Norwegian University
of Life Sciences

Master's Thesis 2021 30 ECTS

Faculty of Science and Technology
Professor Cecilia Marie Futsæther

Radiomics Using MR Brain Scans and RENT for Identifying Patients Receiving ADHD Treatment

Nasibeh Mohammadi
Data Science

Acknowledgements

First and foremost, I want to thank my main supervisor *Prof. Cecilia Marie Futsæther*, for her invaluable guidance through the project.

My special thanks go to my co-supervisor *Ass. Prof. Oliver Tomic* for his professional support and encouragement.

I would like to thank, *Postdoc. Stefan Schrunner* for his practical suggestions and *PhD Candidate Anna Jenul*, and *MSc. Ahmed Albuni*, for answering my questions.

I am grateful to the Computational Radiology & Artificial Intelligence (CRAI) Research Group and *Prof. Liesbeth Reneman's group at Clinical Research Unit of the Academic Medical Center, University of Amsterdam, Amsterdam, Netherlands,* and *MSc. Inger Annett Grünbeck for providing the MR images and dataset for analysis.*

Last but not least, I would like to express my deep gratitude to *my family* for their sincere love and support.

Abstract

The core purpose of this thesis was to investigate whether the methylphenidate-based (MPH) treatment of male children patients having attention-deficit/hyperactivity disorder (ADHD) led to changes to five subcortical brain structures (hippocampus, caudate, pallidum, putamen, and thalamus). The methylphenidate treated trials were compared to the placebo group. This was explored by using magnetic resonance (MR) images obtained from the effects of Psychotropic drugs On Developing brain (ePOD) study.

A radiomics approach was exploited to extract descriptors from T1-weighted MR images. Radiomics features including Local Binary Pattern (LBP), shape features and several texture features were derived from the right and left side of the chosen subcortical structures. In this context, a new feature extraction program for generating 3D LBP biomarkers was developed and a new feature selection method Repeated Elastic Net Technique (RENT) appropriate for short-wide datasets were utilised. Thereafter, four different classification experiments were used to predict the medication class (medicated vs placebo) by using a nested cross-validation algorithm and nine supervised classifiers. The area under receiving operator curve (AUC) metric was used for evaluating the performance of classification tasks.

The performance scores suggested that there was a detectable change in the selected brain structures using MPH medication. The classification models showed AUC scores mostly above 85% especially in experiments where LBP features were used as stand-alone features or in addition to standard radiomics features. It appears that the LBP features were the most informative descriptor in this study.

The classification results were approximately the same in experiments with correlated features and without correlated features. Additionally, the higher performance obtained in our study on the same dataset as in a previous study exploiting several feature selectors indicated the capability of our feature selection method (RENT) in selecting robust features.

Contents

List of Abbreviations

AdaBoost	Adaptive Boosting
ADHD	Attention-Deficit/Hyperactive Disorder
AUC	Area Under the Receiver Operating Curve
CBF	Cerebral Blood Flow
DT	Decision Tree Classifier
ePOD	The effects of Psychotropic drugs On Developing brain
ET	Extremely Randomised Tree Classifier
FN	False Negative
FP	False Positive
FPR	False Positive Rate
GLC	Gray Level Co-occurrence
GLCM	Gray Level Co-occurrence Matrix
GLD	Gray Level Distance
GLDM	Gray Level Distance Matrix
GLRL	Gray Level Run Length
GLRLM	Gray Level Run Length Matrix
GLSZ	Gray Level Size Zone
GLSZM	Gray Level Size Zone Matrix
ID	Identification Number
KNN	K Nearest Neighbours
LBP	Local Binary Pattern
LGBM	Light Gradient Boosting Machine
LR	Logistic Regression
ML	Machine Learning
MLP	Multi-Layer Perceptron

MPH	Methylphenidate
MR	Magnetic Resonance
MRI	Magnetic Resonance Imaging
NGTD	Neighbouring Grey Tone Difference
NGTDM	Neighbouring Grey Tone Difference Matrix
RENT	Repeated Elastic Net Technique
RF	Random Forest
Ridge	Ridge Regression
ROC	Receiver Operating Curve
ROI	Region of Interest
SVC	Support Vector Machine Classifier
TN	True Negative
TP	True Positive
TPR	True Positive Rate

1 Introduction and Motivation

Attention-deficit/hyperactivity disorder (ADHD) is a common psychiatric disorder among adolescents [1]–[6]. The most common medication for ADHD is methylphenidate-based treatment (MPH). However, its precise influence on the brain in the long-term is under debate [7], [8]. Since the maturation of the brain structure takes place during childhood, the usage of the drug during this sensitive phase of life can have persistent effects on brain development [9], [10].

The current research is based on the effects of Psychotropic drugs On the Developing brain (ePOD) study [11]. Currently, there are few papers linked to the ePOD study [11]. In this context, the results of Bouziane et al. demonstrates the influence of methylphenidate on the white matter of the brain [4]. Walhovd et al. (2020) assessed the effect of MPH on cortical thickness in ADHD patients [7]. They found that the usage of methylphenidate affected the development of grey matter in the right medial cortex of children. Schrantee et al. in 2016 [12] presented an age-dependent study of the cerebral blood flow (CBF) response to methylphenidate medication. They observed that the subcortical thalamic CBF was reduced in children treated by MPH. Another study in 2020 by Tamminga et al. [8] explored the effect of MPH on the patient's performance after the treatment. They concluded that the improvement of working memory and response speed in ADHD patients was related to the treatment period and not after the treatment.

Furthermore, Grünbeck (2020) examined the changes in the grey matter of the human brain caused by MPH treatment using radiomics [13]. Grünbeck performed several classification tasks and used various feature selection methods to examine the impact of MPH medication on the five subcortical structures of the brain, including the hippocampus, caudate, thalamus, putamen, and pallidum. Her study found that some image features, particularly from pallidum and putamen, appear to be associated with MPH treatment, but these findings required further confirmation.

Radiomics is a developing field of study that aims to mine quantitative biomarkers from medical images to help the clinical decision-making process [14], [15]. Radiomics exploits advanced technologies in artificial intelligence to ameliorate the accuracy of diagnosis and treatment based on the extracted radiomics features [16]. Radiomics

features refer to the different types of features that can be derived from an image. Generally, they are categorised into four main groups (shape-based, intensity-based, texture-based, and higher-order features) [17]. Radiomics utilisation needs programming and machine learning (ML) knowledge. In this context, having a standard and user-friendly tool for researchers, scientists, radiologists, and oncologists to extract reproducible and comparable biomarkers from images is demanding. The Biorad framework [18], [19] using the pyradiomics package [20] tried to address this issue. The pyradiomics package covered common methods for extracting image texture features like the Grey Level Co-occurrence Matrix, Grey Level Run Length Matrix and so forth. However, the powerful feature extraction method Local Binary Patterns (LBP) [21] was not included.

In radiomics studies, issues regarding medical image acquisition and the privacy policies regarding patient information complicate the sample gathering process [17]. In addition, in the feature extraction phase of radiomics, many biomarkers are extracted from the medical images. In this context, radiomics studies suffer from high dimensional data and few samples [22]. Repeated Elastic Net Technique (RENT) [23] is a brand-new user-friendly feature selection tool that works by training several ensemble sub-models on unique subsets of the dataset. The authors claimed that it is appropriate for short-wide datasets and that it provides high performance relative to other feature selection methods as Laplacian score, relief, mRMR and Fisher score [23].

In this thesis, our primary objectives were to construct a robust classification model and extended the radiomics dataset in Grünbeck's study of MPH on adolescent brains [13]. We utilised the radiomics approach to analyse the changes caused by ADHD medication in five subcortical structures of the brain by comparing treated patients to the control (placebo) group based on the images of the ePOD study [11]. In this thesis, we developed a 3D LBP extraction module that is not included in the pyradiomics package [20] and added it to the Biorad framework [19]. Thereafter, we examined classification results based on LBP features and compare them to the results obtained from other texture features and shape features. We tested RENT for selecting features and examined the robustness of features selected by RENT by modelling.

All in all, the goals of this thesis were: 1) examine whether MPH medication alters the brain structure of ADHD-diagnosed ten- to twelve-year-old male patients; 2) explore the entire radiomics pipeline for an ADHD study; 3) develop a feature extraction tool for 3D LBP features; 4) explore the efficiency of RENT as a feature selection tool; 5) employ methods for examining the short-wide datasets; 6) tackle the lack of unseen data in the radiomics study.

This thesis' chapters are structured according to the IMRaD (Introduction, Method, Results and Discussion) format [24]. Chapter 1 contains a brief introduction to our work and motivations. Chapter 2 contains the theoretical background of the thesis. Methods and Materials used in this thesis are outlined in chapter 3. The thesis's findings and experimental results are described in chapter 4. A discussion of the results and observations, and suggestions for future work are covered in chapter 5. In chapter 6, the conclusion of the goals of this thesis is given. The results that were not covered in chapter 4 are presented in chapter 7 as appendices.

2 Theory

2.1 Attention-Deficit/Hyperactivity Disorder

Attention-deficit/hyperactivity disorder (ADHD) is among the most frequently diagnosed childhood neurodevelopmental disorders, with an overall prevalence of 5%–8% in children worldwide [1], [4]–[6]. Boys are twice as prone to be affected by ADHD as girls [6]. ADHD manifests itself with symptoms such as hyperactivity, severe impulsiveness, distractibility, and inattention. Therefore, it adversely affects social, educational, and emotional activities [1]–[3], [6], [25]. These side effects may continue into adulthood and result in a long-lasting impairment [5], [25].

Methylphenidate (MPH) treatment is a viable and safe medication prescribed broadly for ADHD patients; however, its exact neurochemical behaviour is under discussion, and knowledge about its long-term side effects on the children's brains is limited [3], [9], [25].

Adolescence and childhood are exceptionally sensitive and susceptible time of brain development. During this time, the development of several parts of the brain happens. Hence, medications given during the delicate beginning stages of life may influence neurodevelopmental directions that can have more significant impacts later in life [9], [10].

Studies on Magnetic Resonance Imaging (MRI) have shown that stimulant medication influences brain development, to such an extent that untreated kids with ADHD show faster cortical thinning and smaller white matter volumes than children with ADHD using stimulant prescription [10]. Studies of medical images can play a helpful role in diagnosing and treating diseases and examining long-term changes in brain structure due to medication [1].

Recently, MRIs have been widely used in the study of patient's brain structures [25]. MRIs empower research to examine the structure of brains noninvasively. Thereby, it is possible to study different brain tissues (white matter and grey matter) and various cortical and subcortical brain structures [5].

2.2 Radiomics

Recently, the advancement of digital medical records in clinics and hospitals and the availability of medical images have facilitated the introduction of a new approach to extract data from medical images, called "Radiomics" [17], [26]. Radiomics is concerned with the concept that radiological images can reveal information that is not visible to the human eye. Radiomics investigates the quantitative features of digital images and converts the images into mineable data, incorporated into clinical decision-support [17], [27]–[30].

The radiomics pipeline (Figure 1) comprises several steps, which will be discussed in this chapter. The steps are 1) image acquisition, 2) segmentation, 3) feature selection, 4) feature extraction, and 5) modelling and evaluation.

Figure 1. Radiomics pipeline includes the sequential activities of image acquisition, segmentation, feature selection, feature extraction, modelling and evaluation.

2.2.1 Step 1: Image Acquisition

Radiomics is the process of quantifying the characteristics of medical images [31]. It can be applied to different modalities of digital imaging [32]. There is no standardised image acquisition technology to use in a radiomics study [31].

The most common medical imaging protocols are Computed Tomography (CT) Scans, Positron Emission Tomography (PET) Scans, and Magnetic Resonance Imaging (MRI).

- **PET Scan:** This technique uses radioactive substances to scan the reaction of the organs and tissues to the drug. Utilising PET scans in combination with CT scans or MRI scans can lead to better disease diagnosis [33].
- **CT Scan:** Multiple X-ray images are captured from various angles around the body and combined by a computer algorithm to constitute cross-sectional images of the region inside the body [34].

- **MRI:** This screening technology uses a magnetic field and computerised radio waves. It produces high-resolution images of part of the body [35]. MRI has become an advanced technology that provides a non-invasive analysis of pathology [36].

The imaging protocols between sites and studies are usually not standardised. Also, the devices and scanners used for image acquisition may introduce noise that will affect the radiomics pipline [28]. Hence, in radiomics studies, the raw images are revised by pre-processing procedures, such as noise reduction, artefacts correction, normalisation and so forth [36].

2.2.2 Step 2: Segmentation

Lesion segmentation is a critical step in a radiomics study as the image delineation affects the quality of features extracted from the corresponding region of interest (ROI) [14], [17], [26], [36].

Segmentation can be done in manual, semi-automated or automated ways [14], [31], [32], [36].

- **Manual method:** an expert or group of experts annotate the boundaries of the lesion region [36], [37]. Manual segmentation is vital in the studies as a high degree of lesion border accuracy is necessary [31].
- **Semi-automated algorithms**: refers to the usage of standard segmentation techniques such as thresholding or region-growing. These methods usually use manual correction [32].
- **Automated solutions:** Nowadays, several open-source or commercial software and tools for lesion segmentation are available [14], [32].

2.2.3 Step 3: Feature Extraction

Feature extraction is at the heart of the radiomics pipeline. In this step, the images are converted to mineable data. The different types of biomarkers that can be extracted are categorised into three main groups as follows:

- **Shape features:** are the most direct attributes related to the geometry of the ROI, such as volume, sphericity or compacity [27], [38].
- **First-order features:** refer to the statistical distribution of the voxel intensity values within the segmented region, and include measures like the mean, median, uniformity, randomness, skewness, kurtosis [5], [28], [38].
- **Second-order features:** generally, are described as texture features. This type of features is statistical descriptors related to spatial relationships between

voxels. There are many texture features. Examples of texture features are Local Binary Pattern (LBP), the Grey Level Co-Occurrence Matrix (GLCM), the Grey Level Run Length Matrix (GLRLM), the Neighbouring Grey Tone Difference Matrix (NGTDM) and so forth [26], [38], [39].

- **Higher-order features:** are determined by applying filters and advanced methods to the images to extract patterns difficult to distinguish by eye, such as Laplacian of Gaussian filter, Fourier transform, and Gabor transform [27], [38], [39].

In the rest of this section, the different texture features used in our study will be described.

Three-Dimensional Local Binary Pattern (3D LBP)

LBP is categorised as a texture feature. The basic concept of LBP was introduced by Ojala et al. (1996) [40]. After that, many extensions to it have been proposed. Some studies proposed an extension of LBP to capture 3D textures and patterns for 3D images. In our study, we used the approach presented by Montagne et al. (2013) [41].

LBP Basic Process

The LBP basic calculation for both 2D and 3D space is the same. For calculating LBP code for each pixel/voxel in our image, the steps are as follows [41]–[44]:

1) Calculate the difference between the intensity value of the central cell (g_c) and its neighbours *(g_i),* denoted as ($g_i - g_c$)
2) Provide a sign function ($s(x)$) that means if the neighbour cell has an intensity value greater or equal to the central voxel, it is set to 1 else 0. By concatenating obtained zero or one values, we have a binary code of length *P* for each centre point.
3) Convert the binary code to a base-ten decimal number by the LBP operator (Equation 1). Each decimal number represents a unique textured pattern.

$$LBP_{P,R} = \sum_{i=0}^{P-1} s(g_i - g_c)2^i \quad where \quad s(x) = \begin{cases} 1 & if\ x \geq 0\ , \\ 0 & otherwise \end{cases} \tag{1}$$

P is the number of neighbouring voxels at a distance *R* from the central node.

4) Derive the LBP features by counting the frequency of each decimal output (histograms of patterns). This last step is necessary when we use the LBP operator for extracting texture features.

Figure 2 outlines the process of acquiring the decimal number in a 2D LBP.

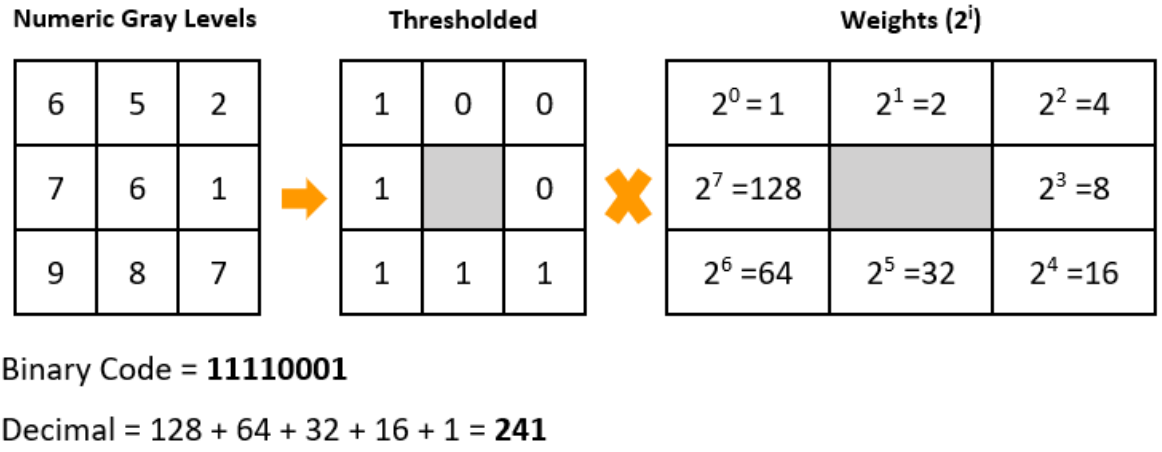
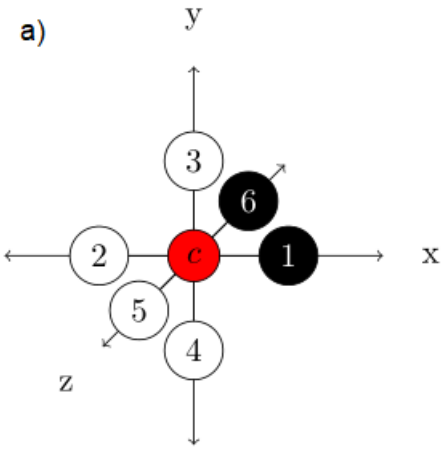


Figure 2. Example of a 2D LBP computation, P = 8 and R = 1 [42]. The intensity value of central cell (6) works as a threshold for assigning zero or one to the eight adjacent cells and making a binary code. i is the index of the neighbouring pixels in a clockwise direction. The decimal is the base-ten format of the binary code.

3D LBP

According to Montagne et al. [41], in 3D LBP, we only considered the direct neighbours on the axes *x, y, z* and excluded the neighbouring voxels on diagonals in 3D space. The direct neighbours are surrounding voxels with *R* equal to 1 (Figure 3).



b)

axis	i	coordinate
x	1	(1 0 0)
	2	(-1 0 0)
y	3	(0 1 0)
	4	(0 -1 0)
z	5	(0 0 1)
	6	(0 0 -1)

Figure 3. Direct neighbours of a specific cell in 3D space on the x, y, z axes. a) The spatial schematic of the central node and its adjacent cells. White nodes are 0 (lower intensity value than c), and black nodes indicate 1 (higher or equal to c). The number on each circle shows the indexing order of nodes (i). c (the intensity value of central voxel) denotes the threshold value. b) The enumeration order and the position of direct neighbours on each axis with the central node as the origin point [41].

The LBP operator output (Equation 1) produces 2^P different values, referring to 2^P binary patterns (each neighbouring voxel can only have the value 0 or 1 in LBP

computation). Hence, if the number of adjacent cells equals 6, we have 2^6 (= 64) patterns [45].

Rotation Invariance

It is important to recognise unique patterns from redundant ones because this increases interpretability and decreases computation time. According to [42], rotation of the image results in a different interpretation of LBP pattern location (Figure 4).

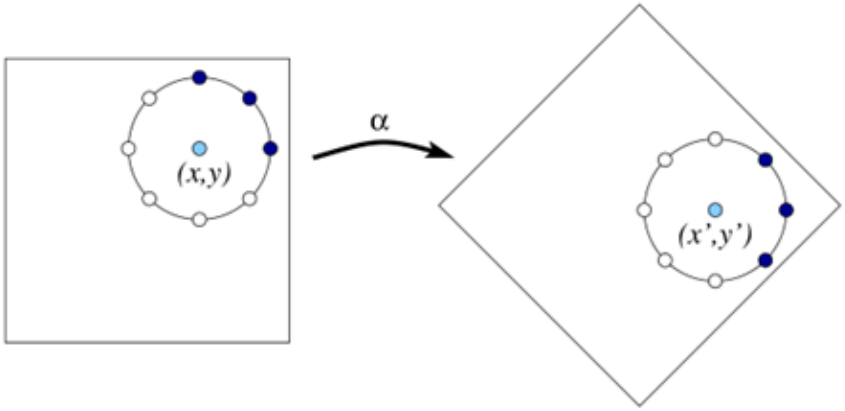


Figure 4. The effect of rotation on the neighbourhood points [42].

The rotation-invariant concept states that some patterns counted as a new pattern are generated repeatedly by rotating the image. Actually, these patterns occur by displacement of the neighbouring cells along the perimeter of the circle (if we assume a circle around the central voxel with the neighbouring cells on its perimeter) [45]. Grouping the patterns of similar scenarios enables us to remove the effect of image rotation.

According to [41], three distinct scenarios can occur at each axis by considering each coordinate (*x*, *y*, *z*) separately. These scenarios are:

I. Both neighbouring nodes on the axis are lower than the centre voxel (zero value)
II. One of the adjacents is lower than the centre value; the other is higher than (or equals) the centre node.
III. Both adjacent cells on one axis are higher than or equal to the centre point (one value)

We used the above scenarios to remove redundant patterns and reduce the number of patterns to 10 distinct patterns instead of 64 (2^6). Figure 5 demonstrates these patterns. The mentioned scenarios were also used to name the different patterns. For example, LBP_300 shows the pattern has only zero points on all three axes (pattern 1 in Figure 5), whilst LBP_003 represents the pattern with all nodes equal to 1 on all three coordinates (pattern 10 in Figure 5). By contrast, the LBP_030 is the pattern in

which there exists one node of 1 and one node of 0 on each axis (pattern 6 in Figure 5). Table 1 provides the number of times each pattern can arise and the pattern name.

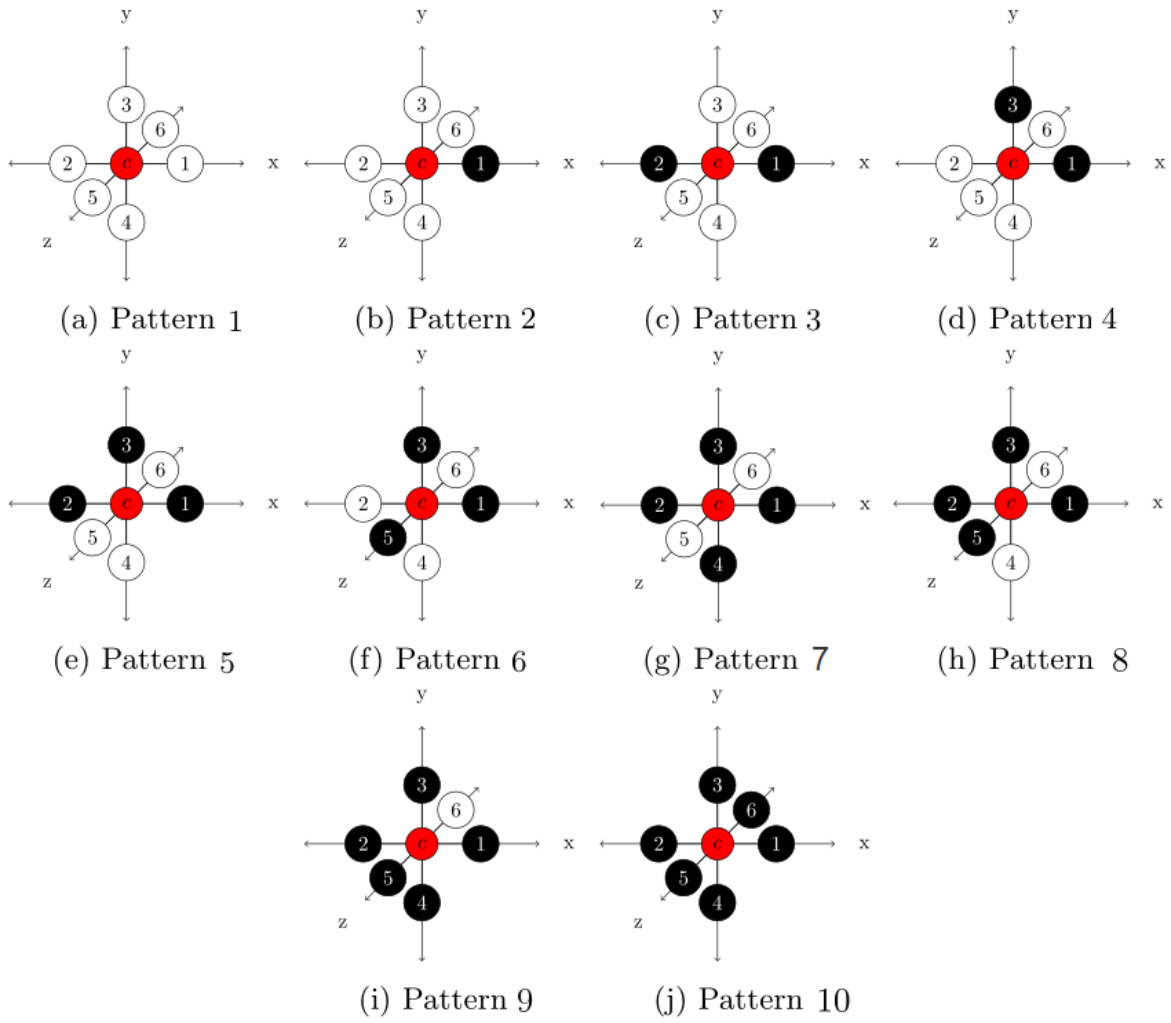


Figure 5. All possible groups of rotation invariant patterns in a 6-neighbour 3D LBP. The intensity value of the central voxel (c) is the threshold value. White nodes are cells with a lower intensity value than c (0), and black nodes indicate cells with a value higher or equal to c (1). The number on each circle shows the indexing order of nodes (i) [41].

Table 1. Name and frequency of different arrangements of 6-neighbour 3D LBP patterns depicted in Figure 5.

Pattern Number	Name	Multiplicities	Pattern Number	Name	Multiplicities
Pattern 1	LBP_300	1	Pattern 6	LBP_030	8
Pattern 2	LBP_210	6	Pattern 7	LBP_102	3
Pattern 3	LBP_201	3	Pattern 8	LBP_021	12
Pattern 4	LBP_120	12	Pattern 9	LBP_012	6
Pattern 5	LBP_111	12	Pattern 10	LBP_003	1

Grey Level Co-occurrence Matrix (GLCM)

The texture representation technique of the co-occurrence matrix is one of the oldest statistical methods. This method determines the texture by means of grey level distribution [46]. The value of $(i,j)^{th}$ element of the matrix is the count of voxels with grey level j that exists in a distance of d along the direction θ from a voxel with the value of i [21], [46]. Figure 6 shows an example of GLCM calculation.

