International Conference of the IEEE Engineering in Medicine & Biology Society, EMBC, 2023, pp. 1–4, http://dx.doi.org/10.1109/EMBC40787.2023.10340517.
[38] F. Mandel, R.P. Ghosh, I. Barnett, Neural networks for clustered and longitudinal data using mixed effects models, Biometrics 79 (2) (2023) 711–721.
[39] A.Q. Aminu, N. Torrance, A. Grant, A. Kydd, Is age discrimination a risk factor for frailty progression and frailty development among older adults? A prospective cohort analysis of the English Longitudinal Study of Ageing, Arch. Gerontol. Geriat. (2023) 105282.
[40] C. Ribeiro, A.A. Freitas, A mini-survey of supervised machine learning approaches for coping with ageing-related longitudinal datasets, in: 3rd Workshop on AI for Aging, Rehabilitation and Independent Assisted Living (ARIAL), Held As Part of IJCAI-2019, 2019, pp. 1–5.
[41] T. Pomsuwan, A.A. Freitas, Feature selection for the classification of longitudinal human ageing data, in: 2017 IEEE International Conference on Data Mining Workshops, ICDMW, IEEE, 2017, pp. 739–746.
[42] C. Ribeiro, A. Freitas, Constructed temporal features for longitudinal classification of human ageing data, in: 2021 IEEE 9th International Conference on Healthcare Informatics, ICHI, IEEE, 2021, pp. 106–112.
[43] D. Phillips, Y.-C. Lin, J. Wight, S. Chien, J. Lee, Harmonized ELSA documentation, 2014, Version C.
[44] RAND Center for the Study of Aging, [RAND HRS], 2023, Produced by the RAND Center for the Study of Aging, with funding from the National Institute on Aging and the Social Security Administration.
[45] Harmonized ELSA Documentation VERSION G.2 (2002–2019), JULY 2021, https://g2aging.org/docs/, Accessed: February 19, 2024.
[46] E. Liflyand, S. Tikhonov, A concept of general monotonicity and applications, Math. Nachr. 284 (8–9) (2011) 1083–1098.
[47] A. Atri, The Alzheimer's disease clinical spectrum: diagnosis and management, Med. Clin. 103 (2) (2019) 263–293.
[48] J.M. Fernández Montenegro, B. Villarini, A. Angelopoulou, E. Kapetanios, J. Garcia-Rodriguez, V. Argyriou, A survey of alzheimer's disease early diagnosis methods for cognitive assessment, Sensors 20 (24) (2020) 7292.
[49] V.G. Costa, C.E. Pedreira, Recent advances in decision trees: An updated survey, Artif. Intell. Rev. 56 (5) (2023) 4765–4800.
[50] S. Huda, J. Yearwood, H.F. Jelinek, M.M. Hassan, G. Fortino, M. Buckland, A hybrid feature selection with ensemble classification for imbalanced healthcare data: A case study for brain tumor diagnosis, IEEE Access 4 (2016) 9145–9154.
[51] P. Zhang, Y. Jia, Y. Shang, Research and application of XGBoost in imbalanced data, Int. J. Distrib. Sens. Netw. 18 (6) (2022) 15501329221106935.
[52] W.-C. Lin, C.-F. Tsai, Y.-H. Hu, J.-S. Jhang, Clustering-based undersampling in class-imbalanced data, Inform. Sci. 409 (2017) 17–26.
[53] P. Hart, The condensed nearest neighbor rule (corresp.), IEEE Trans. Inform. Theory 14 (3) (1968) 515–516.
[54] P. Filippakis, S. Ougiaroglou, G. Evangelidis, Condensed nearest neighbour rules for multi-label datasets, in: Proceedings of the 27th International Database Engineered Applications Symposium, 2023, pp. 43–50.
[55] U. Krishnan, P. Sangar, A rebalancing framework for classification of imbalanced medical appointment no-show data, J. Data Inf. Sci. 6 (1) (2021) 178–192.
[56] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: synthetic minority over-sampling technique, J. Artificial Intelligence Res. 16 (2002) 321–357.
[57] H. Han, W.-Y. Wang, B.-H. Mao, Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning, in: International Conference on Intelligent Computing, Springer, 2005, pp. 878–887.
[58] H.M. Nguyen, E.W. Cooper, K. Kamei, Borderline over-sampling for imbalanced data classification, Int. J. Knowl. Eng. Soft Data Parad. 3 (1) (2011) 4–21.
[59] G. Tutz, S. Ramzan, Improved methods for the imputation of missing data by nearest neighbor methods, Comput. Statist. Data Anal. 90 (2015) 84–99.
[60] C. Ribeiro, A.A. Freitas, A data-driven missing value imputation approach for longitudinal datasets, Artif. Intell. Rev. 54 (8) (2021) 6277–6307.
[61] A. Farhangfar, L. Kurgan, J. Dy, Impact of imputation of missing values on classification error for discrete data, Pattern Recognit. 41 (12) (2008) 3692–3705.
[62] T. Chen, C. Guestrin, Xgboost: A scalable tree boosting system, in: Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794.

[63] M. Stone, An asymptotic equivalence of choice of model by cross-validation and Akaike's criterion, J. R. Stat. Soc. Ser. B Stat. Methodol. 39 (1) (1977) 44–47.
[64] P. Probst, M.N. Wright, A.-L. Boulesteix, Hyperparameters and tuning strategies for random forest, Wiley Interdiscip. Reviews: Data Min. Knowl. Discov. 9 (3) (2019) e1301.
[65] A. Behnamian, K. Millard, S.N. Banks, L. White, M. Richardson, J. Pasher, A systematic approach for variable selection with random forests: achieving stable variable importance values, IEEE Geosci. Remote. Sens. Lett. 14 (11) (2017) 1988–1992.
[66] H. Xu, M. Saar-Tsechansky, M. Song, Y. Ding, Using explainable AI to understand team formation and team impact, Proc. Assoc. Inf. Sci. Technol. 60 (1) (2023) 469–478.
[67] F. Thabtah, S. Ong, D. Peebles, Examining cognitive factors for Alzheimer's disease progression using computational intelligence, in: Healthcare, Vol. 10, (10) MDPI, 2022, p. 2045.
[68] S. Cloutier, H. Chertkow, M.-J. Kergoat, I. Gélinas, S. Gauthier, S. Belleville, Trajectories of decline on instrumental activities of daily living prior to dementia in persons with mild cognitive impairment, Int. J. Geriatr. Psychiatry 36 (2) (2021) 314–323.
[69] H.-F. Mao, L.-H. Chang, A.Y.-J. Tsai, W.-N.W. Huang, L.-Y. Tang, H.-J. Lee, Y. Sun, T.-F. Chen, K.-N. Lin, P.-N. Wang, et al., Diagnostic accuracy of instrumental activities of daily living for dementia in community-dwelling older adults, Age Ageing 47 (4) (2018) 551–557.
[70] J. Lee, The relationship between physical activity and dementia: a systematic review and meta-analysis of prospective cohort studies, J. Gerontol. Nurs. 44 (10) (2018) 22–29.
[71] G. Davis, N. Baboolal, V. Tripathi, R. Stewart, Health status risk factors and quality of life in 75–84-year-old individuals assessed for dementia using the short 10/66 dementia diagnostic schedule, PeerJ 9 (2021) e12040.