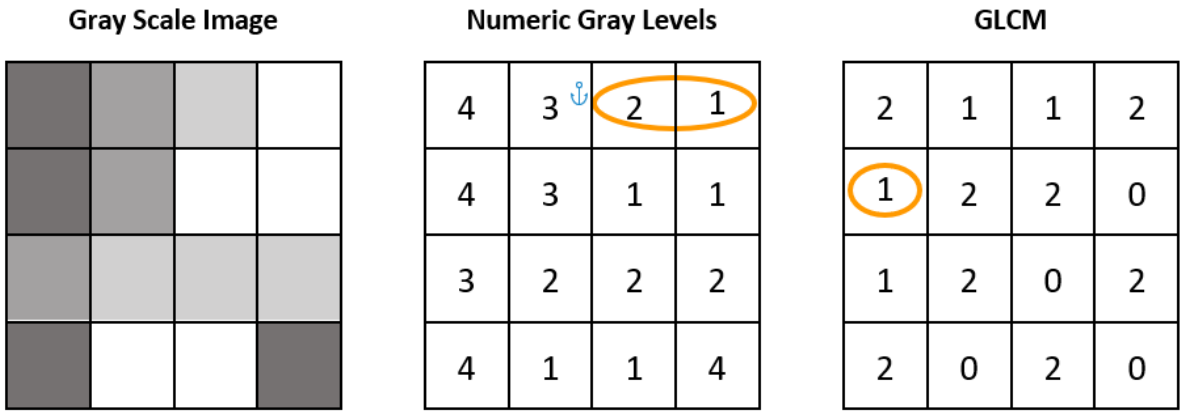


Figure 6. An example of GLCM computation with d = 1 and θ = 0. The element (2,1) of the GLC matrix equals 1 since only one combination of connecting voxels with intensity values of 2 and 1 in the horizontal direction exists. modified from [13], [47].

According to [21], the features such as autocorrelation, difference entropy, contrast and so forth are calculated for the GLC matrix.

Grey Level Dependence Matrix (GLDM)

The GLD matrix presents the dependencies in a grey scale image [21]. The calculation of GLDM is illustrated in Figure 7. The element (i,j) of the output matrix is the number of times that the centre voxel with grey level i has j dependent neighbour voxels. The centre voxel with intensity level i is dependent on its neighbour cell with grey level γ at the distance d if $|i-\gamma| \leq \alpha$ (α is a given scalar) [13], [21].

One can calculate Gray Level Variance, Dependence Non-Uniformity, Large Dependence Emphasis and so forth for the GLD matrix [21].

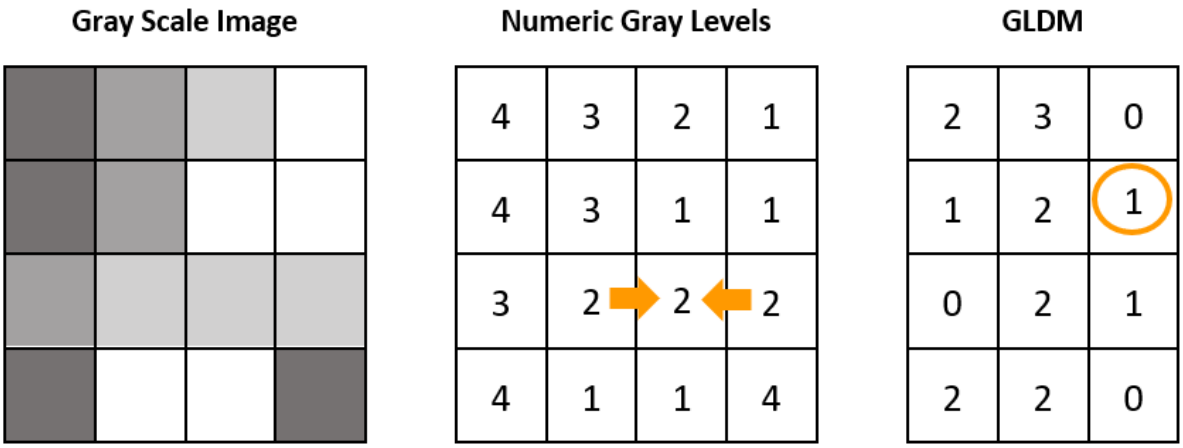


Figure 7. An example of GLDM computation of an image with four grey levels. Here, the distance d equals 1, and the scalar α is 0. The lineated element of the GLD matrix equals 1 since there exists only one centre voxel with value 2 and two dependencies. Modified from [13].

Grey Level Run Length Matrix (GLRLM)

GLRLM is another statistical texture method. This method constructs the output GLRL matrix by calculating the count of cells with the same grey level in a specific direction of α [46]. For instance, two voxels with the same intensity value in the horizontal direction provide one run with length two [47]. In Figure 8, the calculation of a GLRL matrix is shown.

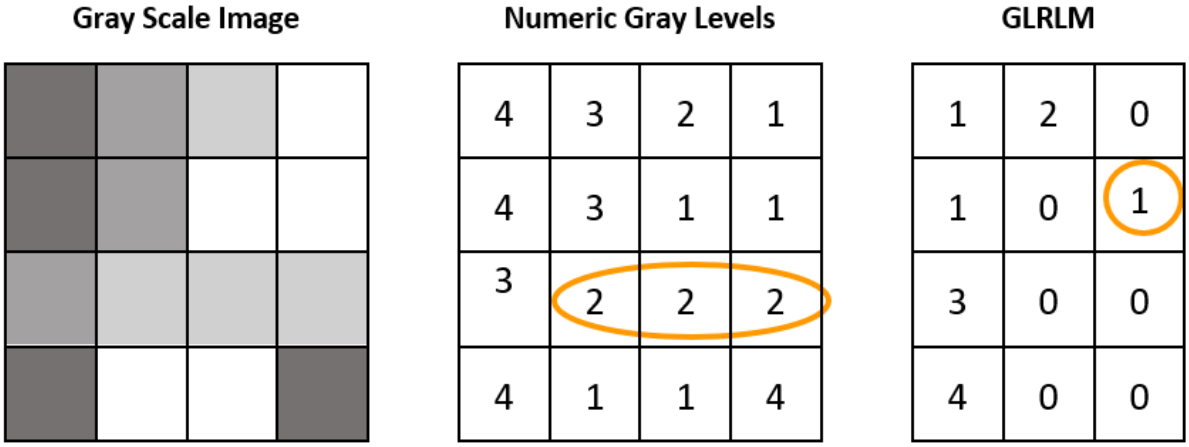


Figure 8. An example of GLRLM computation of an image with four grey levels in the direction of α = 0. The element (2, 3) of the GLRL matrix equals 1 since only one run with the grey level 2 and length 3 in the horizontal direction exists. Modified from [13], [47] .

Several features such as Short Run Emphasis, Run Percentage, Run Length Non-Uniformity and so forth are calculated for the GLRL matrix [21].

Grey Level Size Zone Matrix (GLSZM)

The grey level size zone matrix considers the area with the cells of the same grey level. The basis of extracting GLSZM features is like GLRLM construction. The matrix $(i, j)^{th}$ element shows the number of zones with the size j and intensity value i (Figure 9). This matrix manifests a homogeneous texture if the matrix elements show large areas of the same grey level toward any direction [46].

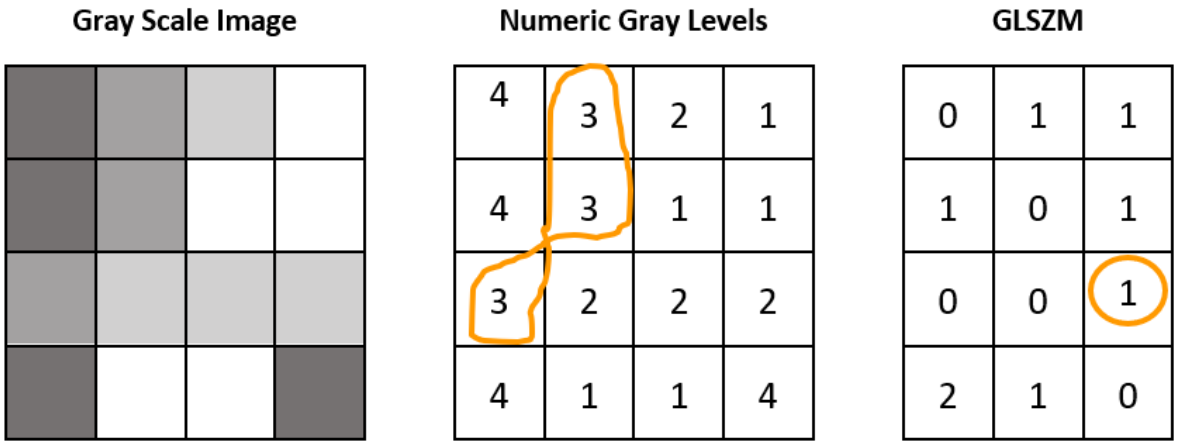


Figure 9. An example of GLSZM computation of an image with four grey levels. The element (3, 3) of the GLSZ matrix equals 1 since only one zone has grey level 3 and size three. Modified from [13].

For the GLSZ matrix, various features such as Zone Percentage, Large Area Emphasis, Size-Zone Non-Uniformity and so forth are calculated [21].

Neighbouring Grey Tone Difference Matrix (NGTDM)

The NGTDM provides the difference between the grey level of a voxel and the average intensity values of its neighbouring voxels at a distance d. The matrix consists of the sum of absolute differences for intensity value i [21]. Figure 10 illustrates the computation of NGTDM.

The features such as coarseness, contrast, busyness, complexity, and strength are calculated for the NGTD matrix [21].

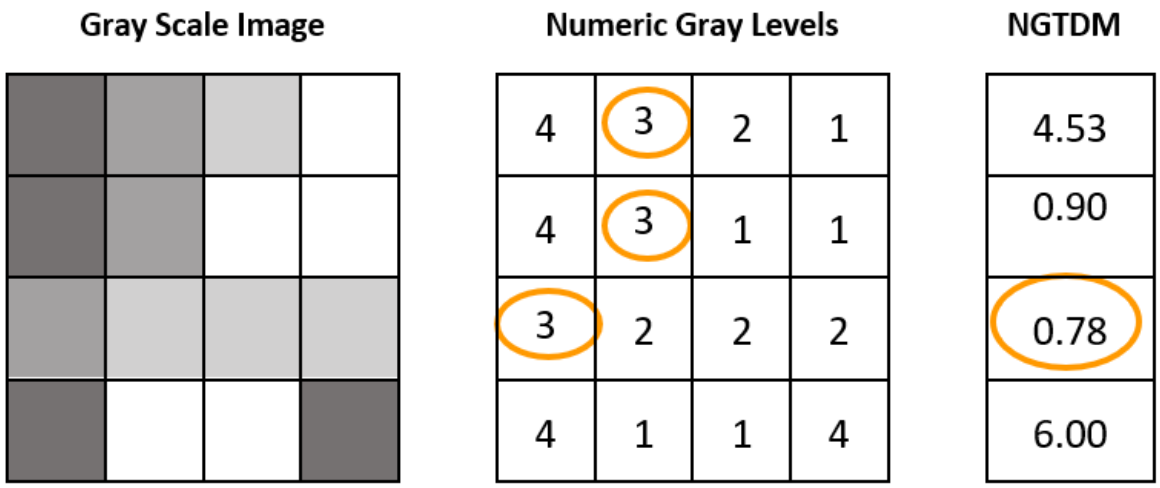


Figure 10. An example of a 2D NGTD matrix calculation. s_i indicates the sum of the absolute difference between grey level i and its adjacent cells. The figure shows the computation of s_3 ($i = 3,\ d = 1$) $s_3 = \left|3 - \frac{4+4+1+2+3}{5}\right| + \left|3 - \frac{4+4+3+2+2+1+2+3}{8}\right| + \left|3 - \frac{4+4+3+1+2}{5}\right| = 0.78$. modified from [13].

Although LBP is a texture feature, from here on, the term "texture feature" refers to the second-order features including GLCM, GLDM, GLRLM, GLSZM and NGTDM, excluding LBP. Whenever we aimed at referring to the LBP feature, we mentioned its name directly.

2.2.4 Step 4: Feature Selection

According to [48], feature selection is crucial for problems with short-wide datasets for three reasons: 1) to tackle the “curse of dimensionality”; 2) to compact the input data for reducing model execution time; 3) to improve the result comprehensibility.

Thus, feature selection is a critical step in the radiomics pipeline because plenty of features are obtained during the feature extraction step. In addition, due to the limitations for gathering samples in clinical studies, the dataset has a small number of samples compared to plenty of features. Therefore, in this context, datasets are short-wide (few samples with many features). Moreover, the radiomics features are highly correlated, redundant, or irrelevant, affecting model performance [17], [22].

Feature selection focuses on searching for a subset of the input data with fewer features that can represent the given dataset effectively and improve the learning accuracy by decreasing the side effects of noise or redundant features [49], [50]. In past years, several methods have been proposed for answering the need for selecting features. In general, the feature selection methods are categorised as filter methods, wrapper methods and hybrid methods.

The filter approach refers to the algorithms for selecting features without training by any predictive model [48]. On the other hand, wrapper methods rely on learning by a predefined model and using its performance as the criteria to select features [51]. In contrast, the hybrid methods are a combination of both filter and wrapper methods [52].

Repeated Elastic Net Technique (RENT)

RENT [23] is a brand-new feature selection method introduced by the Norwegian University of Life Sciences. It is an ensemble based approach and well designed for short–wide datasets [23]. It tries to select robust features by employing logistic regression (LR) model with elastic net regularisation for binary classification tasks.

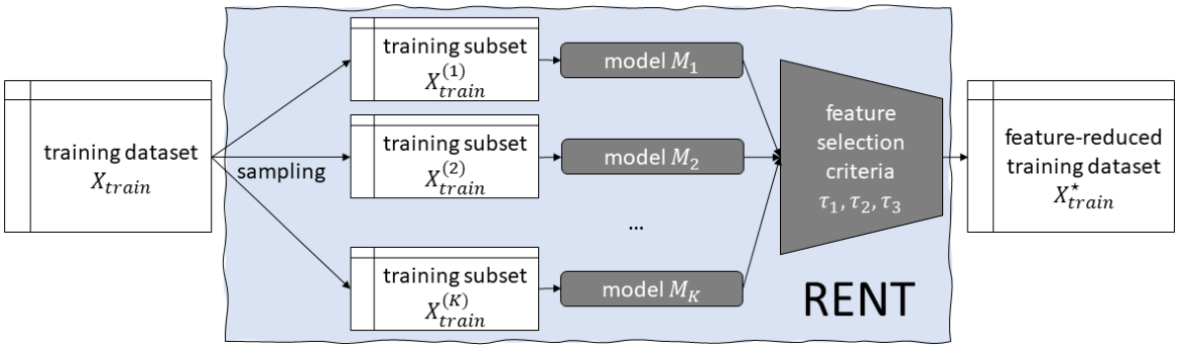


Figure 11. The blue frame demonstrates the RENT process. RENT splits and trains the input dataset across the K sub-models and selects the robust features based on three criteria (τ_1, τ_2, τ_3). The output is a dataset with the selected features [23].

According to [23], in a binary classification problem, RENT first splits the input dataset into several unique subset models (Figure 11). Then, it uses the penalised LR algorithm to train each subset model M_i separately. In each sub-model, a different subset of features may be selected by the elastic net regularisation. Finally, based on the quality criteria and the user given cut-off values. A feature will be added to the output dataset if it fulfils all of the following criteria together [23]:

1. The feature has a high score, which means that it is selected in most of the K models (τ_1). A user-defined threshold (t_1) determines how many times the feature should be selected among all *K* models.
2. The feature is stable if it has few weights' signs alternation (τ_2). A feature with weights of the same sign (either all positive or all negative) is ideal. The user can provide the preferred number of proportions of feature weights with the same sign (t_2).

3. The feature frequently has non-zero weights across the *K* sub-models with low variance (τ_3). User can specify a threshold value (t_3) for the level of significance.

All the quality metrics are bounded between 0 and 1 ($\tau_1, \tau_2, \tau_3 \in [0, 1]$) [23]. It has to be stressed that a feature is selected if and only if $\tau_1 \geq t_1 \; and \; \tau_2 \geq t_2 \; and \; \tau_3 \geq t_3$.

The possibility of defining three threshold values (t_1 , t_2 , t_3), instead of specifying the desired number of features, makes the user capable of adjusting the strictness of the feature selection process.

Elastic Net

Elastic net is a regularisation method introduced by Zou and Hastie (2005) [53]. Equation 2 shows the elastic net regularisation term (λ_{enet}) calculation [23].

$$\lambda_{enet}(\beta) = \gamma[\alpha \, \lambda_1(\beta) + (1 - \alpha) \, \lambda_2(\beta)] \quad (2)$$

The penalty parameter λ_1 (named L1 regularisation) penalise the sum of absolute values of β (regression coefficients), and the penalty parameter λ_2 (named L2 regularisation) penalises the sum of squared values of β [23], [54].

In equation 2, γ denotes the regularisation strength and is a positive decimal number. α is a mixing parameter in the range of [0,1] [23]. RENT uses the LR classifier implemented in scikit-learn [55]. In this package, the α parameter is indicated by the l1_ratio parameter, and instead of γ parameter, the inverse of γ named C parameter is used [55].

Compared to other regularisation models (Lasso and Ridge), the elastic net advantage is in exploiting both λ_1 and λ_2 penalty parameters that empower the algorithm to combine shrinkage and the variable selection [54].

In [23], RENT is compared to various feature selection methods by performing several empirical experiments, and Fisher score (F- Score) and recursive feature elimination (RFE) methods provided competitive results. Thus, we chose these two feature selection methods to describe as examples of filter methods (F- Score) and wrapper methods (RFE).

Fisher Score Method

The Fisher score (F-Score) method selects features by measuring the class discriminant of each feature based on its F-Score value [56]. It is a filter-based method,

which means that features scores are calculated, and the features are selected in terms of their score ranks [57].

The idea behind this method is to provide a subset of features with larger distances between samples in a different class and a smaller distance between data points of an individual class [57]. The final selected feature subset contained features with a higher F-Score [56]. Equation 3 is used for calculating F-Score [58].

$$F(x^j) = \frac{\sum_{k=1}^{c} n_k \left(\mu_k^j - \mu^j\right)^2}{(\sigma^j)^2} \quad (3)$$

Where:
x^j is the *j*-th feature
μ_k^j is the mean of the jth feature in the *k*-th class
σ_k^j is the standard deviation of the *j*-th feature in the *k*-th class c
μ^j is the mean of the *j*-th feature for the whole dataset
σ^j is the standard deviation of the j-th feature in the whole dataset

Although the features selected by the F-Score algorithm are often suboptimal, this heuristic algorithm has some deficiencies. The F-Score method fails in cases where features have low individual score and a very high score when considering together as a whole. Another drawback is related to its weakness in handling redundant features [57].

Recursive Feature Elimination Method

The recursive feature elimination (RFE) method is a simple and popular wrapper method. RFE uses various ML algorithms as the core training method [59].

The algorithm starts by fitting an ML model on the given dataset. Then it continues by eliminating the least important features or features with lower weight coefficients. This process is repeated until the desired number of features is reached [59], [60].

Even though this method is quick and straightforward, it is not appropriate for problems with plenty of highly correlated features [61].

2.2.5 Step 5: Modelling and Evaluation

Machine learning is a subfield of the artificial intelligent area and has evolved remarkably fast [15]. Machine learning shows its unique capabilities in research areas. It plays an essential role as an interface between medical research and computer

science studies [22]. The analysis of image data through machine learning concepts can empower us to understand illnesses and medications, and it can provide effective treatments and personalised medication [62].

Model Building

In radiomics studies, the objective is to exploit machine learning concepts to predict the target based on radiomics features [14]. Machine learning algorithms are categorised into two main groups:

- ***Supervised learning*** uses labelled samples as the target variable to predict the output. The target can have continuous values in a regression model or a categoric value in a classification model [15], [63].
- ***Unsupervised learning*** does not use expert labelled data. Instead, it tries to find the patterns in the data by methods such as clustering and predict the new data structure [62], [63].

Supervised Classifiers

As mentioned before, classification problems belong to the family of supervised learning methods. With regards to the number of class labels, the classification tasks can be binary or multi-class problems. There are only two class labels in binary classification; by contrast, multi-class tasks have more than two class labels. The current study is a binary classification work.

There are a variety of classifiers used in supervised learning for binary classification. In this research, we used the following classifiers:

- ***Logistic Regression (LR) Classifier*** is an easy-to-implement algorithm. It is broadly used in medical studies because it is appropriate for defining the disease state [64]. Despite its name, it is a binary classifier that forecasts the target value using the logistic function [63]. It is not necessary to have normally distributed predictors or linearly related ones, but these will increase the model power. It assumes a linear relationship between the logit of the dependent variable (outcome) and the independent variable (predictor) [64].
- ***Support Vector Machines Classifier (SVC)*** It is an effective machine learning method. Its objective is to maximise the distance between decision boundaries and the samples [55].
- ***K Nearest Neighbors (KNN) Classifier*** seeks the given number of samples (*k*) near the desire training example based on a distance metric and provides the class label of the desired sample by majority voting [55].

- ***Multi-Layer Perceptron (MLP) Classifier*** is a basic neural network algorithm. It has multiple nodes and layers (similar to a direct graph). The layers are the input layer, the hidden layer(s) and the output layer. All nodes in one layer are connected to the nodes in the preceding layer [65].
- ***Decision Tree (DT) Classifier*** is the basic tree classifier that is based on rules. It groups the samples based on rules and decision making [55].
- ***Random Forest (RF) Classifier*** refers to the algorithm that models an ensemble of decision tree sub-models and provides the output based on the majority class label in all sub-models [66]. It uses a random bootstrap sample size [55].
- ***Ridge Classifier*** corresponds to an L2 regularised model [55], [67]. This algorithm moderates the weight coefficients by minimising the sum of squared residuals [68].
- ***Adaptive Boosting (AdaBoost) Classifier*** is an ensemble algorithm. It trains many weak learners by generating a sequence of classifiers and reweighting the importance of samples to find the best classifier. Larger weights are assigned to misclassified samples until the algorithm attains a model that can classify them correctly [67].
- ***Extremely Randomised Tree Classifier*** It is also an ensemble model based on decision trees. Its difference from the random forest is that it uses the entire sample instead of bootstrapping. Also, it randomly chooses the cut-points for splitting the nodes. Similarly to other ensemble models for final prediction, it uses a majority voting [69].
- ***Light Gradient Boosting Machine (LGBM) Classifier*** is a method that uses a gradient boosting decision tree procedure. It uses histogram-based concepts which convert continuous values into discrete groups (bins) [70].

Hyper Parameter Tuning

Hyperparameters correspond to any parameter of the ML algorithm set before model training starts [71]. For instance, in an ANN model, the batch size or the number of layers are hyperparameters because they are fixed before training begins; in contrast, the weights are not hyperparameters since their values are assigned during the training process [71], [72]. Because hyperparameters control the training process directly, they impact model performance significantly [72]. Simple ML algorithms do not have any hyperparameters; conversely, some others require plenty of hyperparameters to be set beforehand; in some cases, the hyperparameters are related to each other [73].

Hyperparameter tuning refers to the process of finding the combinations of hyperparameters that lead to the highest performance [72]. There are a wide variety of automatic tuning methods. Grid Search is a popular hyperparameter search technique that finds the best combination of given hyperparameters by checking the different combination of algorithm parameters from a predetermined parameters grid [72], [73]. Despite being simple, it is time-consuming when the dataset is large, and the parameter grid contained many alternatives [73]. Since the dataset used in this study is very short, we used this method for hyperparameter optimisation.

Model Validation

The model performance should be evaluated on unseen data, ideally data from other institutions [74], [75]. Due to patient privacy policies, gathering many medical images as samples is difficult, and medical datasets can suffer from small samples availability [17].

If independent data is not available, it is possible to split the data into train and validation groups. In this way, the algorithm can learn from the train set and predict the output based on the validation set, which is untouched during the learning process. However, when the dataset contains few samples, the splitting approach does not work properly due to insufficient train and validation data. In this situation, cross-validation techniques are utilised for increasing the model's generalizability [74], [75]. Among the various types of cross-validation methods, the nested cross-validation technique is useful when the model is prone to overfitting (such as small dataset issue) and whenever there is a need for hyperparameter tunning [55], [74].

Nested Cross-Validation

The cross-validation techniques assess the model's generalizability by dividing the data into training and validation sets [63]. In nested cross-validation, instead of a single layer, there are multiple layers, generally, two layers of cross-validation inner loop and outer loop [76].

Figure 12 shows a 5×3 nested cross-validation (five folds in the outer layer and three folds in the inner layer). A 5×3 nested cross-validation splits the data into five folds in the outer loop; four folds are the train set and one fold as a validation set playing the role of unseen data. The train set is again split into three folds in the inner loop, two folds for training and one fold as a test set. The model execution is repeated by changing the folds until all the folds are used as train and validation sets in the outer loop and train and test sets in the inner loop. The hyperparameter tuning is done in the inner loop. In the outer loop, the best hyperparameter set (obtained from the inner loop) is used for making the final prediction on the validation set [63], [75], [76].

The added outer loop removes the bias in the flat cross-validation method since the validation data has not been used to select the optimal model. This process gives us a more reliable model than the basic cross-validation form [77].

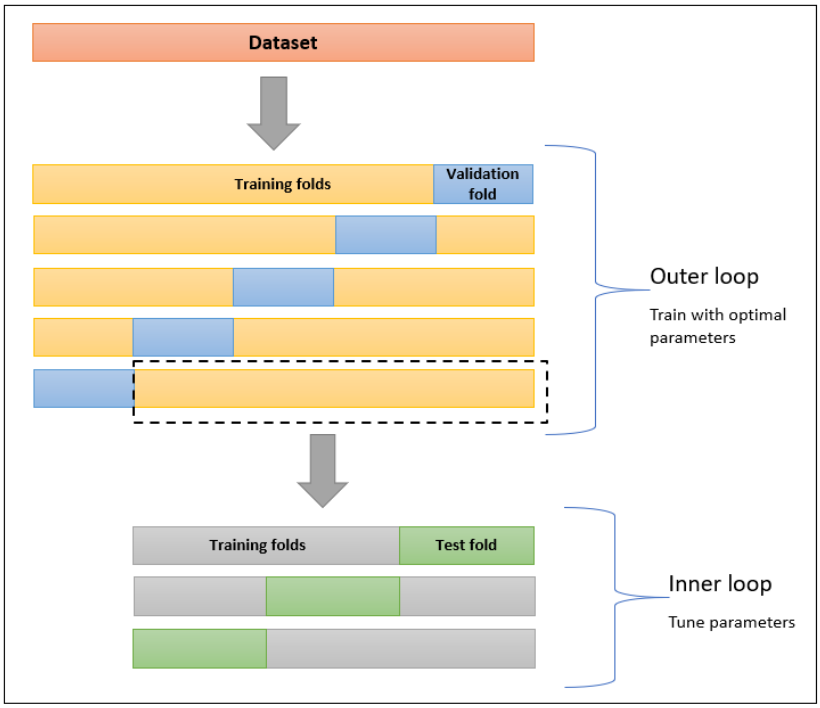


Figure 12. A 5×3 nested cross-validation (five folds in the outer loop and three folds in the inner loop).

Model Evaluation

There are a variety of metrics for evaluating the performance of classification models. In medical studies, it is vital to differentiate between false positive (FP) and false negative (FN) misclassification [17], and the metrics used must take this into account. In classification prediction, true positive (TP) and true negative (TN) refers to the situation that a sample is classified correctly. In contrast, false positive (FP) and false negative (FN) correspond to misclassification cases. Various metrics are calculated using FP and FN concepts; among them, the area under the receiver operating curve (AUC) is a common metric proper for a balanced dataset [74].

Area Under Curve

According to [63], for computing AUC, the first step is to plot the receiver operating curve (ROC) based on true positive rate (TPR) and false positive rate (FPR), then calculate the area under this curve. Equation 4 shows the TPR and FPR computation [63].

$$TPR = \frac{TP}{TP + FN}, \qquad FPR = \frac{FP}{FP + TN} \tag{4}$$

Figure 13 illustrates the receiver operating curve. It is clear from the figure that the AUC ranges from 0.0 (no correct classifications) to 1.0 (no incorrect classifications). In this plot, the AUC of 0.5 shows the random classification rates.

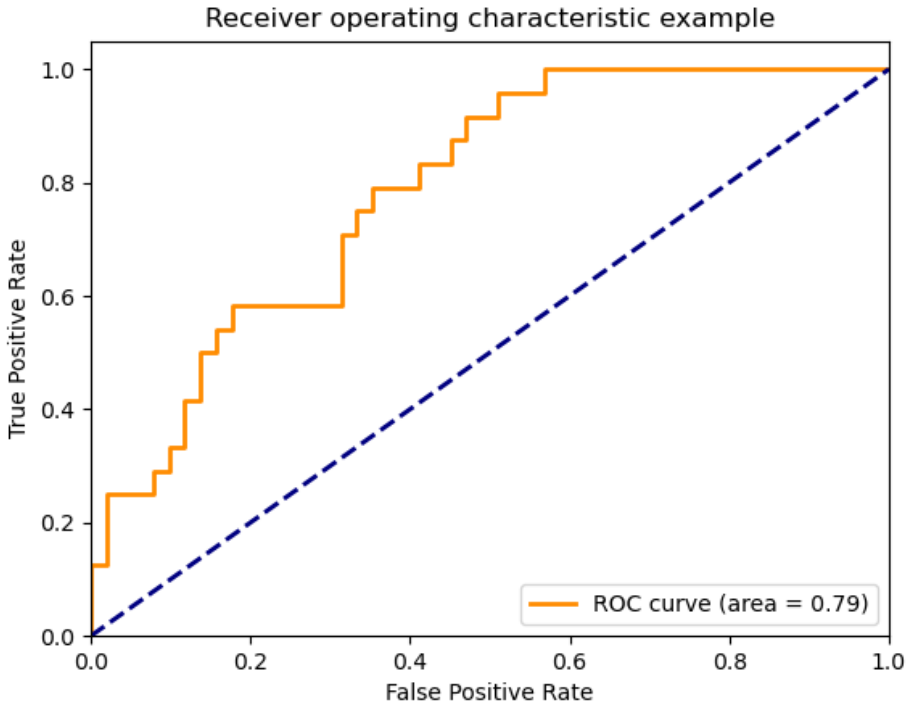


Figure 13. An example of the Receiver Operating Curve with the Area Under Curve of 0.79. The blue dashed line shows the random guess line [55].

3 Materials and Methods

3.1 Image Acquisition and Segmentation

3.1.1 The ePOD-MPH Study

In this study, images from the ePOD-MPH study [11] were considered. In 2011, a randomised double-blinded project named “the effects of Psychotropic drugs On Developing brain (ePOD)" was conducted. This research was designed to be placebo-controlled. The Clinical Research Unit of the Academic Medical Center at the University of Amsterdam in the Netherlands was the responsible authority for this project. The ePOD-MPH study was one of the three categories in the ePOD project.

In the ePOD-MPH study, the participants were randomised to receive methylphenidate or the placebo for 16 weeks. After that, a week wash-out period was conducted. The MRIs were taken before the beginning of treatment (baseline), during the treatment (after the eight weeks), and after the wash-out period (17-week). In our thesis, we referred to the baseline images as pre-treatment images and the 17-week MRIs as post-treatment images.

In the ePOD-MPH study, 100 male ADHD patients were involved in the experiment. There was an equal number of children (10- to 12-year-old) and adults (23- to 40-year-old) among the participants. This master study is limited to examining the MRIs obtained from male adolescents. Four subjects were excluded from the current study regarding the exclusion rule: 1) The trials with no baseline or 17-week MRI led to image exclusion. 2) In addition, the images disrupted by head motion were removed from this study. Thus, 46 samples were included for analysis containing 24 placebo-treated patients and 22 MPH-treated trials, giving a balanced dataset.

Figure 14 shows the distribution of class labels. The trials were labelled in terms of the medication group. The cases in the MPH group were labelled as class 1 (samples' ID from 0 to 21) versus subjects in the placebo group labelled as class 0 (samples' ID from 22 to 45).

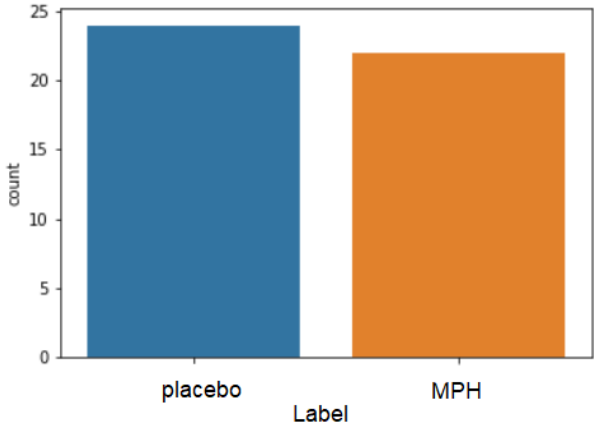


Figure 14. The distribution of class labels. Class 0 indicates the placebo group, while class 1 denotes the MPH treated group.

3.1.2 Image Segmentation

The images and masks used in this thesis were obtained from the study by Grünbeck [13]. In her study, raw T1-weighted MR images were used, and five subcortical brain structures named hippocampus, caudate, putamen, thalamus, and pallidum were selected for analysis. Grünbeck [13] created binary masks for the left and right side of the mentioned subcortical structures separately.

3.2 Feature Extraction

The feature extraction steps in our study contained two separate phases. The first one was related to the extracted shape and texture features done by Grünbeck [13], and the second one was related to the LBP features, which were extracted by a new programme developed in this thesis. We also modified the Biorad application [19] to add our new 3D LBP feature extraction module. This framework was developed by Langberg [18] and upgraded into a user-friendly tool for extracting radiomics features by Albuni [19].

In this section, we first explain generating shape and texture features as done by Grünbeck; then, we elaborated the LBP module and modifications of Biorad to adapt this new LBP module.

3.2.1 Shape and Texture Features Extraction

Grünbeck [13] used Biorad [19] for generating radiomics features. The Biorad framework [18], [19] uses pyradiomics [20], an open-source python package for generating radiomics features. This package aims to provide a reference for radiomics studies and introduce an easy tool for extracting reproducible radiomics features.

Grünbeck used the default parameters of pyradiomics for generating radiomics features. This means that for GLCM, GLDM and NGTDM, the distance between voxels was set to 1, and the threshold scalar of dependence in GLDM was set to zero.

Before extracting texture features, Grünbeck discretised the images' intensity by using bin sizes of two and four to reduce the intensity level range of images from 256 to 128 and 64 intensity levels, respectively. This process generates two distinct feature sets named 128-bin and 64-bin sets. Furthermore, the features were extracted from the left and the right side of each subcortical brain structure. The number of radiomics features extracted by Grünbeck for one side of one of the subcortical brain structures is shown in Table 2.

Table 2. The number of radiomics features extracted from the images for one subcortical structure on one side of the brain.

Shape (3D)	Texture									
	GLCM		GLDM		GLRLM		GLSZM		NGTDM	
	128-bin	64-bin	128-bin	64-bin	128-bin	64-bin	128-bin	64-bin	128-bin	64-bin
14	24	24	14	14	16	16	16	16	5	5

3.2.2 3D LBP Features Extraction

As there is no tool in the python language (at the date of writing this thesis) for generating 3D LBP features, we developed a 3D LBP feature extraction module and integrated it into the Biorad framework [19]. Thus, now it is possible to extract LBP features in addition to pyradiomics features [21] by using the Biorad framework [19]. The code for extracting 3D LBP features is available in Appendix A. We used NiBabel, an open-source python package that supports standard neuroimaging file formats, for converting the images and binary masks into arrays [78].

According to chapter 2, only direct neighbours were considered (6 neighbours located on x, y, z axes).

The steps of extracting 3D LBP features are as follows:

1. Read the image and corresponding mask

2. Convert the image and mask into arrays
3. Calculate the LBP value (considering direct neighbours, which means P=6 and R=1) for the voxels in the binary mask area (as mentioned in chapter 2)
4. Map the LBP value to the corresponding rotation invariant pattern (based on the rotation_invariant_pattern table in Appendix B)
5. Calculate the frequency of patterns
6. Compute the following fraction: $\frac{frequency\ of\ one\ pattern}{Total\ number\ of\ frequencies}$

Ten LBP features were extracted for each side of the subcortical structure of the brain.

Modifications of Biorad Feature Extraction Module

After developing the feature extraction programme for extracting 3D LBP features from medical images, we upgraded the last version of the Biorad framework from the Albuni study [19] to make it compatible with generating 3D LBP features. The modified code of Biorad is available in Appendix C. The list of modifications of the Biorad *feature extraction module* is as follows:

1. Imported the 3D LBP module.
2. Added the “LBP” to its feature list.
3. Added our code to the functions named “extract_radiomics_features“ and “get_selected_features”.
4. Added the LBP column to the template.csv file.
5. Modified the requirement.txt file of the Biorad framework to install our necessary python packages (NiBabel, Collections, Six, Pandas and Numpy). For the required packages of Biorad, see [19].

The input to the feature extraction module of Biorad is a CSV file (named template.csv) containing the location of the images and masks files and the output file location. Also, the user should choose the desired radiomics features to be extracted by inserting a 1 value in the related columns (Figure 15).

image_dir	mask_dir	output_file_name	bin_width	shape	first_order	glszm	glrlm	ngtdm	gldm	glcm	LBP
...\image\	...\mask\	...\output_name									1

Figure 15. A sample data of template.csv file used by Biorad as an input setting for generating radiomics features. In this setting, the LBP features are selected.

The template.csv file contained the following data as an input configuration to the Biorad feature extraction module:

- image_dir: the image files’ directory.

- mask_dir: the mask files' directory. The name of the masks should match exactly the name of the corresponding image.
- output_file_name: the name of the output file or its directory. If the user did not insert the file path, the output file would be saved in the working folder.
- bin_width: the specific grey levels.
- The rest of the columns have the name of radiomics features: if the user typed a 1 in any of these columns, that feature would be extracted.

The output file is a CSV file containing the name of images and the specified radiomics features with their corresponding values. Figure 16 shows an example output file for LBP extracted features and shape features.

Name	Shape_Elongation	Shape_Flatness_	Shape_LeastAxisLength_	Shape_MajorAxisLength_	LBP_030_	LBP_201_	LBP_012_
000.nii	-0.014865775	0.011498284	0.685348743	0.594046463	-0.024	-0.0047	0.0187
001.nii	0.003753401	0.012039936	0.146292359	-1.554109875	0.18295	0.1614	-0.1072
002.nii	-0.019565529	-0.006011581	-0.026517035	0.976208119	-0.0611	0.0517	-0.0609

Figure 16. An example output file from the Biorad feature extraction module showing extracted shape and LBP features with the name of the corresponding image.

3.2.3 The Feature Matrices

Grünbeck [13] used two sets of MR images to analyse the changes in the brain structure of MPH-treated and placebo groups. The pre-treatment set or the baseline images was acquired before treatment started. The other set, the post-treatment, referred to the images captured at the 17-week of treatment. After the features were derived from the pre-treatment and post-treatment images separately, we obtained output files containing the extracted radiomics features for each side of the brain for each subcortical. The goal of this study was to assess potential changes in the brain structure due to MPH treatment. Therefore, for constructing the final datasets for each subcortical structure of the brain, pre-treatment (Pre_set) features were subtracted from the corresponding post-treatment ($Post\_set$) features by using equation 5 [13]:

$$C_{m,n} = Post\_set_{m,n} - Pre\_set_{m,n} \quad (5)$$

Here $C_{m,n}$ indicates the change of the feature value related to feature m, and sample n, in the feature set. Thus, the feature matrices contained the change of the corresponding radiomics feature. For each subcortical structure of the brain, we concatenated the feature matrices of the left and right part of the structure to construct the final datasets. The example in Table 3 illustrates the structure of the final dataset.

Table 3. An example of the structure of the final dataset constructed based on the change between post-treatment and pre-treatment features. Class 1 indicates the MPH-treated group, and class 0 denotes the placebo group [13].

Participant ID	Class	Left Segment Feature 1	Left Segment Feature 2	Right Segment Feature 1	Right Segment Feature 2
0	1	$C_{l1,0}$	$C_{l2,0}$	$C_{r1,0}$	$C_{r2,0}$
1	1	$C_{l1,1}$	$C_{l2,1}$	$C_{r1,1}$	$C_{r2,1}$
…					
22	0	$C_{l1,22}$	$C_{l2,22}$	$C_{r1,22}$	$C_{r2,22}$
23	0	$C_{l1,23}$	$C_{l2,23}$	$C_{r1,23}$	$C_{r2,23}$
…					

3.2.4 Datasets

For each subcortical structure (hippocampus, caudate, putamen, thalamus, and pallidum), three datasets were used as an input for modelling and comparison, the "initial dataset" and an "expanded dataset", and the "LBP dataset".

Initial Dataset

The “initial dataset” contained the shape features and texture features acquired in Grünbeck’s thesis [13]. In this way, we can compare our results with the results of her study. The “initial dataset” for each of the five subcortical brain structures has 46 rows corresponding to the 46 participants. The columns contained an *ID* column identifying patients, a *Label* column indicating class labels (0 or 1) and 328 radiomics features (28 shape feature and 300 texture features). Figure 17 illustrates the type and number of radiomics features included in the “initial dataset”.

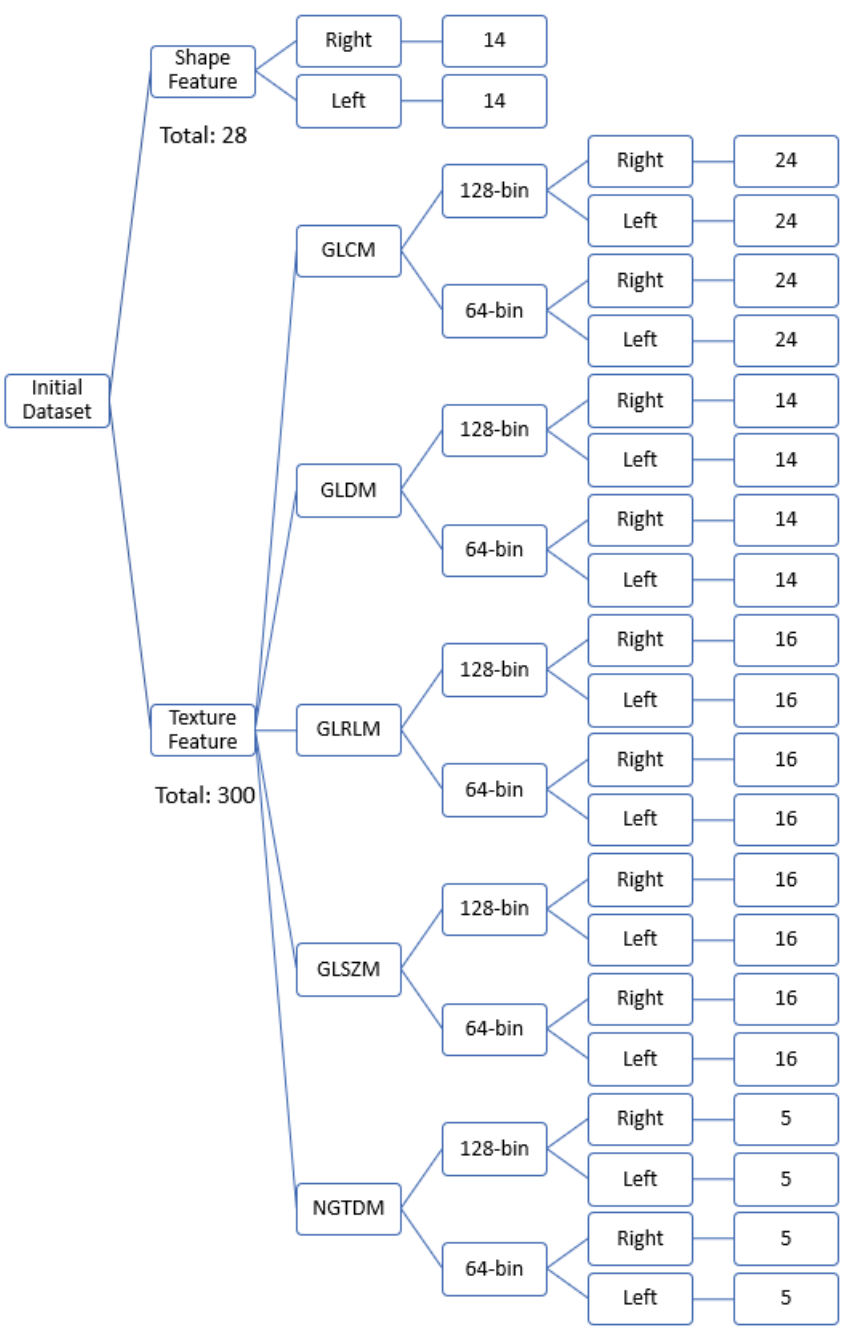


Figure 17. The structure of the radiomics features (in total 328) in the "initial dataset" for every subcortical structure of the brain.

The distribution of features in the "initial dataset" is illustrated in Figure 18. The shape feature comprised 8% of the whole dataset compared to texture feature 128-bin (46%) and 64-bin (46%). There was an equal number of features from each side of the brain.

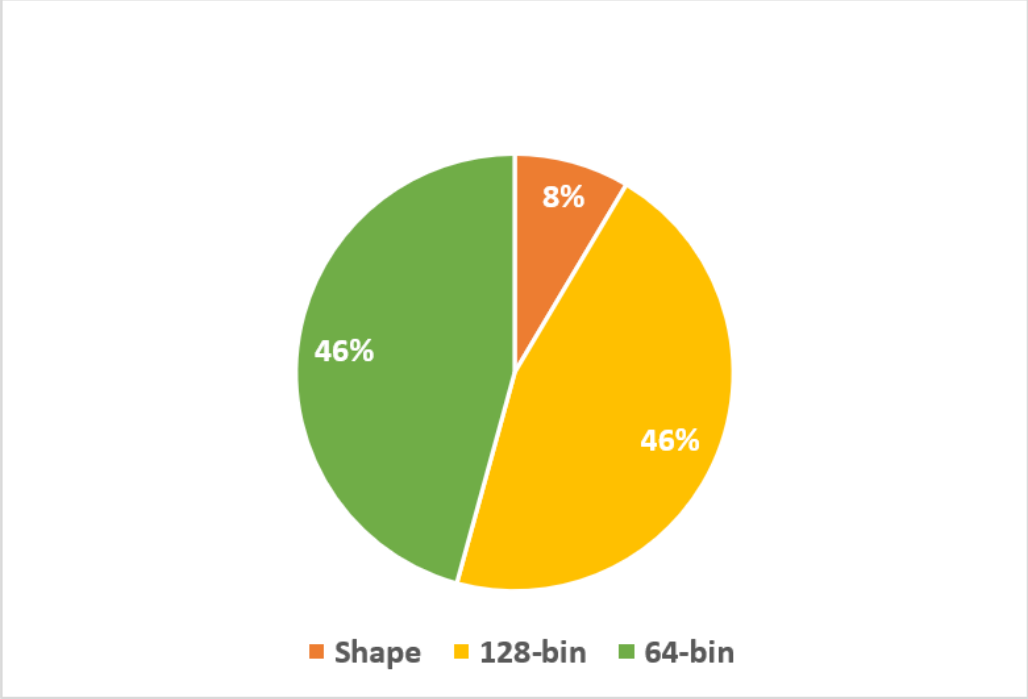


Figure 18. Pie chart shows the distribution of various radiomics features in the "initial dataset". 128-bin and 64-bin refer to the texture features with, respectively, 128 and 64 grey level discretisation. Shape denotes the shape features.

LBP Dataset

The *"LBP dataset"* for each subcortical brain structure has 46 rows corresponding to the 46 participants. The columns are an *ID* column identifying patients, a *Label* column indicating class labels (0 or 1) and 20 radiomics features (referring to 10 LBP pattern for each side of the brain). Figure 19 illustrates the type and number of radiomics features included in the *"LBP dataset"*.

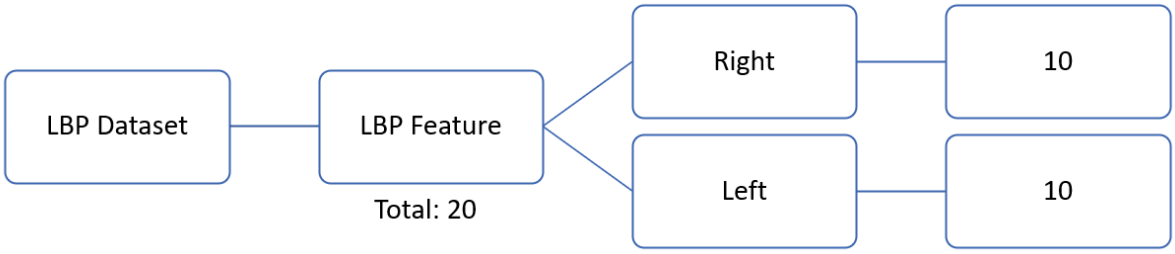


Figure 19. The structure of radiomics features (in total 20) in the "LBP dataset" for every subcortical structure of the brain.

Expanded Dataset

The "expanded dataset" contained the features in the "initial dataset" and the features in the "LBP dataset" (Figure 20). Thus, the "expanded dataset" for each subcortical brain structure, same as the mentioned datasets, has 46 rows corresponding to the

46 participants as well as an *ID and* a *Label* column. This dataset contained 348 radiomics features (28 shape feature, 300 texture features and 20 LBP features)

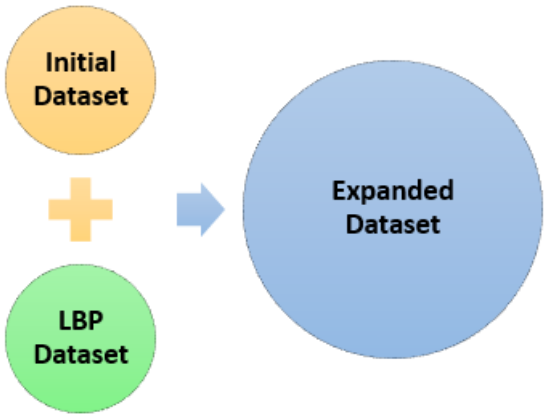


Figure 20. The structure of the "expanded dataset". It contains the features from both the "initial dataset" (shape and texture features) and the "LBP dataset" (LBP features).

Figure 21 shows the distribution of features that exists in the "expanded dataset". The LBP features comprised 6% of the whole dataset in comparison to shape feature (8%) texture feature 128-bin (43%) and 64-bin (43%). The number of features from each side of the brain was equal.

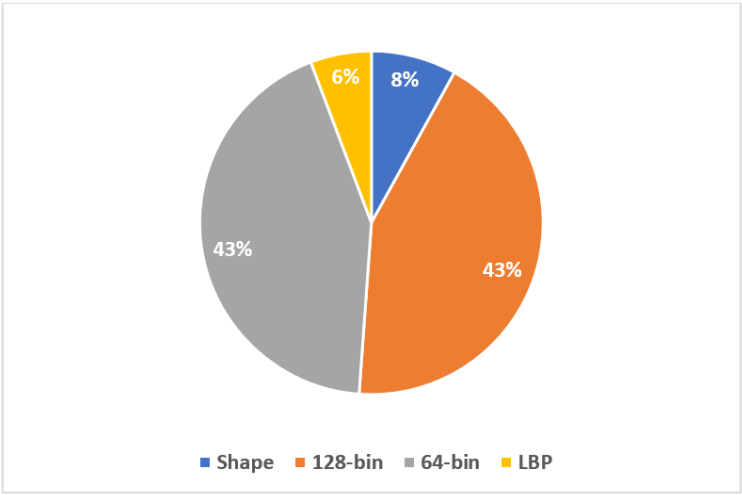


Figure 21. The pie chart shows the distribution of various radiomics features in the "expanded dataset". 128-bin and 64-bin refer to the texture features with, respectively, 128 and 64 grey level discretisation. Shape denotes the shape features, and LBP corresponds to LBP features.

3.3 Experiments

In this study, we performed four different experiments for each subcortical structure of the brain, separately, and we compared the results of these experiments in the result section. The overall workflow for performing each experiment is shown in Figure 22.

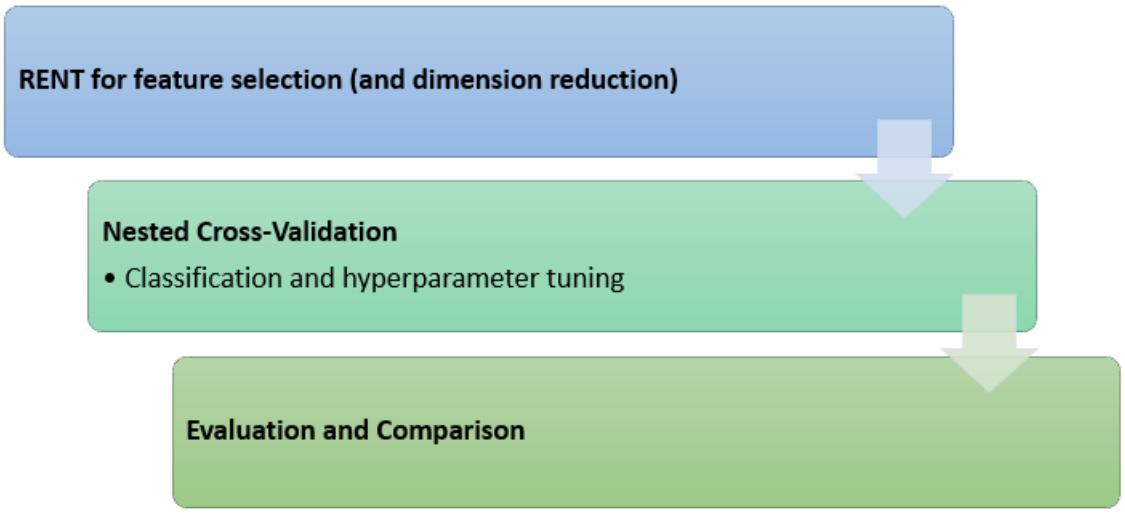


Figure 22. The workflow used for assessment for all experiments.

The only difference in the experiments is the different datasets used as the input dataset per experiment. All other steps are the same (Figure 22). For an overall overview of different experiments, see Table 4.

Table 4. An overview of various experiments. Note that the LBP dataset contained only 20 features; thus, experiment 4 did not have any feature selection step.

	Input Dataset	Feature selection method
Experiment 1	Initial dataset	RENT
Experiment 2	Expanded dataset	RENT
Experiment 3	Cleaned dataset by removing highly correlated features from "expanded dataset".	RENT
Experiment 4	LBP dataset	Not Applicable

3.3.1 Correlation Analysis

The radiomics features are prone to be highly correlated. Therefore, in experiment 3, we tried to examine the correlation between features. We aimed to investigate how RENT selects correlated features and assess our model without the correlated features. The “expanded dataset” was used in this experiment to analyse the

correlation coefficient of features. A cleaned dataset was created by removing the highly correlated features from the "expanded dataset" and used as the input dataset in experiment 3. We used Spearman's Rank Correlation Coefficient (SCC) [79] to find the correlation between features, and we removed one of the highly correlated (having SCC above 95%) features from pairs. We used the code from [80], which is available in Appendix D. In this code, one of the features from highly correlated pairs were arbitrarily removed. Also, the threshold value (95%) was selected arbitrarily. The cleaned dataset for each subcortical brain structure differed. In the result section, we describe the selected features and removed features in more detail.

Spearman's Rank Correlation Coefficient (SCC)

Spearman's Rank Correlation Coefficient (SCC) [79] measures the statistical association of two features in terms of their ranks [81]. It is a nonparametric (distribution free) metric [81]. It is used when the data between two features is not normally distributed [82]. Equation 6 presents the calculation of SCC for two features, A and B:

$$SCC(A,B) = \rho_{r_A,r_B} = \frac{COV(r_A,r_B)}{\sigma_{r_A}.\sigma_{r_B}} \quad (6)$$

Here ρ is the Pearson correlation coefficient of the ranked features (r_A, r_B), $COV(r_A, r_B)$ is the covariance matrix, and σ_{r_A} and σ_{r_B} are the standard deviations.

SCC ranges from -1 to +1, where both endpoints show a perfect negative or positive correlation, respectively [82]. Thus, SCC = 0 represents lack of correlation.

3.3.2 Feature Selection Using RENT

In current study, we used RENT for selecting feature (and dimension reduction). RENT is a new feature selection method designed for short-wide datasets [23]. By RENT, we could reduce the dimension of our dataset from hundreds of features to a few features.

As is mentioned in chapter 2, RENT tries to select robust features by creating many ensemble models [23]. In our study, we configured RENT to build the ensemble penalised LR models based on accuracy score. The accuracy score is a widespread metric in classification defined as the proportion of correctly predicted samples [63]. Here, we used 100 sub-models. In each model, 80% of the samples were randomly assigned to the train set, and the remaining 20% of the samples were assigned to the test set. All of the configurations of RENT models performed in this research are available in Appendix E.

It should be pointed out that in experiments related to the hippocampus set, we performed RENT two times to make robust models and to test RENT's ability in making polynomial features. The polynomial option of RENT makes new features by squaring each variable to capture nonlinearities and multiplying pairs of variables to obtain variable interactions. In the rest of the structures sets (putamen, caudate, thalamus and pallidum), we only performed RENT once without polynomial features because the original features had sufficient power to show satisfactory prediction performances without modification.

3.3.3 Modelling and Evaluation

Nested Cross-Validation

One of the limitations of this study was the lack of an unseen validation set. In addition, there were very few samples that make the modelling prone to be overfitted. For decreasing the generalisation error (overfitting degree), we used nested cross-validation with five outer and three inner folds. For implementing the nested cross-validation, we used the modified code published in [83].

Nested cross-validation is also appropriate when we need to optimise the hyperparameters [55]. We used GridSearchCV from the Scikit-learn package [55], which performs an exhaustive search over different combinations of given parameters for a classifier. GridSearchCV is one of the basic tuning methods which is suitable for small datasets. Since our final datasets (after feature selection) were small, the execution time of performing GridSearchCV was not an issue in our modelling phase.

Figure 23 illustrates the modelling and evaluation process employed in our research for each classifier. First, the hyperparameters were tuned in the inner loop of the nested cross-validation. Then, in the outer loop, the prediction model was established on the validation fold using the best hyperparameters of training in the inner loop. After that, one model showing the less difference between training and validation performance scores from all outer loops was chosen to make the final prediction on the whole dataset. In the case of an available external dataset, this final prediction should be made on this dataset.

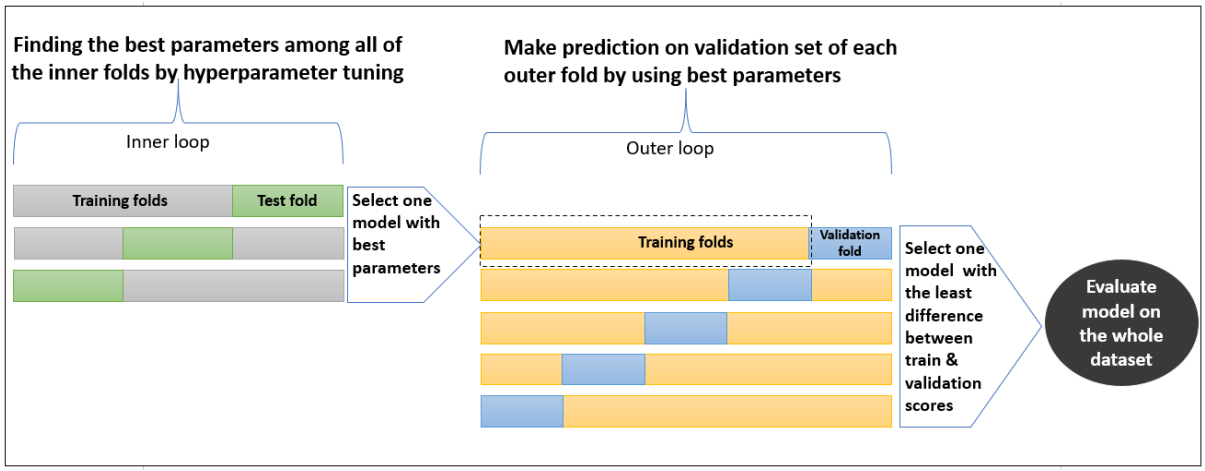


Figure 23. The entire process of modelling for a typical classifier using nested cross-validation and hyperparameter tuning.

Supervised Classifiers

The modelling has been done by applying several supervised classifiers suitable for binary classification to examine whether the class labels were detectable based on radiomics features. We tried to include all of the algorithms used in Grünbeck's study [13] to compare our results with the results achieved in that study (Ridge, LGBM, SVC, DT, LR, ET). We also applied some other popular classifiers (such as RF, KNN, MLP, AdaBoost). The name of the classifiers used is shown in Table 5. For an understanding of abbreviations, see the List of Abbreviations.

Table 5. Supervised classifiers names. All of the classifiers were from scikit-learn except Light Gradient Boosting Machine. This classifier was implemented by the LGBM python package.

Classifier	Source
Logistic Regression (LR)	Scikit-learn [55]
Support Vector Machine (SVC)	
K Nearest Neighbours (KNN)	
Multi-Layer Perceptron (MLP)	
Decision Tree (DT)	
Random Forest (RF)	
Ridge Regression (Ridge)	
AdaBoost	
Extremely Randomised Tree (ET)	
Light Gradient Boosting Machine (LGBM)	LightGBM [70]

4 Results

This chapter outlines the results obtained from performing the experiments mentioned earlier in chapter 3, including feature selection and modelling and evaluation steps from the radiomics pipeline (Figure 1).

First, we presented the detailed results of experiments related to the brain's hippocampus region (both left and right side). Then, we described the final selected features and the classifiers' performance scores regarding other subcortical structures (caudate, pallidum, putamen, thalamus).

4.1 The Hippocampus

4.1.1 Feature Selection by RENT

In this study, RENT was used as the feature selection method for all the experiments. As mentioned in chapter 3, for the hippocampus set, we applied RENT two times; first, we performed RENT for feature reduction. Then we applied RENT to both feature reduction and improve our model by generating polynomial features (quadratic form and interaction between features). We noted that this approach ameliorated the final classification results remarkably by enhancing the effect of powerful features.

RENT Parameters Selection Matrices

RENT tries to find the best combination of C (the inverse of regularisation strength) and l1_ratio (the elastic net mixing parameter) by training several sub-models. In RENT, the three matrices (scores, zeroes and harmonic) are the basis for choosing the best C and l1_ratio combination. RENT aims to attain the highest possible performance by the largest possible feature reduction. We can perceive how RENT decides on the best combination of parameters by observing these three matrices:

- The scores matrix shows the model performance of the various combinations of C and l1_ratio based on accuracy score. We can acquire the maximum performance by observing this matrix.
- The proportion of features set to zero in every combination are shown in the zeroes matrix. This matrix gives us the combination of C and l1_ratio with the greatest possible feature reduction.
- Finally, to choose the best combination and fulfil the highest possible performance with the largest possible feature reduction goal, we should bring the mentioned metrics in one comparable scale. Thus, RENT makes a harmonic matrix contained the harmonic mean of scores and zeroes matrices. The maximum value from the harmonic matrix provides the optimal combination of parameters. The harmonic mean formula is shown in equation 7.

$$Harmonics\ mean\ (scores, zeroes) = 2 \times \left(\frac{scores \times zeroes}{scores + zeroes}\right) \quad (7)$$

It should be borne in mind that the rows and columns in the zeroes matrix with precisely 0 or 1 values are overlooked because the values of this matrix provide the ratio of features weights set to zero. Hence, a 1 value presents the case of choosing no feature (all the weights are zero), while the 0 value corresponds to the situation that all the features are selected (none of the weights is set to zero). It is clear that these two situations are not desired.

The values for the optimal l1_ratio and the C parameter found for each experiment below were used as the basis for the RENT feature selection for these experiments.

RENT Matrices for Experiment 1

As shown in Figure 24, the maximum accuracy score (0.664) was obtained by the combination l1_ratio = 1 and C = 10, whereas the largest possible feature reduction was not obtained for this combination. The greatest proportion of features (0.999) were set to zero by having l1_ratio = 0.1 and C = 0.01 and also with l1_ratio = 1 and C = 0.1. To determine which combination accomplished the aim of RENT (the highest accuracy score with the highest feature reduction), the harmonic mean of these two matrices (scores and zeroes) was calculated.

In the harmonic matrix (Figure 24), we could observe that the best combination with the highest harmonic mean value (0.796) was obtained for l1_ratio = 0.2 and C = 0.1. This combination was the optimal parameters' combination for RENT to select features in experiment 1. For this combination, the accuracy score was 0.634, and 0.773 of the feature weights were set to zero.

Experiment 1 Matrices

(Scores	0.01	0.10	1.00	10.00	100.00
0.0	0.639	0.644	0.626	0.630	0.635
0.1	0.500	0.650	0.648	0.634	0.635
0.2	0.500	0.634	0.656	0.637	0.635
0.3	0.500	0.574	0.656	0.642	0.635
0.4	0.500	0.530	0.646	0.646	0.635
0.5	0.500	0.509	0.637	0.645	0.639
0.6	0.500	0.522	0.634	0.649	0.639
0.7	0.500	0.526	0.630	0.654	0.640
0.8	0.500	0.501	0.625	0.662	0.640
0.9	0.500	0.497	0.620	0.660	0.643
1.0	0.500	0.500	0.617	0.664	0.645,

Zeroes	0.01	0.10	1.00	10.00	100.00
0.0	0.0	0.0	0.0	0.0	0.0
0.1	0.999882	0.627	0.303647	0.064412	0.002176
0.2	1.0	0.773882	0.477118	0.143824	0.007882
0.3	1.0	0.849588	0.582353	0.216412	0.016
0.4	1.0	0.904118	0.648118	0.279941	0.026059
0.5	1.0	0.944824	0.699941	0.334706	0.037294
0.6	1.0	0.971059	0.743941	0.388765	0.047588
0.7	1.0	0.987294	0.777412	0.437824	0.059706
0.8	1.0	0.996412	0.806941	0.479059	0.070529
0.9	1.0	0.999	0.833	0.519706	0.081412
1.0	1.0	0.999882	0.859294	0.555	0.092353,

Harmonic Mean	0.01	0.10	1.00	10.00	100.00
0.0	0.000000	0.000000	0.000000	0.000000	0.000000
0.1	0.035294	0.744491	0.454622	0.119445	0.004342
0.2	0.035294	0.796443	0.635681	0.245525	0.015616
0.3	0.035294	0.597752	0.722678	0.346468	0.031392
0.4	0.035294	0.324325	0.750825	0.426168	0.050524
0.5	0.035294	0.133555	0.762908	0.485900	0.071454
0.6	0.035294	0.259410	0.780284	0.544820	0.090132
0.7	0.035294	0.295356	0.786795	0.597422	0.111628
0.8	0.035294	0.046780	0.786184	0.645255	0.130324
0.9	0.035294	0.000000	0.781799	0.678264	0.148953
1.0	0.035294	0.035294	0.782652	0.713826	0.167274)

Figure 24. RENT determinative matrices in experiment 1 for the hippocampus. Scores, Zeroes and Harmonic Mean matrices show the accuracy score, the fraction of feature weights set to zero, and the harmonics mean of these two matrices, respectively. The l1_ratio values are specified in the first column, and the C parameter (inverse of regularisation strength) values are specified in the remaining columns. The red boxes lineate the highest value of that matrix.

RENT Matrices for Experiment 2

In Figure 25, one can see that in the scores matrix, the highest accuracy (0.856) was attained by the l1_ratio = 0.4 and C = 1. At the same time, in the zeroes matrix, the maximum fraction of features (0.982) was set to zero by the combination of the l1_ratio = 1 and C = 0.1. However, the best combination was acquired by the l1_ratio = 1 and

C = 1 from the harmonic matrix with the harmonic value of 0.884. This combination had a performance score of 0.832, and 0.841 of features' weights was set to zero.

Experiment 2 Matrices

(Scores	0.01	0.10	1.00	10.00	100.00
0.0	0.841	0.832	0.822	0.837	0.840
0.1	0.500	0.850	0.851	0.846	0.841
0.2	0.500	0.847	0.851	0.851	0.842
0.3	0.500	0.835	0.854	0.854	0.843
0.4	0.500	0.830	0.856	0.852	0.845
0.5	0.500	0.814	0.852	0.854	0.846
0.6	0.500	0.798	0.847	0.852	0.847
0.7	0.500	0.787	0.840	0.853	0.849
0.8	0.500	0.765	0.837	0.849	0.851
0.9	0.500	0.702	0.840	0.850	0.852
1.0	0.500	0.603	0.832	0.842	0.853,

Zeroes	0.01	0.10	1.00	10.00	100.00
0.0	0.0	0.0	0.0	0.0	0.0
0.1	0.969889	0.490444	0.286778	0.078667	0.002667
0.2	1.0	0.653667	0.456889	0.182	0.013222
0.3	1.0	0.751111	0.559111	0.276889	0.028667
0.4	1.0	0.817778	0.623556	0.364667	0.046
0.5	1.0	0.869333	0.672667	0.437889	0.062889
0.6	1.0	0.912556	0.717333	0.499222	0.081444
0.7	1.0	0.941	0.753778	0.553667	0.100222
0.8	1.0	0.958889	0.788556	0.601	0.116667
0.9	1.0	0.971444	0.815333	0.640222	0.137
1.0	1.0	0.982556	0.841444	0.672444	0.157111,

Harmonic Mean	0.01	0.10	1.00	10.00	100.00
0.0	0	0.000000	0.000000	0.000000	0.000000
0.1	0	0.654427	0.444319	0.145552	0.005319
0.2	0	0.782544	0.624422	0.307279	0.026085
0.3	0	0.835405	0.715768	0.433162	0.055677
0.4	0	0.868955	0.768136	0.532823	0.087831
0.5	0	0.875632	0.800646	0.608026	0.118134
0.6	0	0.873189	0.826450	0.663464	0.150328
0.7	0	0.868388	0.842563	0.710571	0.181853
0.8	0	0.838128	0.860392	0.745170	0.208645
0.9	0	0.716391	0.879681	0.775464	0.240656
1.0	0	0.447021	0.884673	0.791124	0.271245)

Figure 25. RENT determinative matrices in experiment 2 for the hippocampus. Scores (the accuracy score), Zeroes (the fraction of feature weights set to zero) and Harmonic Mean (harmonics mean of Scores and Zeroes two matrices). The l1_ratio values are specified in the first column, and the C parameter (inverse of regularisation strength) values are specified in the remaining columns. The red boxes marked the highest value of that matrix.

RENT Matrices for Experiment 3

Although the best score (0.856) was attained by the l1_ratio = 0.5 and C = 1, the largest feature reduction (0.991) was achieved by the l1_ratio = 1 and C = 0.1.

Therefore, the harmonic mean was computed to find the best trade-off. The l1_ratio = 0.8 and C = 1 was used by RENT as the optimal parameters with the highest harmonic mean value of 0.909. This combination had a score of 0.840, and 0.868 of feature weights were set to zero (Figure 26).

Experiment 3 Matrices

Scores	0.01	0.10	1.00	10.00	100.00
0.0	0.820	0.817	0.806	0.806	0.805
0.1	0.504	0.828	0.818	0.807	0.806
0.2	0.500	0.823	0.836	0.807	0.805
0.3	0.500	0.825	0.841	0.809	0.804
0.4	0.500	0.816	0.851	0.818	0.804
0.5	0.500	0.809	0.856	0.818	0.805
0.6	0.500	0.791	0.850	0.817	0.806
0.7	0.500	0.783	0.847	0.825	0.806
0.8	0.500	0.765	0.840	0.836	0.807
0.9	0.500	0.702	0.828	0.840	0.807
1.0	0.500	0.602	0.818	0.843	0.806,

Zeroes	0.01	0.10	1.00	10.00	100.00
0.0	0.0	0.0	0.0	0.0	0.0
0.1	0.985714	0.648095	0.437566	0.108413	0.00455
0.2	1.0	0.77709	0.622381	0.237196	0.015979
0.3	1.0	0.851058	0.707778	0.356561	0.031376
0.4	1.0	0.897407	0.761534	0.456138	0.048995
0.5	1.0	0.931852	0.798042	0.531111	0.066772
0.6	1.0	0.956349	0.825608	0.589048	0.08582
0.7	1.0	0.971429	0.849206	0.63582	0.106032
0.8	1.0	0.980265	0.868942	0.67328	0.123651
0.9	1.0	0.986296	0.884921	0.702011	0.140741
1.0	1.0	0.991693	0.900212	0.725556	0.158783,

Harmonic Mean	0.01	0.10	1.00	10.00	100.00
0.0	0.000000	0.000000	0.000000	0.000000	0.000000
0.1	0.022219	0.760934	0.587395	0.192611	0.009053
0.2	0.000000	0.837164	0.750115	0.372056	0.031373
0.3	0.000000	0.880905	0.814047	0.505475	0.060527
0.4	0.000000	0.892497	0.859334	0.603898	0.092672
0.5	0.000000	0.898781	0.887679	0.666147	0.123889
0.6	0.000000	0.881441	0.897517	0.709048	0.156059
0.7	0.000000	0.874370	0.907644	0.749581	0.188777
0.8	0.000000	0.846192	0.909966	0.785920	0.216289
0.9	0.000000	0.720391	0.902767	0.809212	0.241988
1.0	0.000000	0.444585	0.896722	0.827761	0.268050

Figure 26. RENT determinative matrices in experiment 3 for the hippocampus. Scores, Zeroes and Harmonic Mean matrices show the accuracy score, the fraction of feature weights set to zero, and the harmonics mean of these two matrices, respectively. The l1_ratio values are specified in the first column, and the C parameter (inverse of regularisation strength) values are specified in the remaining columns. The red boxes indicate the highest value of that matrix.

Selected Features Characteristics

As mentioned above, for the hippocampus set, we performed RENT two times. It should be noted that for presenting the selection rate in pie charts of this section (related to the hippocampus set), the polynomial features were counted in such a way that if we had an interaction between two features, we counted each one as one time of selection. A feature in quadratic form was counted two times. We used this approach to consider the effect of polynomial features.

Selected Features in Experiment 1

As mentioned in chapter 3, experiment 1 used the “initial dataset” for analysis (see Figure 18). In experiment 1, at the first round of applying RENT, we obtained a reduced dataset with 17 features (from the 328 radiomics features in the "initial dataset"). However, this was not the final selected features set. We used this reduced dataset to perform RENT a second time to generate polynomial features of these 17 features and RENT again selected some features from this new dataset. The final selected features set contained 14 features where 5 features were polynomial, and the remaining 9 features were as in the “initial dataset”. This dataset with 14 features was used for modelling and evaluation for the hippocampus in experiment 1. A list of selected features’ names for experiment 1 is provided in Table 6.

Table 6. Selected features attribute in experiment 1 for the hippocampus. 128-bin and 64-bin refer to the texture features with 128 and 64 grey level discretisation. Right or Left indicate the right or left side of the brain, respectively.

Feature Name	Polynomial	Side	Feature Type
128_GrayLevelVariance_right	No	Right	128-bin
128_LargeAreaLowGrayLevelEmphasis_right	No	Right	128-bin
128_HighGrayLevelRunEmphasis_left	No	Left	128-bin
128_HighGrayLevelEmphasis_left	No	Left	128-bin
128_LargeDependenceLowGrayLevelEmphasis_right	No	Right	128-bin
128_SmallDependenceHighGrayLevelEmphasis_right	No	Right	128-bin
64_GrayLevelVariance_right	No	Right	64-bin
64_HighGrayLevelEmphasis_left	No	Left	64-bin
64_LargeDependenceLowGrayLevelEmphasis_right	No	Right	64-bin
128_GrayLevelVariance_right*128_HighGrayLevelRun Emphasis_left	Yes	Right & Left	128-bin
128_GrayLevelVariance_right*128_ShortRunHighGray LevelEmphasis_left	Yes	Right & Left	128-bin
128_GrayLevelVariance_right*128_HighGrayLevelEm phasis_left	Yes	Right & Left	128-bin
128_GrayLevelVariance_right*64_HighGrayLevelEmp hasis_left	Yes	Right & Left	128-bin & 64-bin
64_LargeDependenceLowGrayLevelEmphasis_right^2	Yes	Right	64-bin

The final selected features characteristics are depicted in Figure 27. It can be perceived that all the selected features were texture features; 68% were of the 128-bin type versus 32% from the 64-bin type. No features from the shape features category were chosen for the final reduced dataset in experiment 1. Figure 27b shows that most of the selected features were from the right side of the brain (65%).

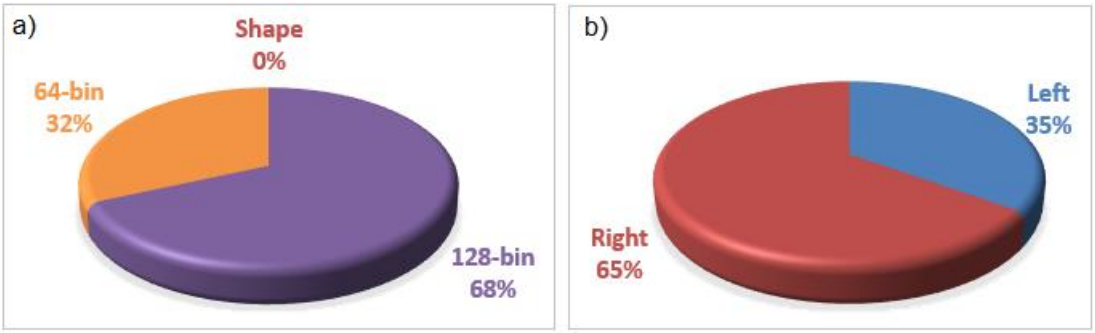


Figure 27. Pie charts show the distribution of selected features from the "initial dataset" in experiment 1 for the hippocampus after the second run of RENT considering polynomial features. a) the distribution of selected features based on the feature type. 128-bin and 64-bin refer to the texture features with 128 and 64 grey level discretisation. Shape denotes the shape features. b) the distribution of features selected from the left or right sides of the brain.

Selected Features in Experiment 2

In experiment 2, we utilised the “expanded dataset” containing LBP features plus the shape feature and texture features of 128 and 64 grey scale discretisation (see Figure 21).

Here, 12 features (from 348 radiomics features in the "expanded dataset”) were selected at the first run of RENT. We used these selected features and their polynomial forms as the input for the second round of performing RENT for constituting the final reduced dataset. From this set, RENT selected 13 features (4 polynomial features versus 9 features that existed in the “expanded dataset”) as the final selected feature set. Subsequently, this reduced dataset (with 13 features) was used for modelling and evaluation in experiment 2. Table 7 provides the list of selected features’ names for experiment 2.

Table 7. Selected features attribute in experiment 2 for the hippocampus. Shape denotes the shape features. LBP corresponds to LBP features. 128-bin refers to the texture features with 128 grey level discretisation. Right or Left indicate the right or left side of the brain, respectively.

Feature Name	Polynomial	Side	Feature Type
Shape_MeshVolume_right	No	Right	Shape
128_SizeZoneNonUniformity_right	No	Right	128-bin
128_LargeDependenceLowGrayLevelEmphasis_right	No	Right	128-bin
LBP_111_left	No	Left	LBP
LBP_300_left	No	Left	LBP
LBP_300_right	No	Right	LBP
LBP_021_right	No	Right	LBP
LBP_030_right	No	Right	LBP
LBP_102_right	No	Right	LBP
LBP_111_left^2	Yes	Left	LBP
LBP_012_left^2	Yes	Left	LBP
LBP_012_left*LBP_030_right	Yes	Left & Right	LBP
LBP_300_right*LBP_102_right	Yes	Right	LBP

By observing the selection rates of the final selected features set (Figure 28a) and the selected feature set attributes (Table 7), it is apparent that LBP features were dominated having the selection rate of 82%. Notably, the polynomial forms were all from LBP features. On the other hand, only 12% of selected features were texture 128-bin features versus none of the 64-bin features. In this experiment, 6% of selected features were shape feature. Figure 28b shows that the features from the right side of the brain had higher selection rate (56%) than from the left side (44%).

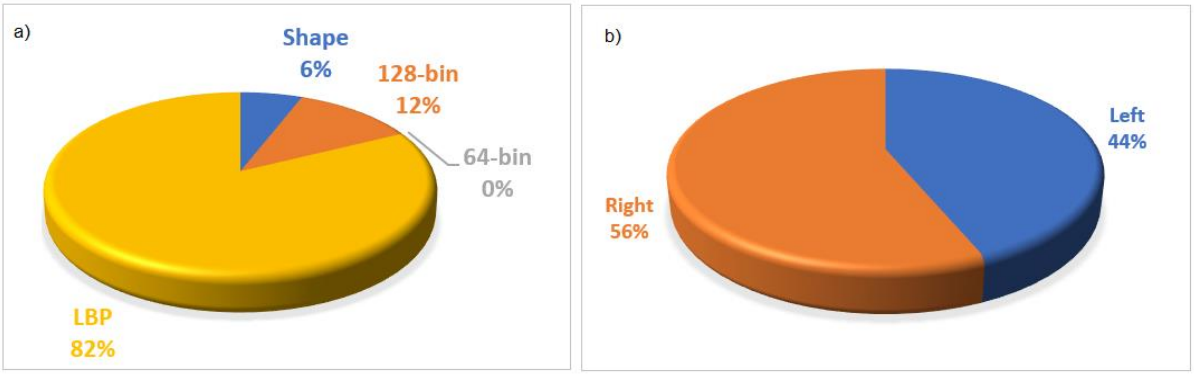


Figure 28. Pie charts show the characteristics of selected features from the "expanded dataset" in experiment 2 for the hippocampus after the second run of RENT considering polynomial features. a) the distribution of selected features based on the feature type. LBP corresponds to the LBP features. 128-bin and 64-bin refer to the texture features with, respectively, 128 and 64 grey level discretisation. Shape denotes the shape features. b) the distribution of features selected from the left or right sides of the brain.

Selected Features and Feature Correlation in Experiment 3

Features Collinearity

We examined the correlation of the features selected by RENT in experiment 2 after the first round of feature selection. The corresponding heatmap of selected features' correlations in terms of Spearman Correlation Coefficient (SCC) is shown in Figure 29. The features are the 12 features (from 348 radiomics features in the "expanded dataset") obtained after the first round of performing RENT in experiment 2.

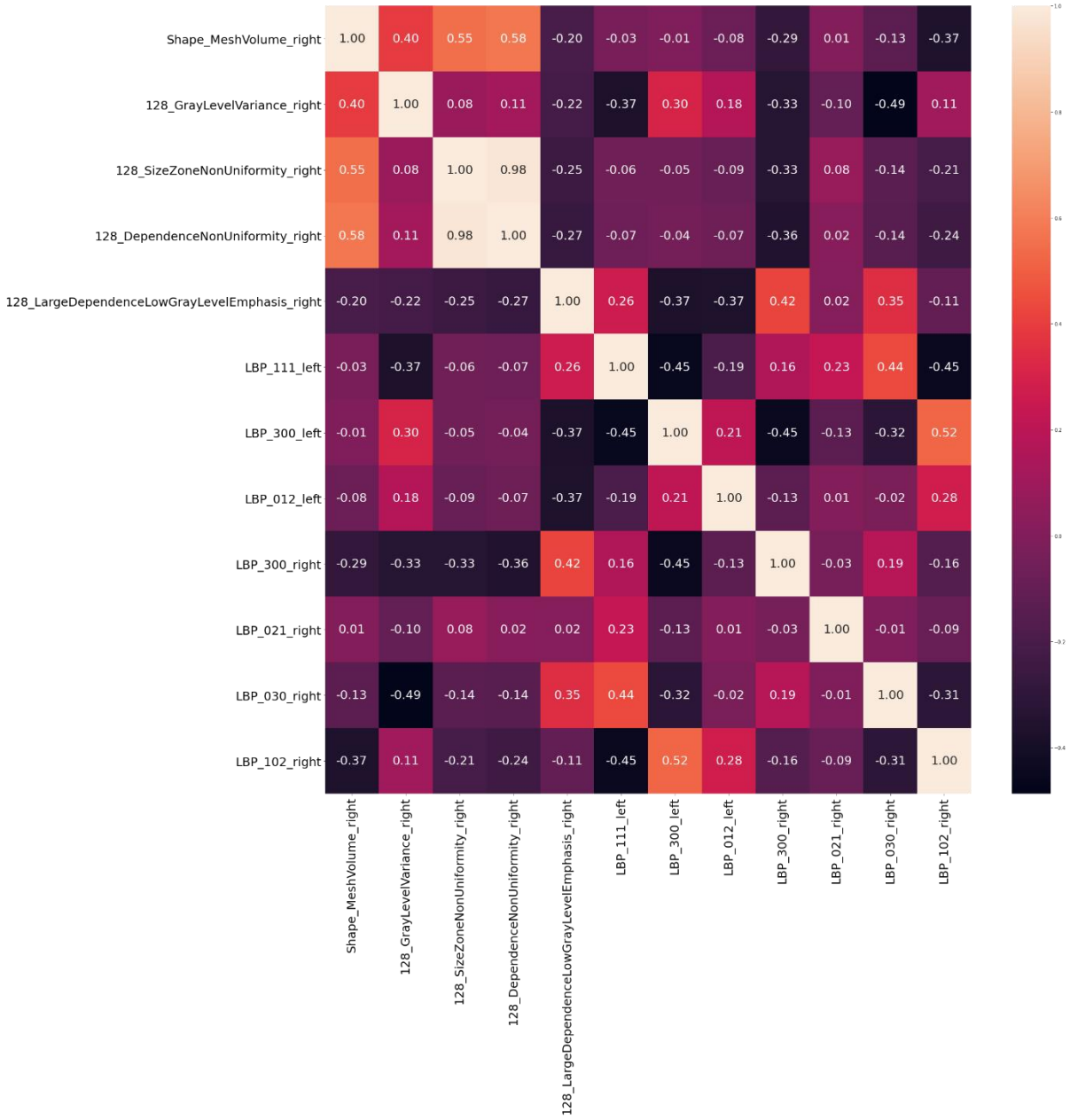


Figure 29. The correlation heatmap of features selected by RENT in the first round of performing RENT in experiment 2 for the hippocampus. The values show the Spearman Correlation Coefficient between pairs of features.

Figure 29 demonstrates that the only features with correlation above 70% in the features selected by RENT in experiment 2 were *128_DependenceNonUniformity_right* and *128_SizeZoneNonUniformity_right*. We can see in Table 7 that after the second run of RENT, only one of these two features (*128_SizeZoneNonUniformity_right*) was included in the final reduced dataset of experiment 2.

Selected Features Selection in Experiment 3

In experiment 3, we removed one of the features from pairs having above 95% SCC in the "expanded dataset" (having 348 features). There were 159 features highly correlated to another feature. We removed these features from the "expanded dataset" and used this reduced dataset (with 189 features) as the input to RENT. After that, we performed RENT two times (for the hippocampus set).

The distribution of features in the dataset obtained after removing highly correlated features is shown in Figure 30. The LBP features comprised 11% of the whole dataset compared to shape feature 14%, texture feature 128-bin 43% and 64-bin 32% (Figure 30a). It should be pointed out that all the LBP features were included after removing highly correlated features. The features set contained features 52% from the right side of the brain (Figure 30b).

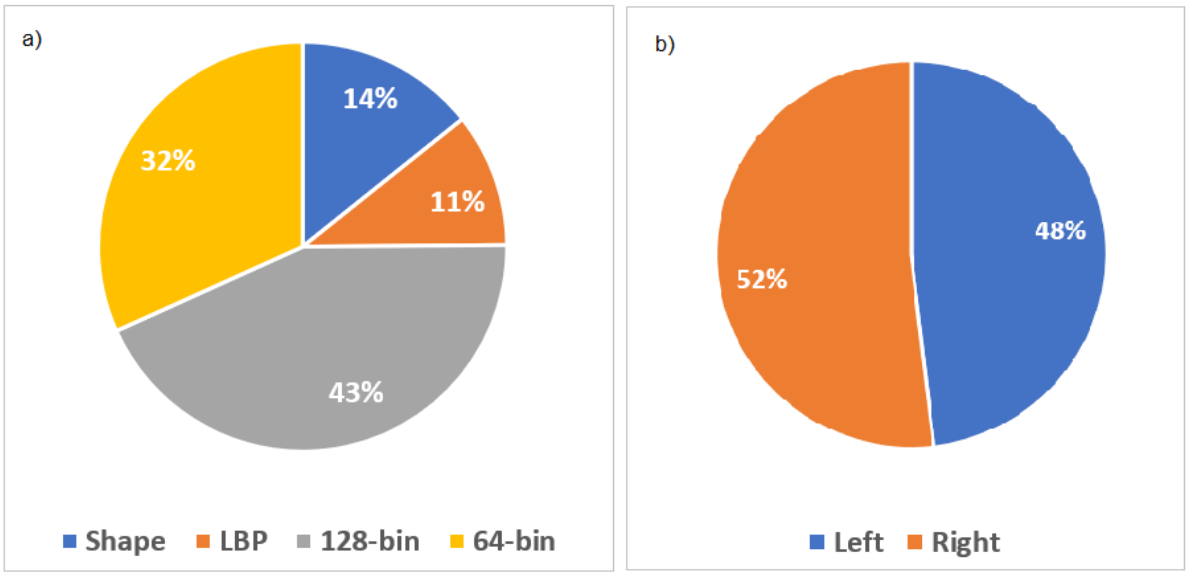


Figure 30. The pie charts show the distribution of various radiomics features in the dataset obtained after removing highly correlated features from the "expanded dataset" in experiment 3 for the hippocampus. a) the distribution of features based on the feature type. 128-bin and 64-bin refer to the texture features, respectively, 128 and 64 grey level discretisation. Shape denotes the shape features. LBP corresponds to LBP features. b) the distribution of features from the left or right sides of the brain.

In experiment 3, after removing highly correlated features, we used this dataset as the input to RENT. At the first round of applying RENT, we obtained a reduced dataset with 18 features (from 189 radiomics features). Then, we performed RENT a second time to generate polynomial forms of these 18 features. The selected features (contained 10 features where 3 features were polynomial) were used as the input for modelling and evaluation in experiment 3. In Table 8, the name and characteristics of selected features in experiment 3 is illustrated.

Table 8. Selected features attribute in experiment 3 for the hippocampus. Shape denotes the shape features. LBP corresponds to LBP features. 128-bin refers to the texture features with 128 grey level discretisation. Right or Left indicate the right or left side of the brain, respectively.

Feature Name	Polynomial	Side	Feature Type
Shape_MeshVolume_right	No	Right	Shape
128_SizeZoneNonUniformity_right	No	Right	128-bin
LBP_111_left	No	Left	LBP
LBP_300_left	No	Left	LBP
LBP_300_right	No	Right	LBP
LBP_021_right	No	Right	LBP
LBP_102_right	No	Right	LBP
Shape_Maximum2DDiameterColumn_right* 128_SizeZoneNonUniformity_right	Yes	Right	Shape & 128-bin
LBP_111_left^2	Yes	Left	LBP
LBP_300_right*LBP_102_right	Yes	Right	LBP

From Figure 31a and Table 8 one could observe that most of the selected features (69%) were LBP features. On the other hand, only 15% of selected features were texture 128-bin features versus none of the 64-bin features. In this experiment, 16% of selected features were shape feature. Figure 31b demonstrates that the features from right side of the brain were preferred to the left side (69% versus 31%).

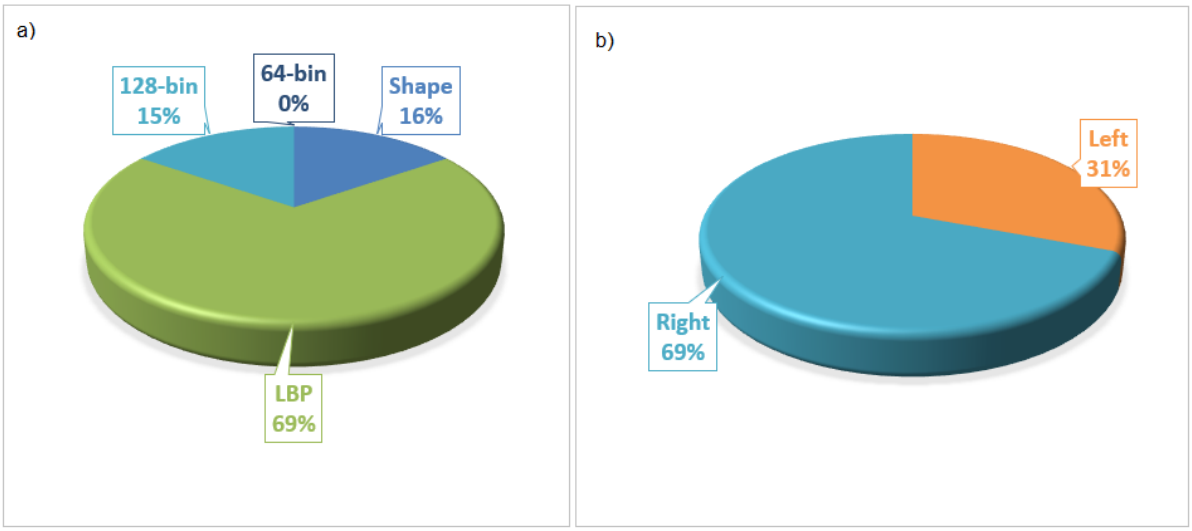


Figure 31. The pie charts show the characteristics of selected features from the dataset obtained after removing highly correlated features from the "expanded dataset" and after the second run of RENT considering polynomial features in experiment 3 for the hippocampus. a) the distribution of selected features based on the feature type. LBP corresponds to the LBP features. 128-bin and 64-bin refer to the texture features with, respectively, 128 and 64 grey level discretisation. Shape denotes the shape features. b) the distribution of features selected from the left or right sides of the brain.

4.1.2 Classification Modelling and Evaluation

We conducted four classification experiments regarding different datasets. In each classification experiment, various classifiers (for each) with the best hyperparameters and a nested cross-validation model approach were used. First, we present how the best models were selected by using nested cross-validation and hyperparameter tuning, and after that, the models' performances are compared.

Train and Validation Curves

The train and validation curves were the basis for finding the best classifiers' hyperparameters in a nested cross-validation model. These train and validation curves also helped us perceive classifiers' prediction behaviour on unseen data. Moreover, in the lack of independent validation set, it was a way of reducing the risk of overfitting and making more reliable decision in selecting the final hyperparameters.

In this section, the train curves present the GridSearchCV's best score of training folds of every outer fold, and validation curves demonstrate the prediction performance of the classifier (with the best hyperparameters set) on the validation fold of the same outer fold (see Figure 23). We chose the estimator's configuration having the smallest difference between its train and validation score as the hyperparameters of the final prediction task.

Figure 32 to Figure 35 presents the train and validation curves used to choose the classifier's best hyperparameters' configuration for each experiment. There was variation between folds observed in the train and validation curve plots, showing how dependent this small dataset is on the splits. Although the train and validation AUC of the LGBM classifier overlapped, we will see in the overall performance that in our study, LGBM had a poor performance.

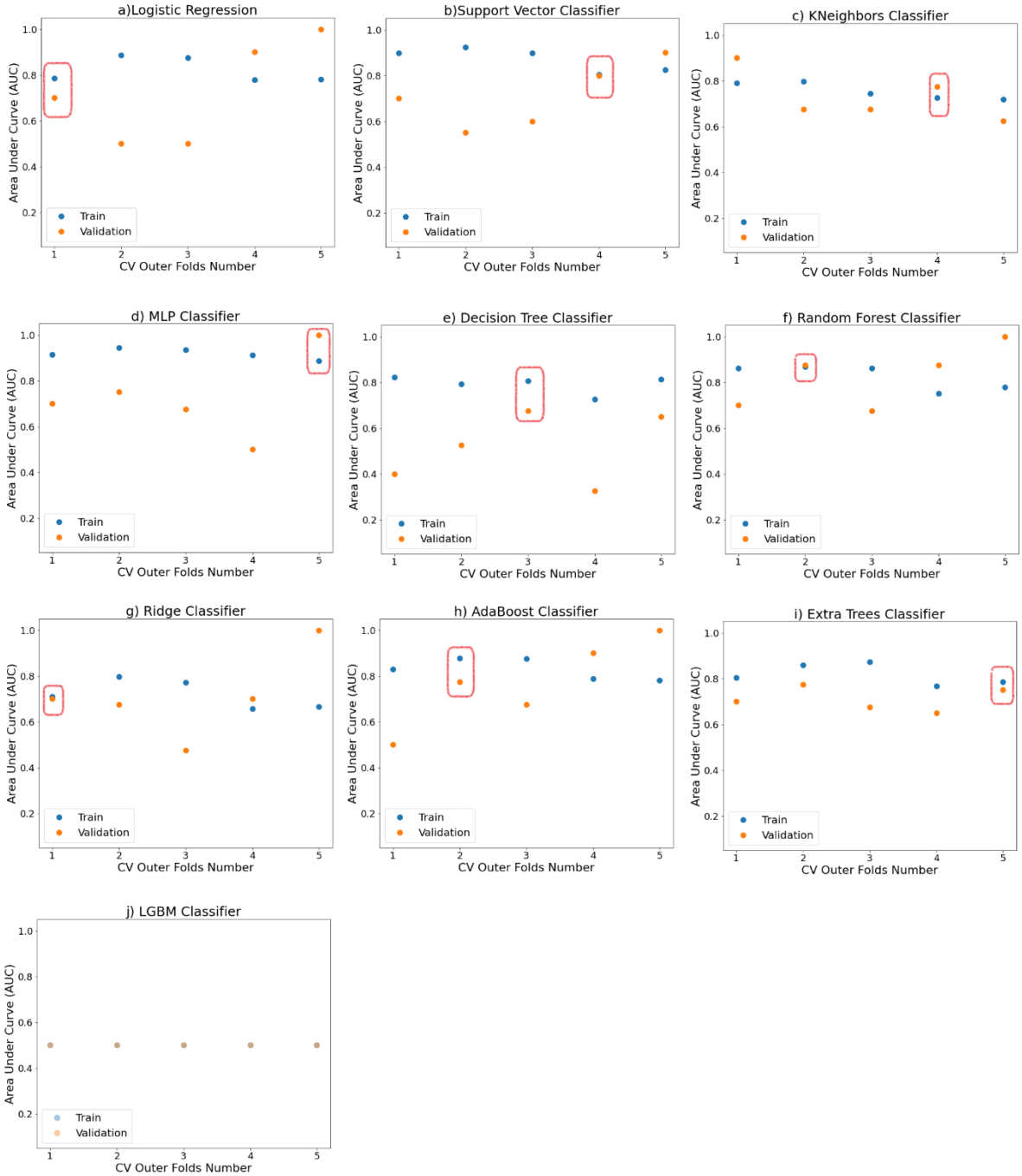


Figure 32. Train and validation curves of nested cross-validation in different outer folds using various classifiers in experiment 1 on the "initial dataset" for hippocampus. The red squares lineate the smallest difference between train and validation curves selected as the configuration of hyperparameters for the final prediction task.

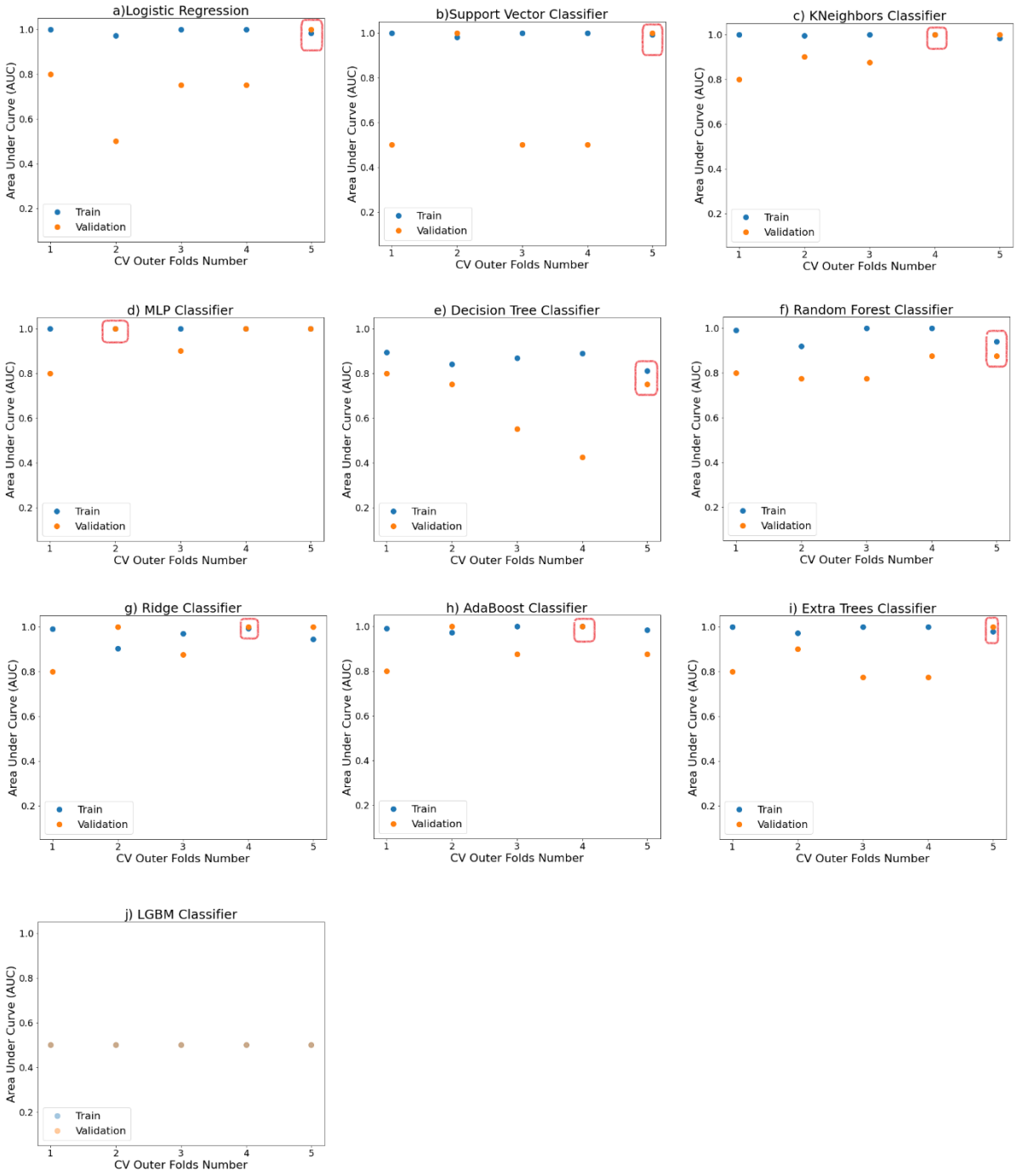


Figure 33. Train and validation curves of nested cross-validation in different outer folds using various classifiers in experiment 2 on the "expanded dataset" for hippocampus. The red squares specified the smallest difference between train and validation curves selected as the configuration of hyperparameters for the final prediction task.

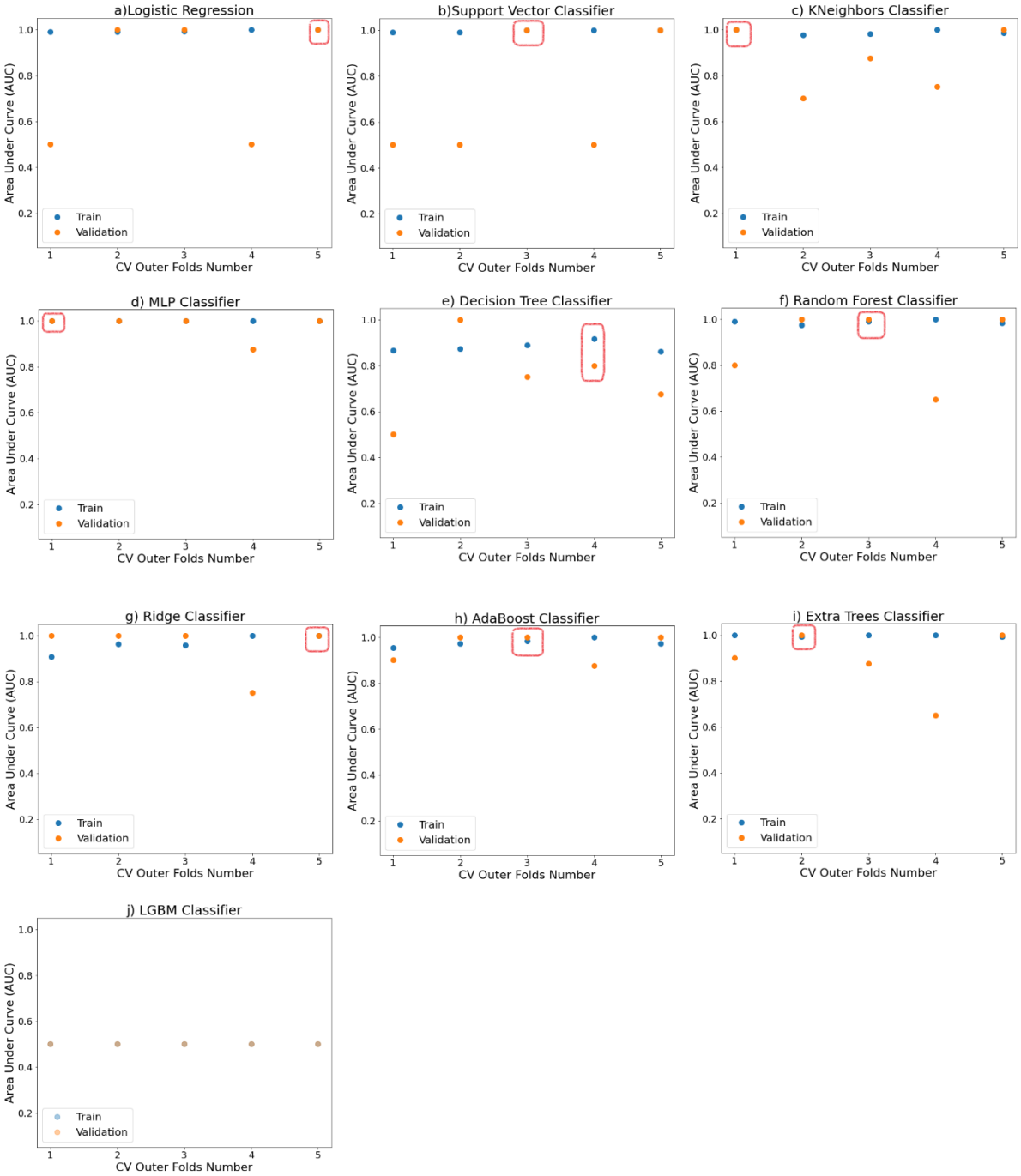


Figure 34. Train and validation curves of nested cross-validation in different outer folds using various classifiers in experiment 3 using the dataset obtained after removing highly correlated features from the "expanded dataset" for hippocampus. The red squares showed the smallest difference between train and validation curves selected as the configuration of hyperparameters for the final prediction task.

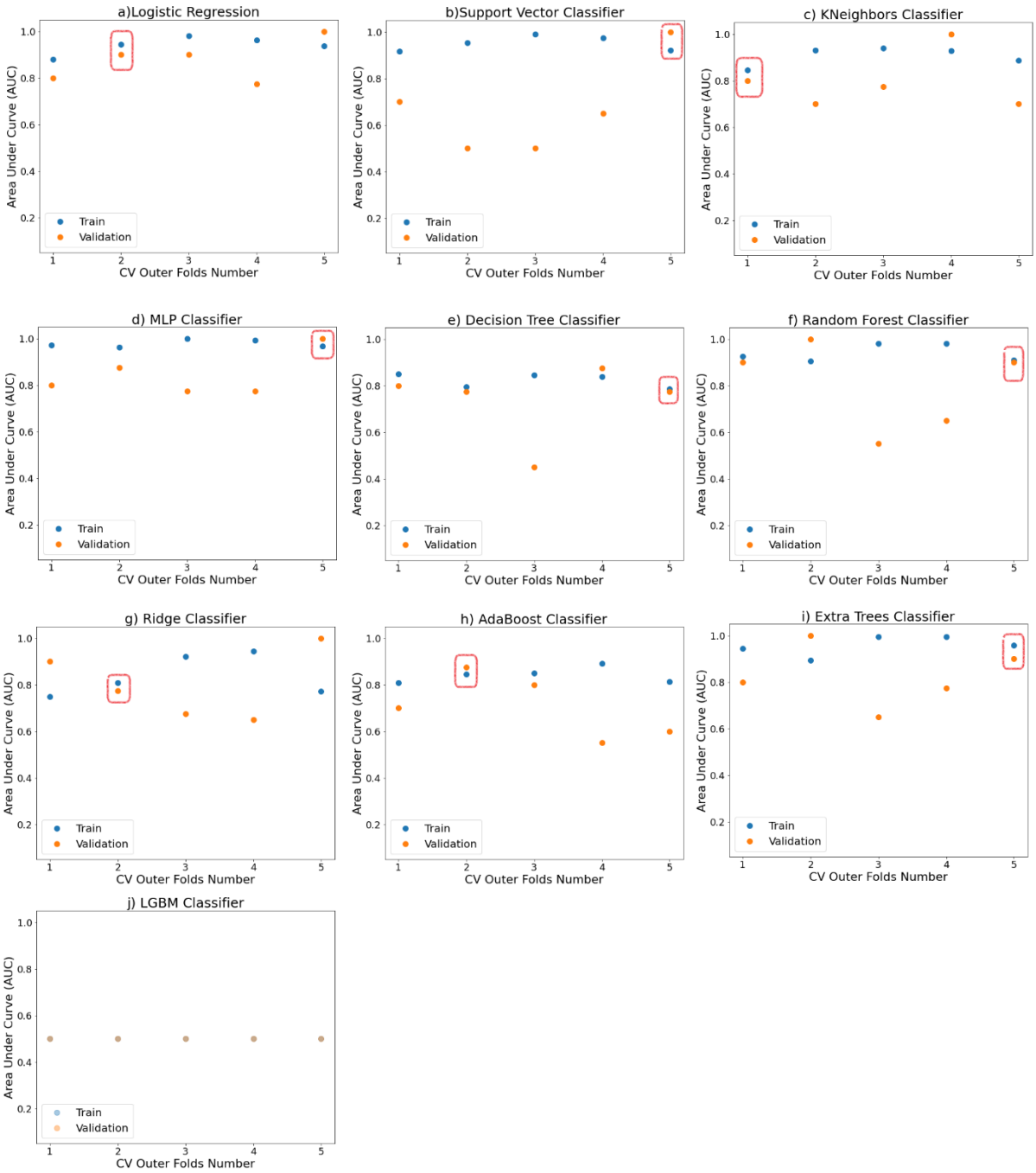


Figure 35. Train and validation curves of nested cross-validation in different outer folds using various classifiers in experiment 4 using the LBP dataset for hippocampus. The red squares lineate the smallest difference between train and validation curves selected as the configuration of hyperparameters for the final prediction task.

Comparing Classification Performance

The datasets containing selected feature sets for each experiment were described in the previous section. We used these reduced datasets as the input of classification tasks for the corresponding experiment. In this section, we elaborated the classification performance results per experiment in terms of the AUC metric.

Experiment 1 AUC Comparison

In Figure 36, the receiver operating curve (ROC) shows the performance of different classifiers in experiment 1 on the “initial dataset”. The highest AUC performance (85%) was obtained by AdaBoost, following by LR, SVC and MLP (83%). The poorest performance was related to the LGBM classifier (50%). Apart from LGBM, other classifiers had an AUC of around 70% or more. The Random Forest classifier with 76% AUC outperformed the Ridge classifier (70%), KNN and ET (69%).

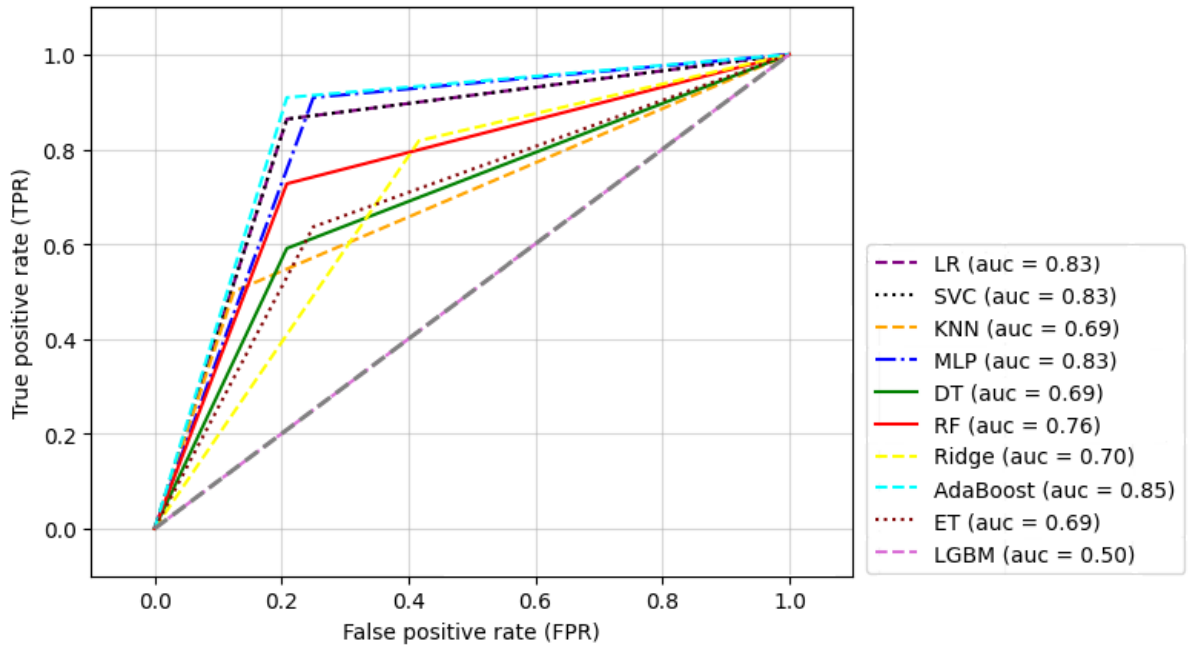


Figure 36. The ROC diagram shows the performance of classifiers based on AUC in experiment 1 on the “initial dataset” for the hippocampus.

Experiment 2 AUC Comparison

The AUC scores of performing several classification tasks in experiment 2 on the “expanded dataset” (Figure 37) shows that LR obtained the highest performance score (94%). Ridge and AdaBoost had a score of 93%. By the score of 91%, ET was superior to RF and MLP and KNN (all have a score of 87%), SVC 78% and DT 72%. As for the other experiments, the worst case was related to the LGBM classifier (50%).

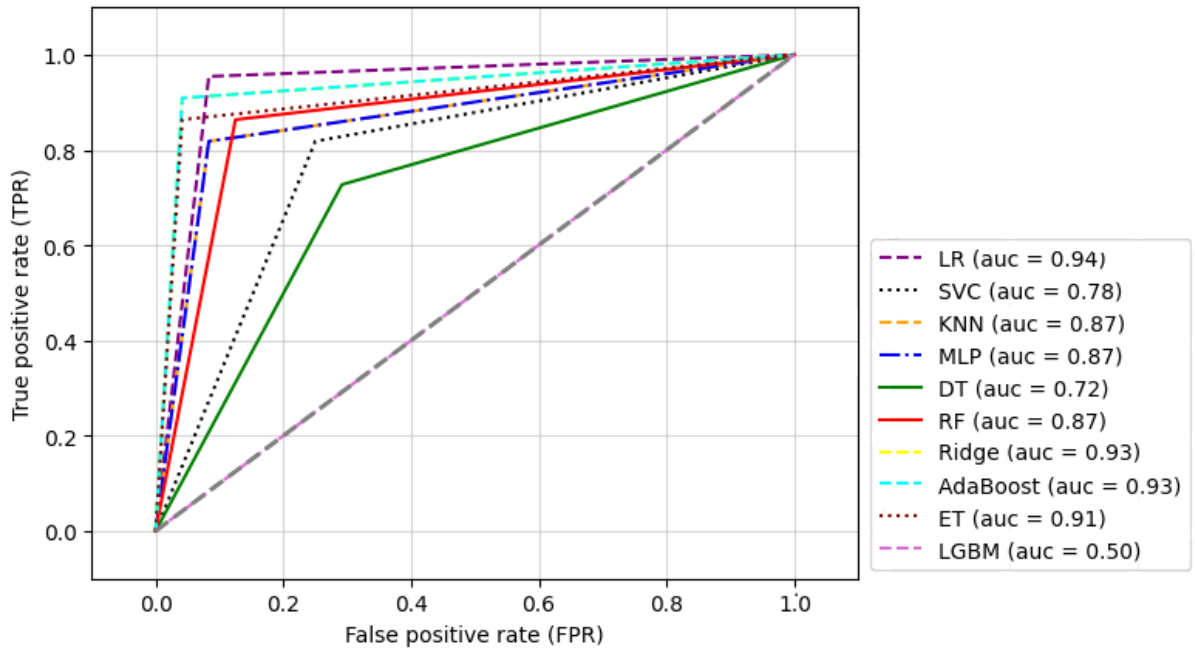


Figure 37. The performance of classifiers based on AUC in experiment 2 on the "expanded dataset" for the hippocampus.

Experiment 3 AUC Comparison

Figure 38 demonstrates the classification performance of various classifiers in experiment 3 on the dataset obtained after removing highly correlated features from the "expanded dataset".

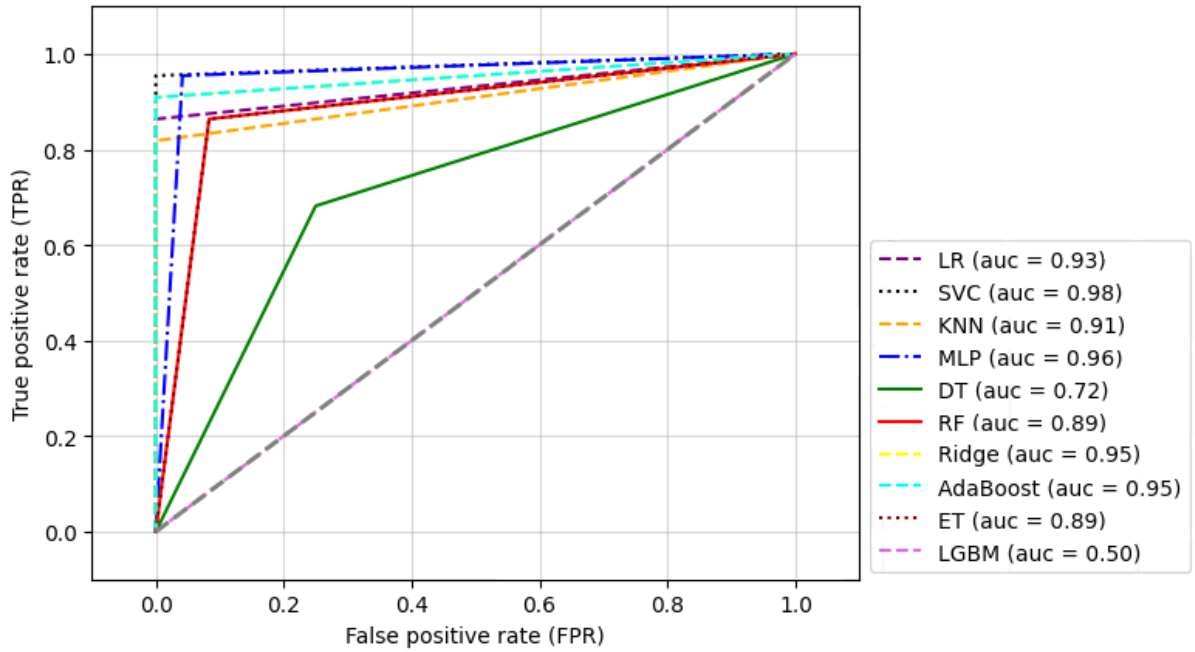


Figure 38. The ROC diagram illustrates the AUC score of classifiers' performances in experiment 3 for the hippocampus on the dataset obtained after removing highly correlated features from the "expanded dataset".

SVC, with an excellent performance score of 98%, had the best prediction score. MLP (96%), Ridge and AdaBoost with 95% also exhibited high prediction power. LR (93%), KNN (91%), RF and ET (89%) also had competitive results. Apart from the DT classifier with a score of 72% and LGBM with 50%, all other classifiers had scores around 90% or more.

Experiment 4 AUC Comparison

The ROC diagram in Figure 39 presents the prediction performance of different classification models in experiment 4 on the LBP dataset. KNN, with a score of 87%, had superior results in comparison to other classifiers. LR, SVC and AdaBoost had an AUC score of 85%, followed by ET (82%), Ridge (80%) and RF (76%). MLP had a score of 70%, and DT had a score of 68%. Again, the LGBM classifier score was the worst (50%).

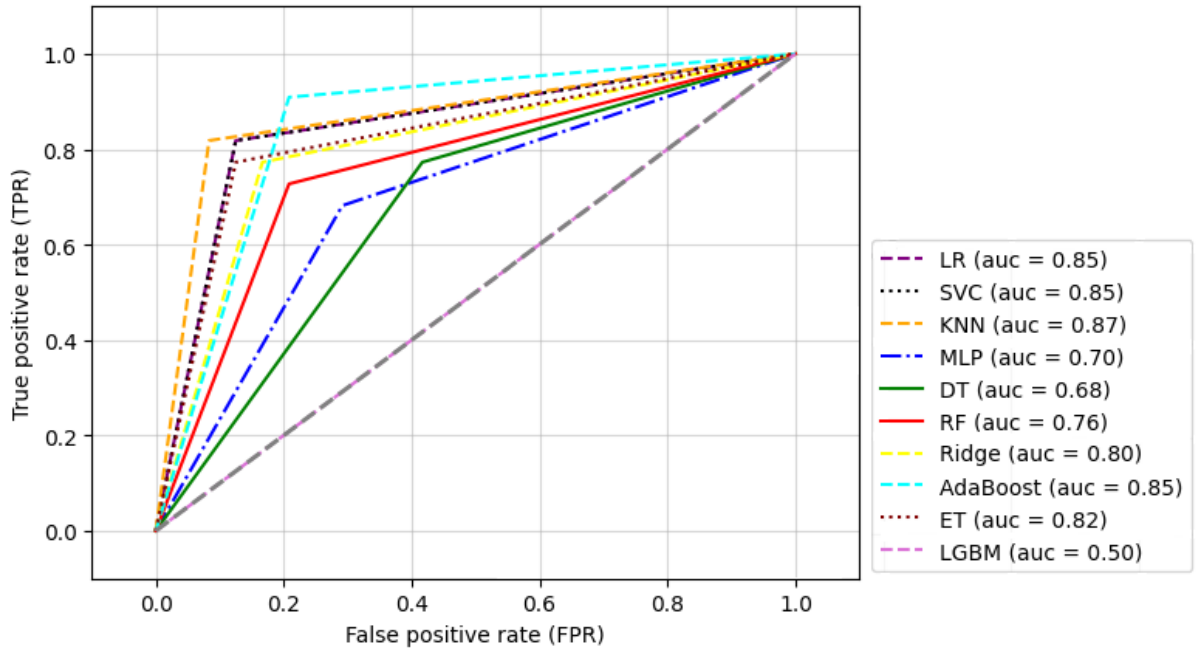


Figure 39. The diagram shows the AUC score of classifiers' prediction performances in experiment 4 on the LBP dataset.

4.1.3 Heatmap Comparison of the Experiments

Figure 40 shows a heatmap providing a performance comparison of the four experiments on datasets related to the hippocampus brain structure. We removed the result from LGBM as it had constantly poor results.

The scores varied from 68% to 98%. The highest score belonged to SVC in experiment 3 with 98%; in contrast, the lowest score (68%) was obtained by the DT classifier in experiment 4. From Figure 40, we observe that the DT classifier had a relatively consistent performance across all experiments since it varied from 68% to 72%.

Furthermore, one could see that the LR and AdaBoost showed acceptable results by having scores around 85% and more in all experiments. Apart from DT, the rest of the classifiers presented competitive results across all experiments.

It is clear that the performance scores in experiment 3 outperformed classifiers performance in experiment 1 and 4. The scores of classifiers in experiment 2 and 3 were very close to each other. Apart from SVC, classifiers in experiment 2 obtained higher scores than in experiment 1.

In experiment 3, by overlooking DT performance (72%), all classifiers had a high performance score of around 90% or more. Moreover, if we exclude DT (72%) and SVC (78%) in experiment 2, the rest of the classifiers showed scores higher than 85%. In experiments 1 and 4, the results showed acceptable prediction performance of around 70% and more.

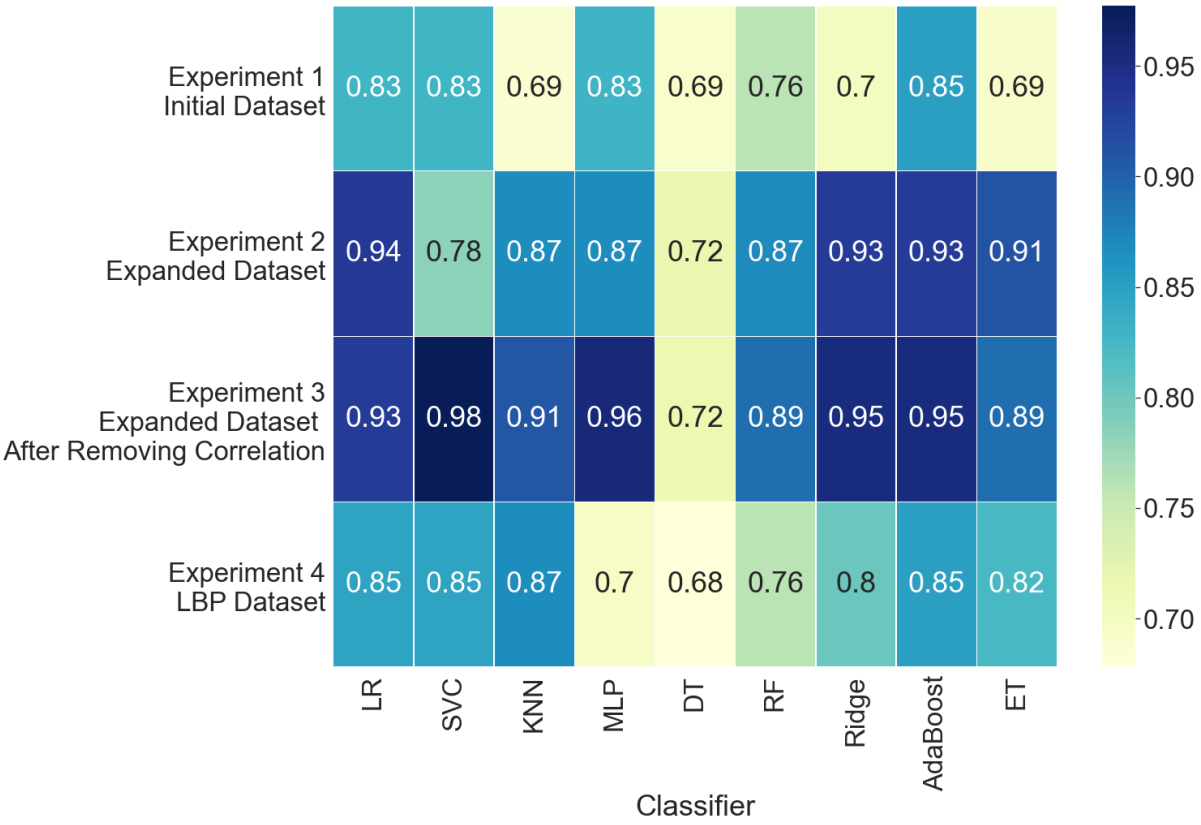


Figure 40. The overall heatmap compares the classifier performance based on the AUC score in four experiments on hippocampus datasets.

4.2 The Caudate

4.2.1 Selected Features using RENT

As mentioned in chapter 3, we performed RENT one time in the caudate experiments without generating any polynomial features. Therefore, the features in their original format were selected. In the continuation of this section, we described the selected features set attributes in every experiment on the caudate.

Selected Features in Experiment 1

In experiment 1 the “initial dataset” was used for analysis, consisting of shape and texture features of 128 and 64 grey level discretisation from the left and right side of the brain (see Figure 18).

In this experiment, RENT selected 19 features (from 328 radiomics features in the “initial dataset”) to be used as the input for modelling and evaluation. A list of selected features’ names for experiment 1 is provided in Table 9.

Table 9. Selected features attribute in experiment 1 on the caudate using the “initial dataset”. 128-bin and 64-bin refer to the texture features with 128 and 64 grey level discretisation. Right or Left indicate the right or left side of the brain, respectively.

	Feature Name	Side	Feature Type
1	Shape_Flatness_left	Left	Shape
2	Shape_MinorAxisLength_right	Right	Shape
3	128_ClusterProminence_d_1_left	Left	128-bin
4	128_MaximumProbability_d_1_right	Right	128-bin
5	128_Complexity_right	Right	128-bin
6	128_LowGrayLevelZoneEmphasis_left	Left	128-bin
7	128_SmallAreaLowGrayLevelEmphasis_left	Left	128-bin
8	128_GrayLevelNonUniformity_left.1	Left	128-bin
9	128_GrayLevelNonUniformity_left.2	Left	128-bin
10	128_SmallDependenceLowGrayLevelEmphasis_left	Left	128-bin
11	64_ClusterProminence_d_1_left	Left	64-bin
12	64_HighGrayLevelZoneEmphasis_right	Right	64-bin
13	64_SmallAreaHighGrayLevelEmphasis_right	Right	64-bin
14	64_HighGrayLevelRunEmphasis_right	Right	64-bin
15	64_LongRunLowGrayLevelEmphasis_right	Right	64-bin
16	64_LowGrayLevelRunEmphasis_right	Right	64-bin
17	64_ShortRunLowGrayLevelEmphasis_right	Right	64-bin
18	64_DependenceEntropy_right	Right	64-bin
19	64_LowGrayLevelEmphasis_right	Right	64-bin

Figure 41a illustrates that texture features of the 128-bin type were selected more than other feature types (128-bin type 47% versus 42% from 64-bin type and 11% from shape feature category). Figure 41b shows that 58% of the selected features were from the right side of the brain.

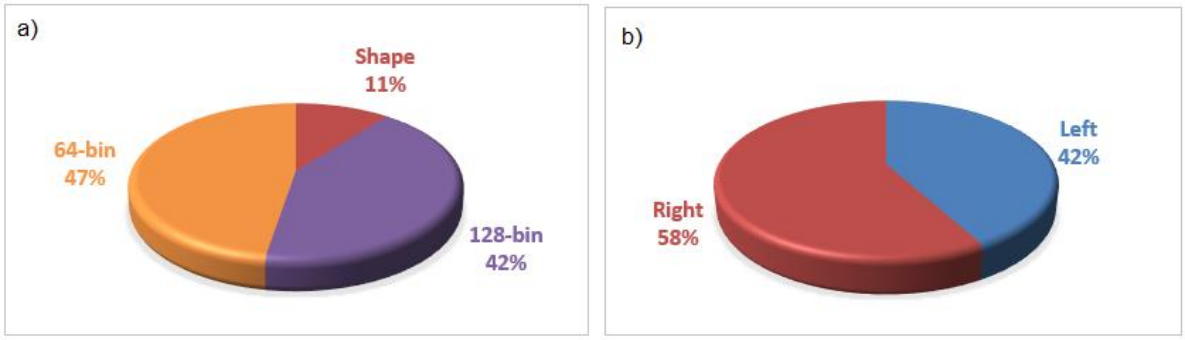


Figure 41. Pie charts show the distribution of selected features for the caudate from "initial dataset" in experiment 1. a) the distribution of selected features based on the feature type. 128-bin and 64-bin refer to the texture features with 128 and 64 grey level discretisation. Shape denotes the shape features. b) the distribution of features selected from the left or right sides of the brain.

Selected Features in Experiment 2

As mentioned in chapter 3, experiment 2 utilised the “expanded dataset” containing LBP features plus the shape feature and texture features of 128 and 64 grey scale discretisation (see Figure 21).

17 features (from 348 radiomics features in the "expanded dataset”) were selected by running RENT. We used these selected features for constituting the final reduced dataset. Subsequently, this reduced dataset (with 17 features) was used to model and evaluate experiment 2. Selected features’ names for experiment 2 is listed in Table 10.

Most of the selected features were texture features, the 64-bin (47%) plus the 128-bin (6%) (Figure 42a). The LBP features constituted 47% of the selected features in the reduced dataset. None of the shape features was selected. 59% of the selected features were from the right side of the brain (Figure 42b).

Table 10. Selected features attribute in experiment 2 on the caudate using the "expanded dataset". LBP corresponds to LBP features. 128-bin and 64-bin refer to the texture features with, respectively, 128 and 64 grey level discretisation. Right or Left indicate the right or left side of the brain, respectively.

	Feature Name	Side	Feature Type
1	128_MCC_d_1_right	Right	128-bin
2	64_LowGrayLevelZoneEmphasis_left	Left	64-bin
3	64_GrayLevelVariance_right	Right	64-bin
4	64_SmallAreaHighGrayLevelEmphasis_right	Right	64-bin
5	64_LongRunLowGrayLevelEmphasis_right	Right	64-bin
6	64_LowGrayLevelRunEmphasis_right	Right	64-bin
7	64_ShortRunLowGrayLevelEmphasis_right	Right	64-bin
8	64_LargeDependenceLowGrayLevelEmphasis_right	Right	64-bin
9	64_LowGrayLevelEmphasis_right	Right	64-bin
10	LBP_111_left	Left	LBP
11	LBP_030_left	Left	LBP
12	LBP_021_left	Left	LBP
13	LBP_300_left	Left	LBP
14	LBP_102_left	Left	LBP
15	LBP_003_left	Left	LBP
16	LBP_012_right	Right	LBP
17	LBP_003_right	Right	LBP

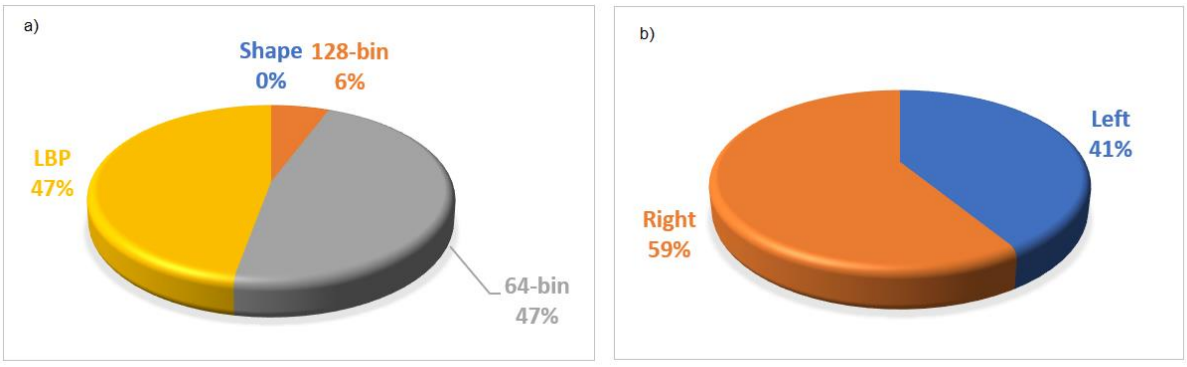


Figure 42. Pie charts show the characteristics of selected features from the "expanded dataset" in experiment 2 for the caudate. a) the distribution of selected features based on the feature type. LBP corresponds to the LBP features. 128-bin and 64-bin refer to the texture features with 128 and 64 grey level discretisation. Shape denotes the shape features. b) the distribution of features selected from the left or right sides of the brain.

Selected Features and Feature Correlation in Experiment 3

Features Collinearity

Figure 43 shows the heatmap of correlations between the 17 features selected by RENT from the “expanded dataset” in experiment 2. It is clear, the features with correlation strictly greater than 70% (in the selected features by RENT in experiment 2) were:

- 64_LowGrayLevelRunEmphasis_right
- 64_ShortRunLowGrayLevelEmphasis_right
- 64_LowGrayLevelEmphasis_right
- 64_LongRunLowGrayLevelEmphasis_right

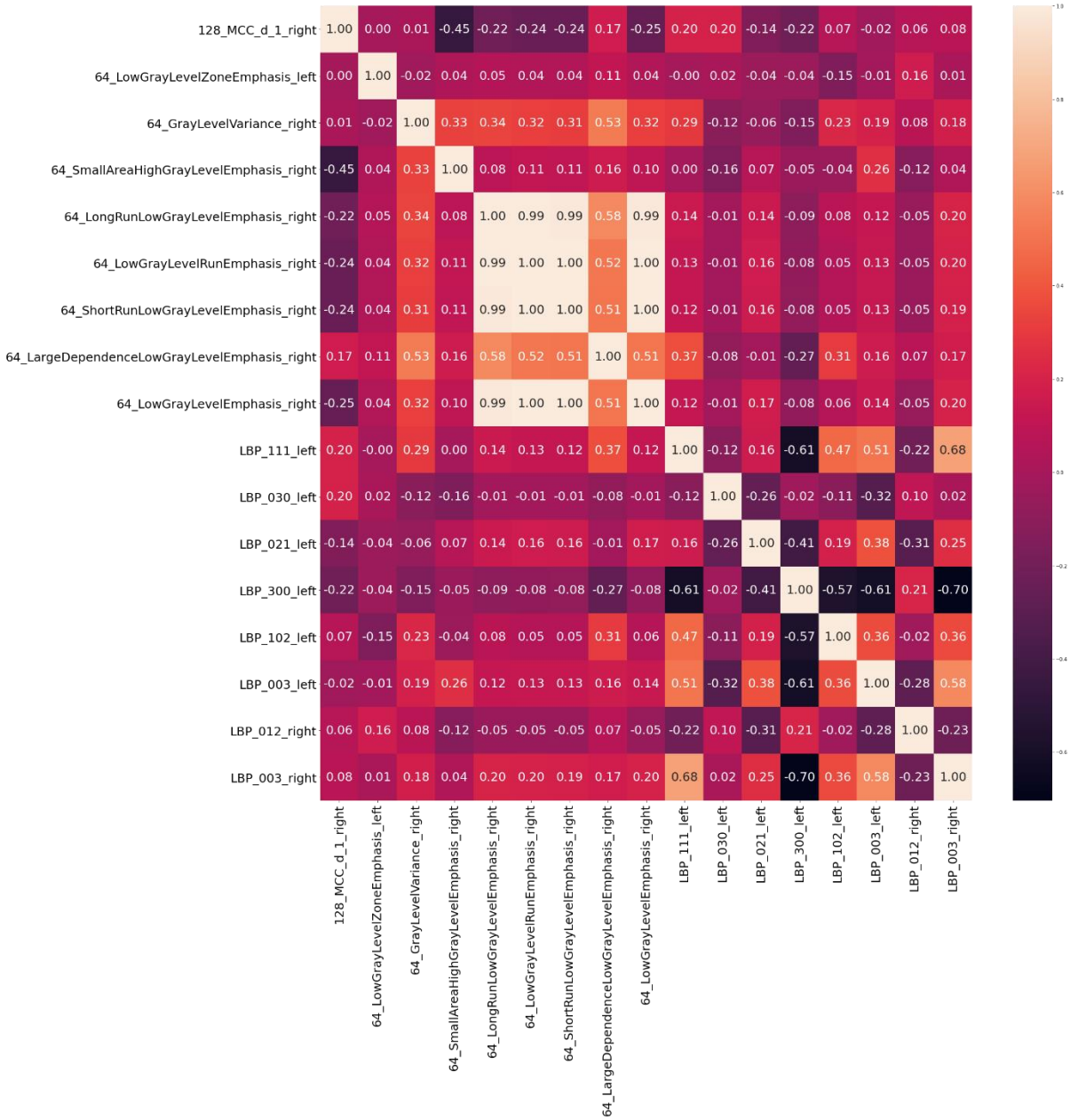


Figure 43. The correlation heatmap of features selected by RENT in experiment 2 for the caudate. The values show the Spearman Correlation Coefficient between pairs of features.

Selected Features in Experiment 3

In this experiment we were left with 170 features (out of the 348 features) after removing one of the features from pairs (having SCC above 95%) in the “expanded dataset”.

Figure 44 illustrates the distribution of features in the dataset obtained after removing highly correlated features. The LBP features comprised 12% of this dataset compared to shape feature 17%, texture feature 128-bin 42% and 64-bin 29% (Figure 44a). It should be pointed out that all the LBP features were included in this reduced dataset, which means they did not correlate highly.

The features set contained 46% features from the right side of the brain versus 54% from the left side (Figure 44b).

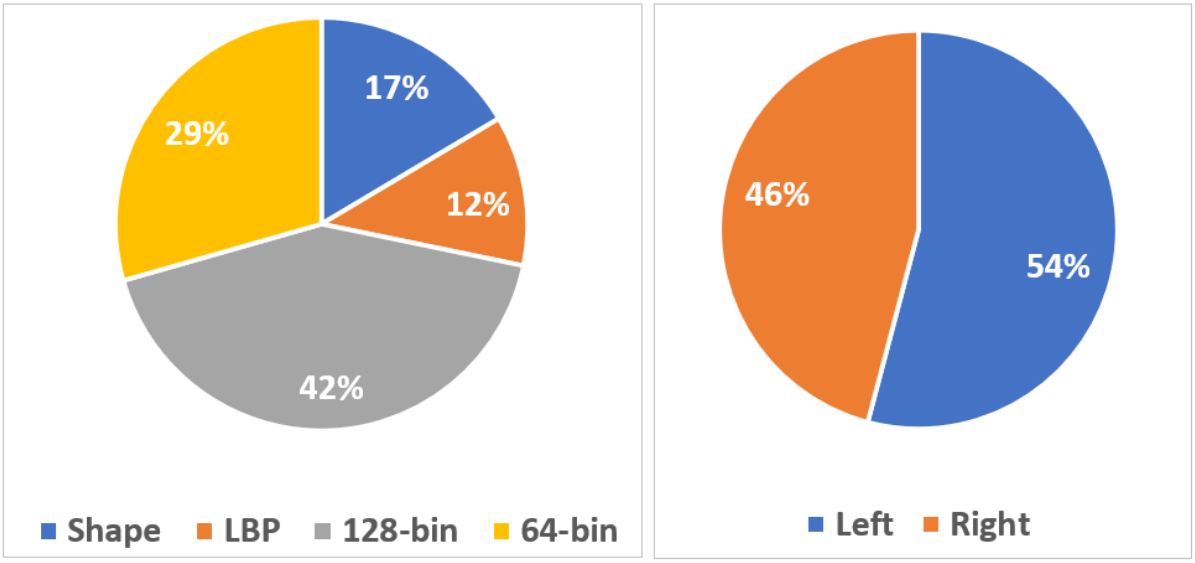


Figure 44. Pie charts show the distribution of various radiomics features in the dataset obtained after removing highly correlated features from the "expanded dataset" in experiment 3 for the caudate. a) the distribution of features based on the feature type. 128-bin and 64-bin refer to the texture features, respectively, 128 and 64 grey level discretisation. Shape denotes the shape features. LBP corresponds to LBP features. b) the distribution of features from the left or right sides of the brain.

In experiment 3, RENT was performed on this dataset with 170 radiomics features (after removing highly correlated features). The reduced dataset contained 13 features was used as the final selected features for modelling and evaluation in experiment 3. The list of selected features’ names and characteristics for experiment 3 is presented in Table 11.

Table 11. Selected features attribute in experiment 3 for the caudate. LBP corresponds to LBP features. 128-bin and 64-bin refer to the texture features with 128 and 64 grey level discretisation, respectively. Right or Left indicate the right or left side of the brain.

	Feature Name	Side	Feature Type
1	128_MCC_d_1_right	Right	128-bin
2	64_LowGrayLevelZoneEmphasis_left	Left	64-bin
3	64_GrayLevelVariance_right	Right	64-bin
4	64_SmallAreaHighGrayLevelEmphasis_right	Right	64-bin
5	64_LongRunLowGrayLevelEmphasis_right	Right	64-bin
6	64_LargeDependenceLowGrayLevelEmphasis_right	Right	64-bin
7	LBP_030_left	Left	LBP
8	LBP_021_left	Left	LBP
9	LBP_300_left	Left	LBP
10	LBP_102_left	Left	LBP
11	LBP_003_left	Right	LBP
12	LBP_012_right	Right	LBP
13	LBP_003_right	Right	LBP

In experiment 3, the LBP features had the highest selection rate (54%) (Figure 45a). The 64-bin texture features comprised 38% of selected features, versus 128-bin features were 8% of selected features. In this experiment, none of the selected features was shape features. Figure 45b shows that were mostly from the right side of the brain (62%).

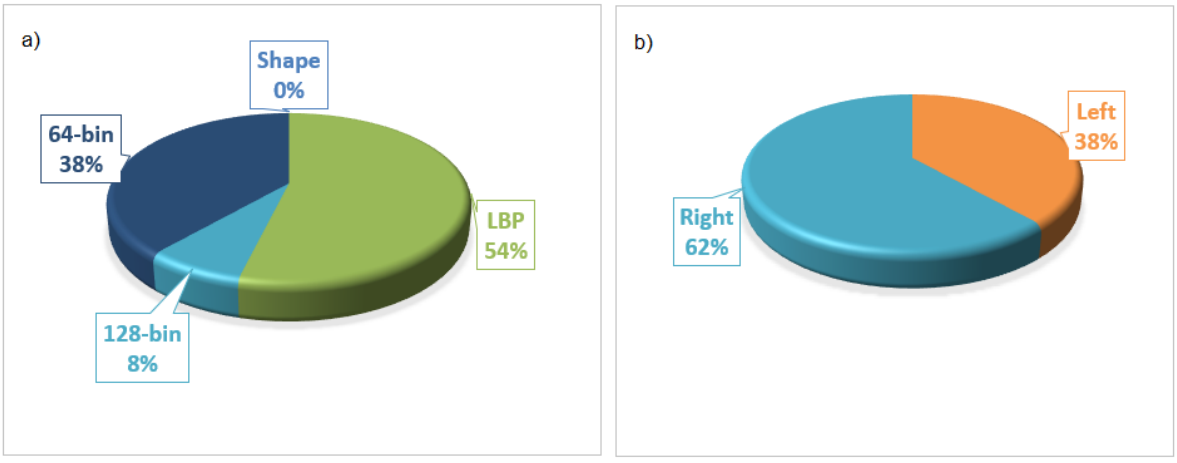


Figure 45. Pie charts show the characteristics of selected features from the dataset obtained after removing highly correlated features from the "expanded dataset" in experiment 3 for the caudate. a) the distribution of selected features based on the feature type. LBP corresponds to the LBP features. 128-bin and 64-bin refer to the texture features with, respectively, 128 and 64 grey level discretisation. Shape denotes the shape features. b) the distribution of features selected from the left or right sides of the brain.

4.2.2 Heatmap Comparison of the Experiments

The overall heatmap of the caudate experiments is shown in Figure 46. Because of very poor performance, LGBM was excluded from this diagram. The scores ranges from 50% to 100%. The highest score obtained by LR and AdaBoost in experiment 2 as well as MLP and AdaBoost in experiment 3 with a score of 100%; in contrast, the lowest score (50%) was obtained by the SVC classifier in experiments 2 and 3. From Figure 46, we observe that the LR classifier had performance scores with high variations across all experiments since it varied from 57% to 100%. Similarly, SVC scores dropped strangely in experiments 2 and 3 (50%) while it had acceptable scores in experiments 1 and 4 (70% and 85%, respectively). Furthermore, one could see that the rest of the classifiers showed promising results by having scores above 80% in experiments 2, 3 and 4.

The performance scores in experiment 3 (apart from SVC and LR) outperformed experiments 1 and 4. Classifiers in experiment 2, except for SVC, outperformed classifiers' scores in experiment 1.

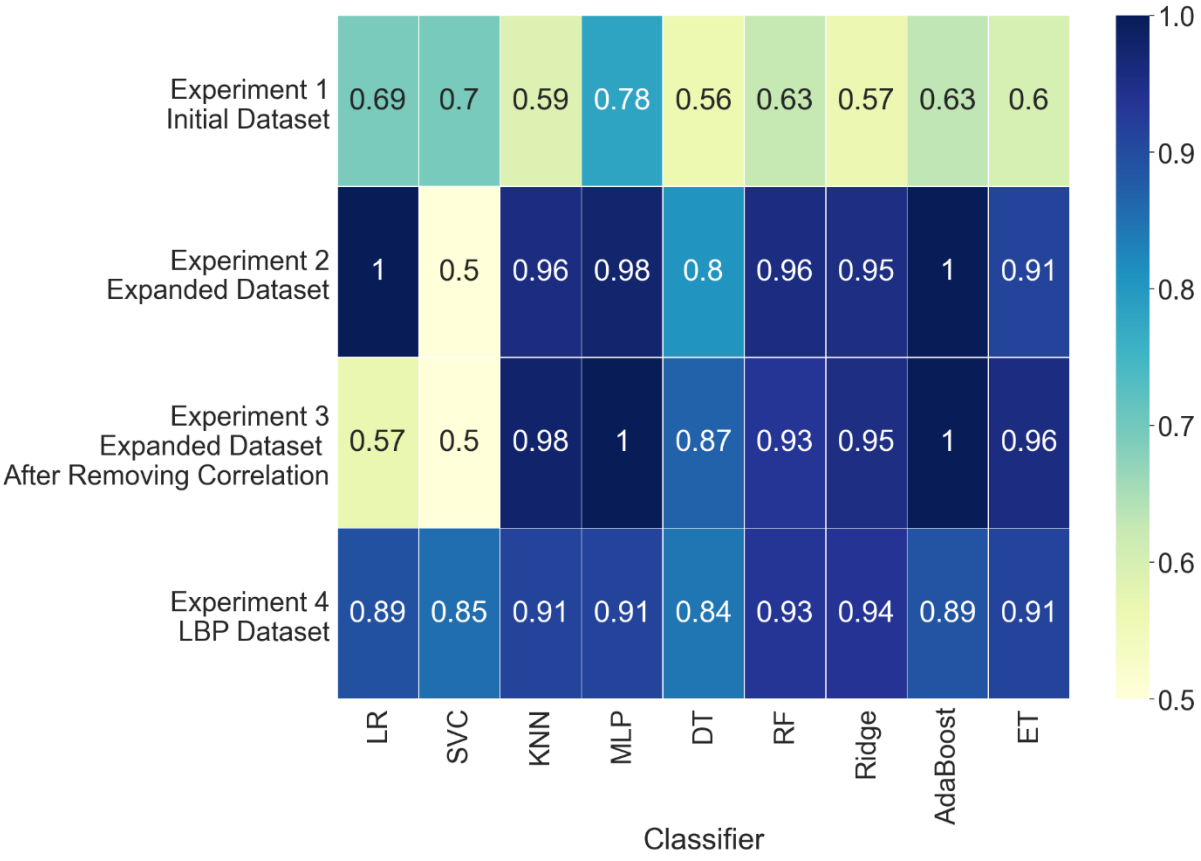


Figure 46. The overall heatmap shows the comparison between the performance of classifiers based on the AUC score in four experiments on caudate datasets.

4.3 The Putamen

4.3.1 Selected Features using RENT

For the putamen's experiments, RENT was performed once without generating any polynomial features.

Selected Features in Experiment 1

In experiment 1, by applying RENT on the "initial dataset" (see Figure 18), we obtained a reduced dataset with 16 features out of 328 radiomics features. The modelling and evaluation tasks in experiment 1 was done on this reduced dataset. The names of selected features are provided in Table 12.

Table 12. Selected features attribute in experiments for the putamen on the "initial dataset". Shape denotes shape features. 128-bin and 64-bin refer to the texture features with, respectively, 128 and 64 grey level discretisation. Right or Left indicate the right or left side of the brain.

	Feature Name	Side	Feature Type
1	Shape_Maximum3DDiameter_left	Left	Shape
2	Shape_Sphericity_right	Right	Shape
3	Shape_SurfaceVolumeRatio_right	Right	Shape
4	128_GrayLevelVariance_right	Right	128-bin
5	128_DependenceEntropy_right	Right	128-bin
6	128_DependenceVariance_right	Right	128-bin
7	64_GrayLevelNonUniformityNormalized_right	Right	64-bin
8	64_GrayLevelVariance_right	Right	64-bin
9	64_LongRunLowGrayLevelEmphasis_left	Left	64-bin
10	64_LowGrayLevelRunEmphasis_left	Left	64-bin
11	64_ShortRunLowGrayLevelEmphasis_left	Left	64-bin
12	64_LongRunLowGrayLevelEmphasis_right	Right	64-bin
13	64_LargeDependenceLowGrayLevelEmphasis_left	Left	64-bin
14	64_LowGrayLevelEmphasis_left	Left	64-bin
15	64_LargeDependenceLowGrayLevelEmphasis_right	Right	64-bin
16	64_LowGrayLevelEmphasis_right	Right	64-bin

Figure 47 illustrates that most of the selected features were texture features of the 64-bin type (62%) versus 19% from the 128-bin type. 19% of selected features were from the shape features category (Figure 47a). And 63% of the selected features were from the right side of the brain (Figure 47b).

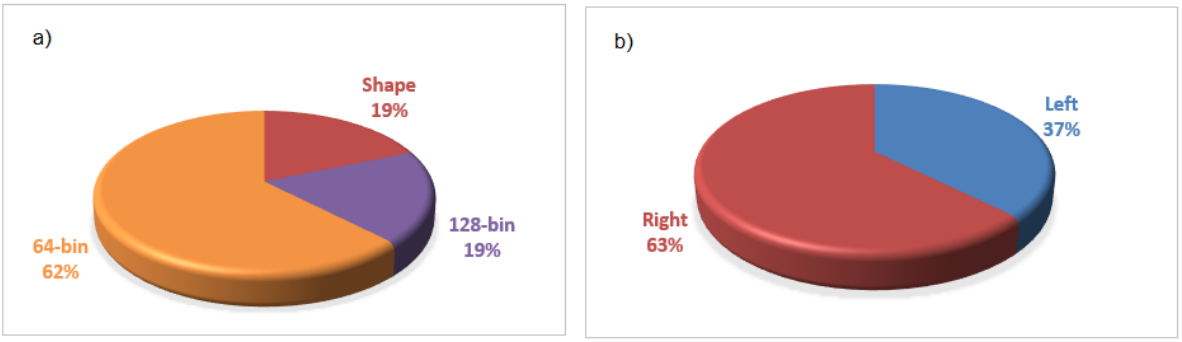


Figure 47. Pie charts show the distribution of selected features for the putamen from "initial dataset" in experiment 1. a) the distribution of selected features based on the feature type. 128-bin and 64-bin refer to the texture features with, respectively, 128 and 64 grey level discretisation. Shape denotes the shape features. b) the distribution of features selected from the left or right sides of the brain.

Selected Features in Experiment 2

For the “expanded dataset” containing LBP features plus the shape feature and texture features of 128 and 64 grey scale discretisation (see Figure 21), 15 features out of 348 radiomics features were selected by RENT. These 15 features, listed in Table 13, constituted the reduced dataset used for modelling in experiment 2.

Most of the selected features (Figure 48a) were LBP features comprising 46% of the features in the reduced dataset. Both groups of texture features (128-bin and 64-bin) had a selection rate of 27%. On the other hand, none of the shape features was selected. Figure 48b shows that features from the left side of the brain had the majority (67%).

Table 13. Selected features attribute in experiment 2 for the putamen on the "expanded dataset". LBP corresponds to LBP features. 128-bin and 64-bin refer to the texture features with 128 and 64 grey level discretisation, respectively. Right or Left indicate the right or left side of the brain.

	Feature Name	Side	Feature Type
1	128_MCC_d_1_left	Left	128-bin
2	128_LongRunLowGrayLevelEmphasis_left	Left	128-bin
3	128_LargeDependenceLowGrayLevelEmphasis_left	Left	128-bin
4	128_LowGrayLevelEmphasis_left	Left	128-bin
5	64_LowGrayLevelRunEmphasis_left	Left	64-bin
6	64_ShortRunLowGrayLevelEmphasis_left	Left	64-bin
7	64_LowGrayLevelEmphasis_left	Left	64-bin
8	64_SmallDependenceLowGrayLevelEmphasis_left	Left	64-bin
9	LBP_120_left	Left	LBP
10	LBP_102_left	Left	LBP
11	LBP_021_right	Right	LBP
12	LBP_030_right	Right	LBP
13	LBP_201_right	Right	LBP
14	LBP_012_right	Right	LBP
15	LBP_003_right	Right	LBP

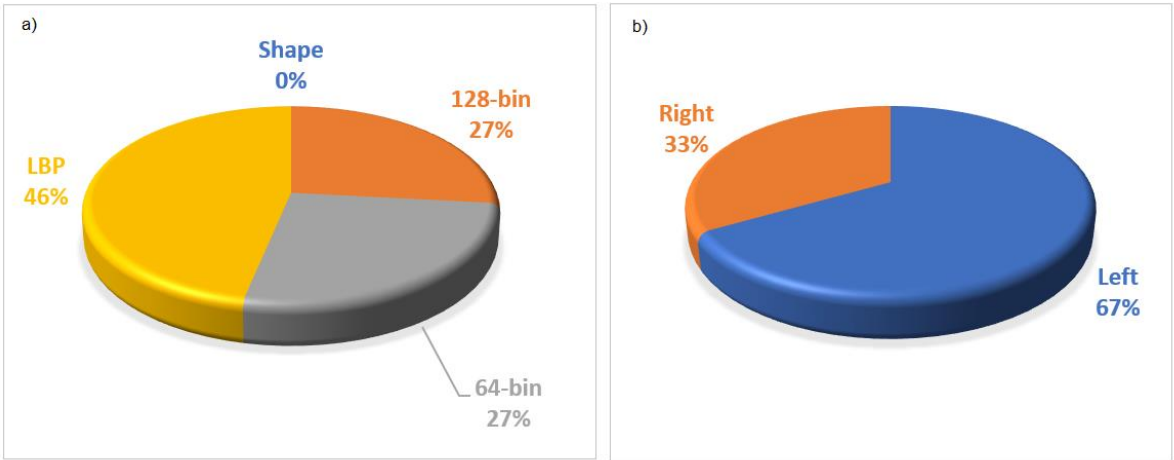


Figure 48. Pie charts show the characteristics of selected features from the "expanded dataset" in experiment 2 for the putamen. a) the distribution of selected features based on the feature type. LBP corresponds to the LBP features. 128-bin and 64-bin refer to the texture features with, respectively, 128 and 64 grey level discretisation. Shape denotes the shape features. b) the distribution of features selected from the left or right sides of the brain.

Selected Features and Feature Correlation in Experiment 3

Features Collinearity

Figure 49 shows the heatmap obtained for correlations between the 15 features (from 348 radiomics features in the "expanded dataset”), selected by RENT in experiment 2). Features with correlation above 70% in the selected features were:

- 128_LargeDependenceLowGrayLevelEmphasis_left
- 128_LowGrayLevelEmphasis_left
- 128_LongRunLowGrayLevelEmphasis_left
- 64_LowGrayLevelRunEmphasis_left
- 64_ShortRunLowGrayLevelEmphasis_left
- 64_LowGrayLevelEmphasis_left
- 64_SmallDependenceLowGrayLevelEmphasis_left

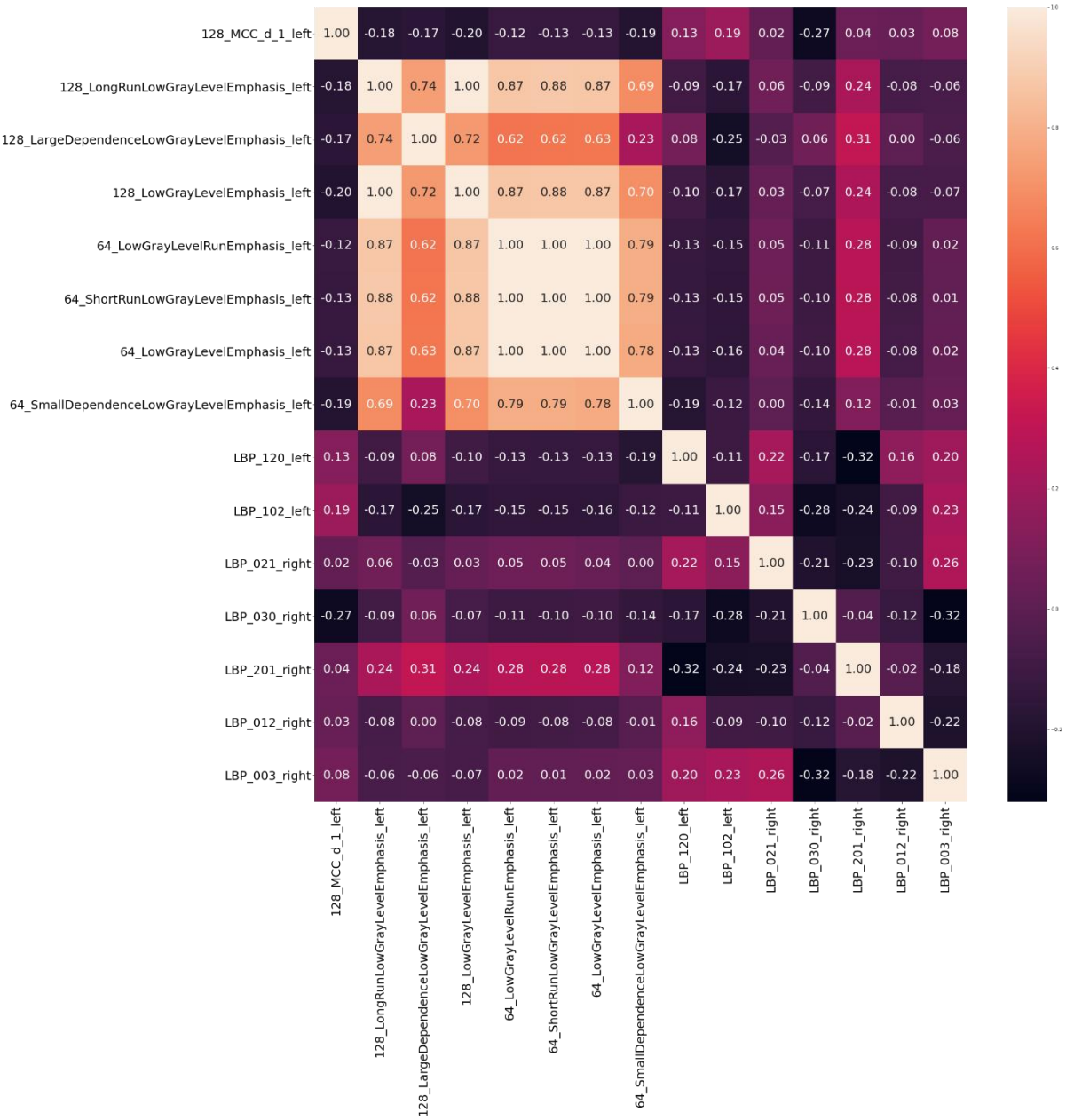


Figure 49. The correlation heatmap of features selected by RENT in experiment 2 for the putamen. The values show the Spearman Correlation Coefficient between pairs of features.

Selected Features in Experiment 3

We constructed a reduced dataset with172 features (out of 348 features) by removing one of the features from pairs having above 95% SCC in the “expanded dataset”. The LBP features comprised 12% of these features (Figure 50) in comparison to shape feature (14%), texture feature 128-bin (47%) and 64-bin (27%). All the LBP features were included in this dataset, and none of them was removed because of high correlation. The dataset contained 51% features from the right side of the brain (Figure 50b).

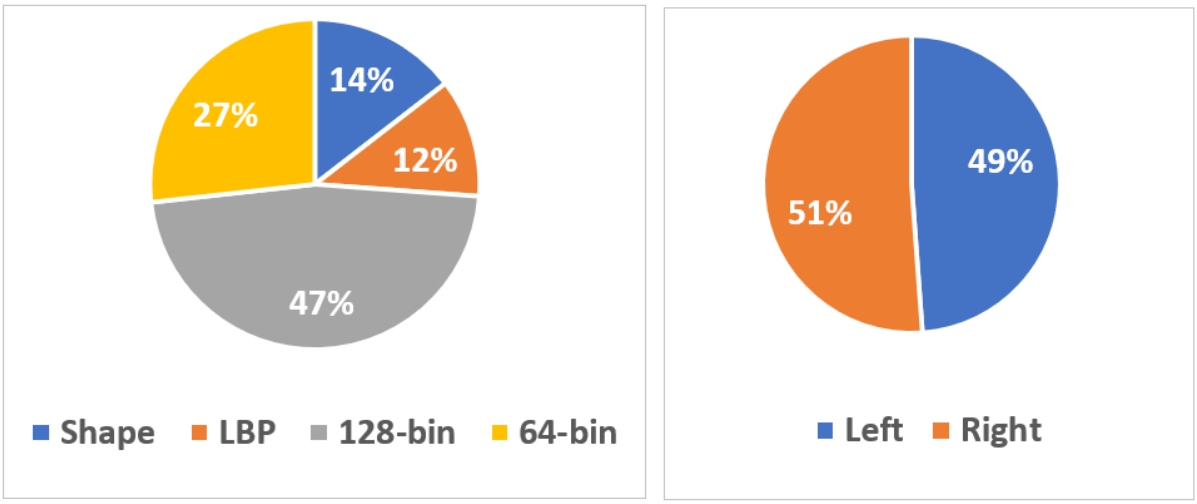


Figure 50. Pie charts show the distribution of various radiomics features in the dataset obtained after removing highly correlated features from the "expanded dataset" in experiment 3 for the putamen. a) the distribution of features based on the feature type. 128-bin and 64-bin refer to the texture features, respectively, 128 and 64 grey level discretisation. Shape denotes the shape features. LBP corresponds to LBP features. b) the distribution of features selected from the left or right sides of the brain.

In experiment 3, we performed RENT on the dataset with 172 features constituting a reduced dataset with 14 features (Table 14). This reduced dataset was used as the final selected features for modelling and evaluation in experiment 3.

Table 14. Selected features attribute in experiment 3 for the putamen. LBP corresponds to LBP features. 128-bin and 64-bin refer to the texture features with 128 and 64 grey level discretisation, respectively. Right or Left indicate the right or left side of the brain, respectively.

	Feature Name	Side	Feature Type
1	128_MCC_d_1_left	Left	128-bin
2	128_LongRunLowGrayLevelEmphasis_left	Left	128-bin
3	128_LargeDependenceLowGrayLevelEmphasis_left	Left	128-bin
4	64_LowGrayLevelZoneEmphasis_left	Left	64-bin
5	64_SmallAreaLowGrayLevelEmphasis_left	Left	64-bin
6	64_LongRunLowGrayLevelEmphasis_left	Left	64-bin
7	LBP_111_left	Left	LBP
8	LBP_120_left	Left	LBP
9	LBP_102_left	Left	LBP
10	LBP_021_right	Right	LBP
11	LBP_030_right	Right	LBP
12	LBP_201_right	Right	LBP
13	LBP_012_right	Right	LBP
14	LBP_003_right	Right	LBP

Figure 51 shows that the LBP features had the highest selection rate (57%). 21% of selected features were 64-bin features versus 22% of the 128-bin features. In this experiment, none of the selected features was shape features. 64% of the selected features were from the left side of the brain (Figure 51b).

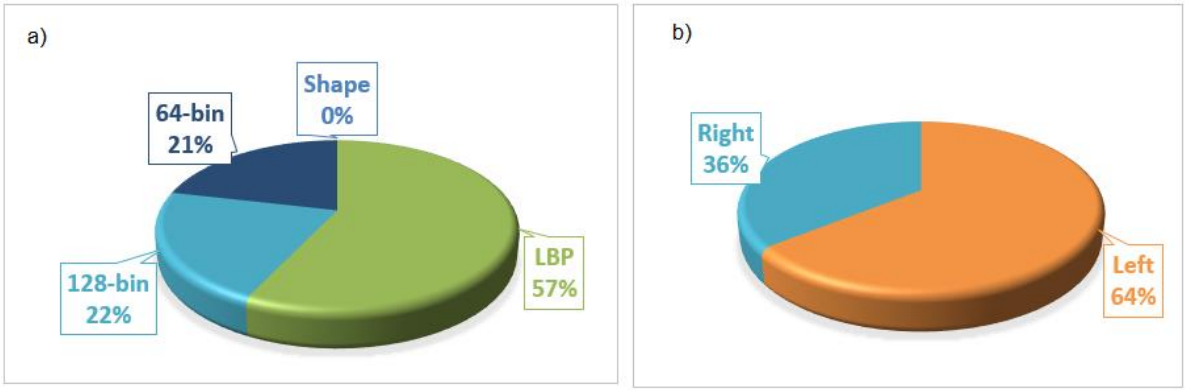


Figure 51. Pie charts show the characteristics of selected features from the dataset obtained after removing highly correlated features from the "expanded dataset" in experiment 3 for the putamen. a) the distribution of selected features based on the feature type. LBP corresponds to the LBP features. 128-bin and 64-bin refer to the texture features with, respectively, 128 and 64 grey level discretisation. Shape denotes the shape features. b) the distribution of features selected from the left or right sides of the brain.

4.3.2 Heatmap Comparison of the Experiments

Figure 52 shows the overall heatmap of the putamen experiments. The result from LGBM was excluded because of its poor performance. The scores varied from 61% to 96%. LR achieved the highest score in experiment 3 (96%); in contrast, the AdaBoost classifier in experiment 1 and the DT classifier in experiment 4 had the lowest score (61%). The scores in experiment 3 outperformed the scores of experiments 1 and 4. All classifiers in experiment 2 outperformed the classifiers' scores in experiment 1. Furthermore, the performance of classifiers in experiments 2 and 3 are very close to each other (except for DT and KNN).

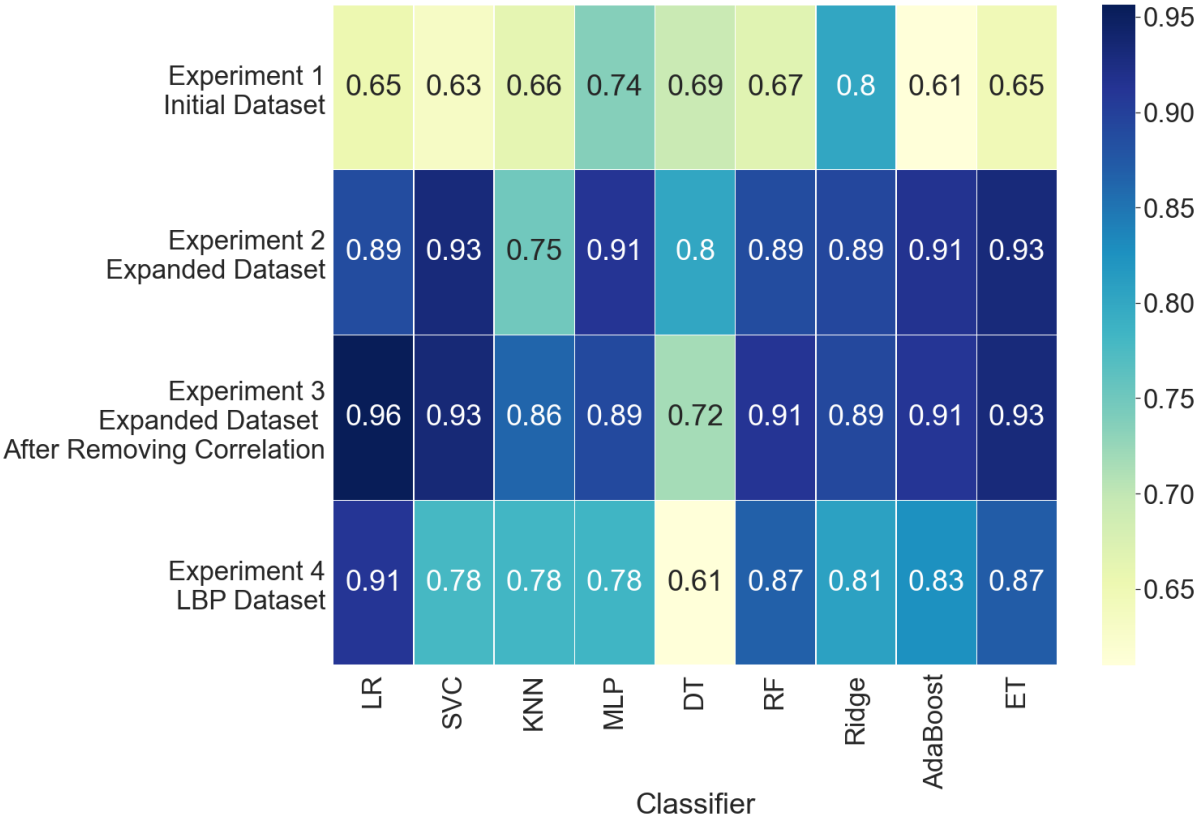


Figure 52. The overall heatmap compares the prediction performance of classifiers based on the AUC score in four experiments on putamen datasets.

4.4 The Thalamus

4.4.1 Selected Features using RENT

In the thalamus experiments, we performed RENT one time without generating any polynomial features.

Selected Features in Experiment 1

In experiment 1, by applying RENT, we obtained a reduced dataset with 16 features (out of 328 features) given in Table 15.

Table 15. Selected features attribute in experiment 1 on "initial dataset" for the thalamus. Shape denoted shape features 128-bin and 64-bin refer to the texture features with 128 and 64 grey level discretisation. Right or Left indicate the right or left side of the brain.

	Feature Name	Side	Feature Type
1	Shape_MajorAxisLength_left	Left	Shape
2	Shape_Elongation_right	Right	Shape
3	Shape_Maximum2DDiameterRow_right	Right	Shape
4	Shape_MinorAxisLength_right	Right	Shape
5	Shape_Sphericity_right	Right	Shape
6	Shape_SurfaceArea_right	Right	Shape
7	128_ClusterShade_d_1_left	Left	128-bin
8	128_DifferenceVariance_d_1_left	Left	128-bin
9	128_SmallAreaLowGrayLevelEmphasis_left	Left	128-bin
10	128_SizeZoneNonUniformityNormalized_right	Right	128-bin
11	128_SmallAreaEmphasis_right	Right	128-bin
12	128_LargeDependenceLowGrayLevelEmphasis_left	Left	128-bin
13	128_LargeDependenceHighGrayLevelEmphasis_right	Right	128-bin
14	64_ClusterShade_d_1_left	Left	64-bin
15	64_DifferenceVariance_d_1_left	Left	64-bin
16	64_Busyness_left	Left	64-bin

The distribution of selected features in Figure 53 shows that most of the selected features were texture features of the the128-bin type (44%) versus 19% of the 64-bin type. 37% of selected features were from the shape features category. Figure 53b shows that 53% of the selected features were from the right side of the brain.

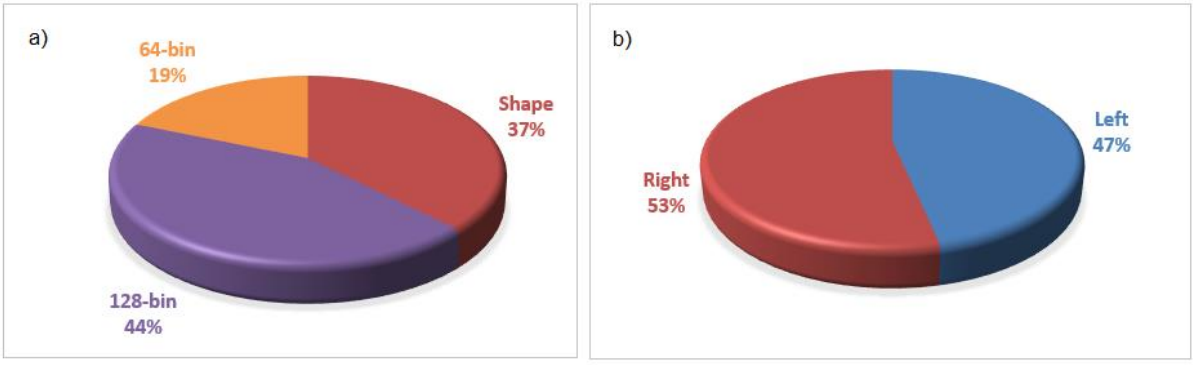


Figure 53. Pie charts show the distribution of selected features from the "initial dataset" in experiment 1 for the thalamus. a) the distribution of selected features based on the feature type. 128-bin and 64-bin refer to the texture features with, respectively, 128 and 64 grey level discretisation. Shape denotes the shape features. b) the distribution of features selected from the left or right sides of the brain.

Selected Features in Experiment 2

In experiment 2, 15 features out of 348 radiomics features in the "expanded dataset" (see Figure 21) were selected by RENT. We used these selected features, listed in Table 16, for constituting the final reduced dataset.

Table 16. Selected features attribute in experiment 2 on "expanded dataset" for the thalamus. Shape denotes the shape features. LBP corresponds to LBP features. 128-bin and 64-bin refer to the texture features with 128 and 64 grey level discretisation, respectively. Right or Left indicate the right or left side of the brain, respectively.

	Feature Name	Side	Feature Type
1	Shape_Sphericity_right	Right	shape
2	Shape_SurfaceArea_right	Right	shape
3	128_ClusterShade_d_1_right	Right	128-bin
4	128_LargeAreaHighGrayLevelEmphasis_left	Left	128-bin
5	64_ClusterShade_d_1_left	Left	64-bin
6	64_ClusterShade_d_1_right	Right	64-bin
7	LBP_111_left	Left	LBP
8	LBP_021_left	Left	LBP
9	LBP_300_left	Left	LBP
10	LBP_201_left	Left	LBP
11	LBP_003_left	Left	LBP
12	LBP_021_right	Right	LBP
13	LBP_012_right	Right	LBP
14	LBP_003_right	Right	LBP
15	LBP_102_right	Right	LBP

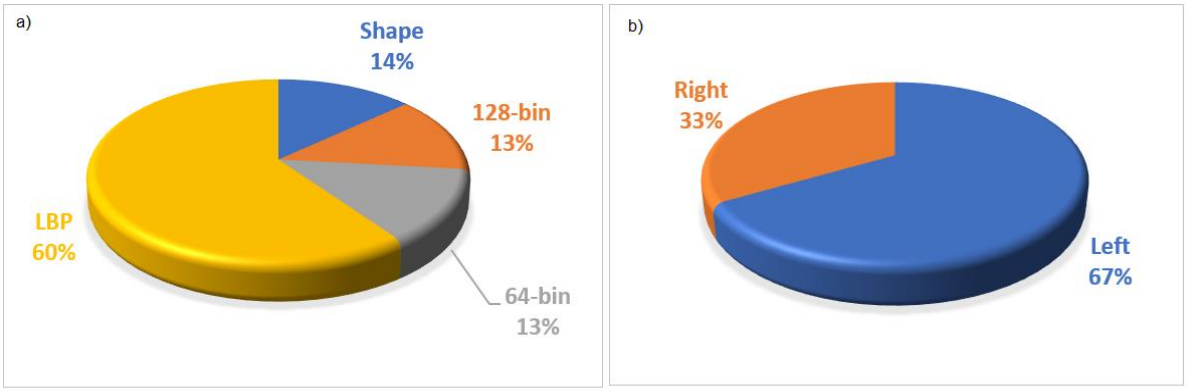


Figure 54. Pie charts show the characteristics of selected features from the "expanded dataset" in experiment 2 for the thalamus. a) the distribution of selected features based on the feature type. LBP corresponds to the LBP features. 128-bin and 64-bin refer to the texture features with, respectively, 128 and 64 grey level discretisation. Shape denotes the shape features. b) the distribution of features selected from the left or right sides of the brain.

From Figure 54a and Table 16, we can observed that the majority of the selected features were LBP features (60%), followed by texture features (26%) and shape features (14%) (Figure 54a) and from the left side of the brain (Figure 54b).

Selected Features and Feature Correlation in Experiment 3

Features Collinearity

The heatmap of features correlations between 13 features selected by RENT in experiment 2 is shown in Figure 55. The only pairs of features with correlation above 70% was *64_ClusterShade_d_1_right* and *128_ClusterShade_d_1_right*.

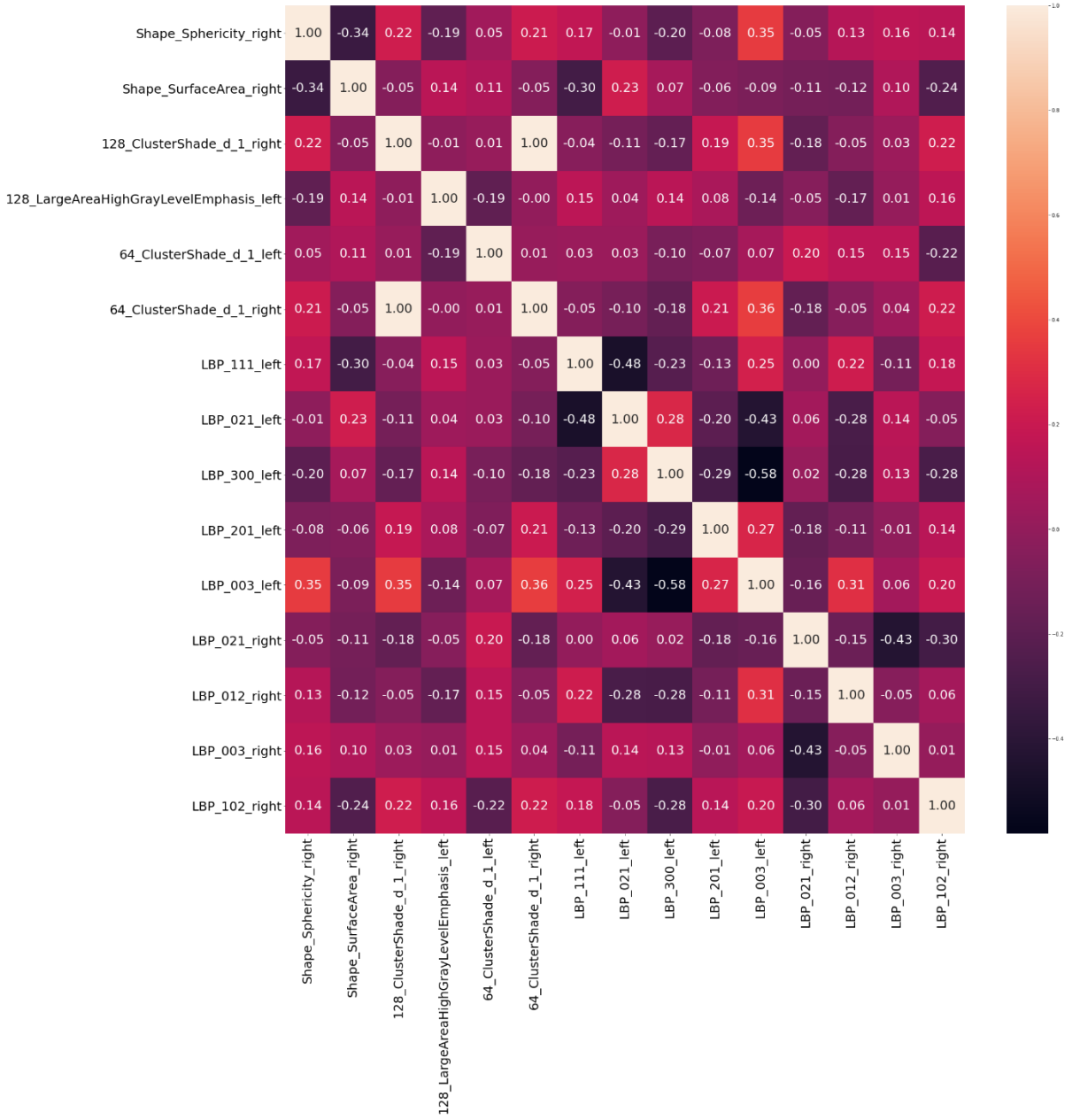


Figure 55. The correlation heatmap of the 13 features selected by RENT in experiment 2 for the thalamus. The values show the Spearman Correlation Coefficient between pairs of features.

Selected Features in experiment 3

The distribution of features in the “expanded dataset” obtained after removing highly correlated features is shown in Figure 56. 195 out of the 348 features were highly correlated to another feature and were removed, giving a reduced dataset with 153 features. The LBP features constructed 13% of this dataset in comparison to shape feature (16%), texture feature 128-bin (41%) and 64-bin (30%). All the LBP features were included in this reduced dataset showing no highly correlated features among LBP features. The features were approximately equally distributed from the left and right side of the brain (Figure 56b).

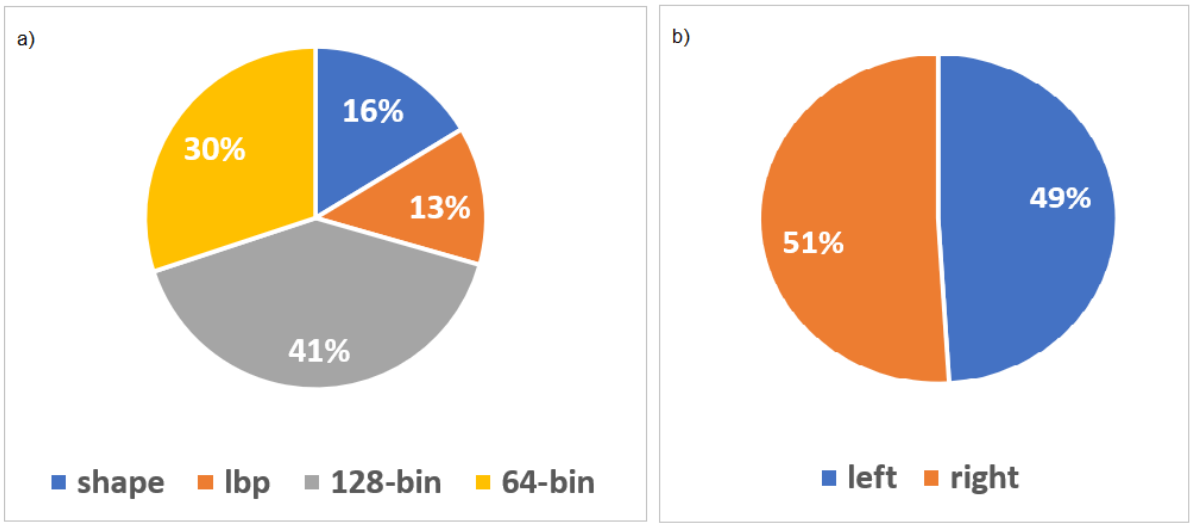


Figure 56. Pie charts show the distribution of various radiomics features in the dataset obtained after removing highly correlated features from the "expanded dataset" in experiment 3 for the thalamus. a) the distribution of features based on the feature type. 128-bin and 64-bin refer to the texture features, respectively, 128 and 64 grey level discretisation. Shape denotes the shape features. LBP corresponds to LBP features. b) the distribution of features from the left or right sides of the brain.

After performing RENT on the dataset without highly correlated feature, we obtained a reduced dataset with 13 features (from 153 radiomics features), given in Table 17. 57% of these features were LBP features (Figure 57a), and most of the features were selected from the left side of the brain (Figure 57b).

Table 17. Selected features attribute in experiment 3 for the thalamus. Shape denotes the shape features. LBP corresponds to LBP features. 128-bin refers to the texture features with 128 grey level discretisation. Right or Left indicate the right or left side of the brain, respectively.

	Feature Name	Side	Feature Type
1	Shape_Sphericity_right	Right	Shape
2	Shape_SurfaceArea_right	Right	Shape
3	128_ClusterShade_d_1_left	Left	128-bin
4	128_ClusterShade_d_1_right	Right	128-bin
5	128_LargeAreaHighGrayLevelEmphasis_left	Left	128-bin
6	LBP_111_left	Left	LBP
7	LBP_021_left	Left	LBP
8	LBP_300_left	Left	LBP
9	LBP_003_left	Left	LBP
10	LBP_021_right	Right	LBP
11	LBP_012_right	Right	LBP
12	LBP_003_right	Right	LBP
13	LBP_102_right	Right	LBP

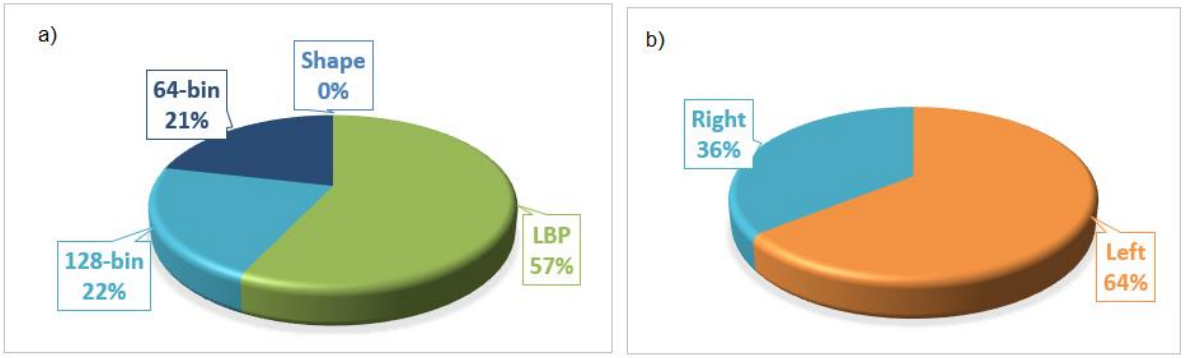


Figure 57. Pie charts show the characteristics of selected features from the dataset obtained after removing highly correlated features from the "expanded dataset" in experiment 3 for the thalamus. a) the distribution of selected features based on the feature type. LBP corresponds to the LBP features. 128-bin and 64-bin refer to the texture features with, respectively, 128 and 64 grey level discretisation. Shape denotes the shape features. b) the distribution of features selected from the left or right sides of the brain.

4.4.2 Heatmap Comparison of the Experiments

Figure 58 shows the overall heatmap of the thalamus experiments having AUC scores from 40% to 100%. The LGBM result was excluded from the heatmap. SVC had the highest performance in experiment 2 with a score of 100%; in contrast, the lowest

score (40%) was related to the MLP classifier in experiment 1. The scores in experiments 2,3, and 4 outperformed the scores of experiments 1.

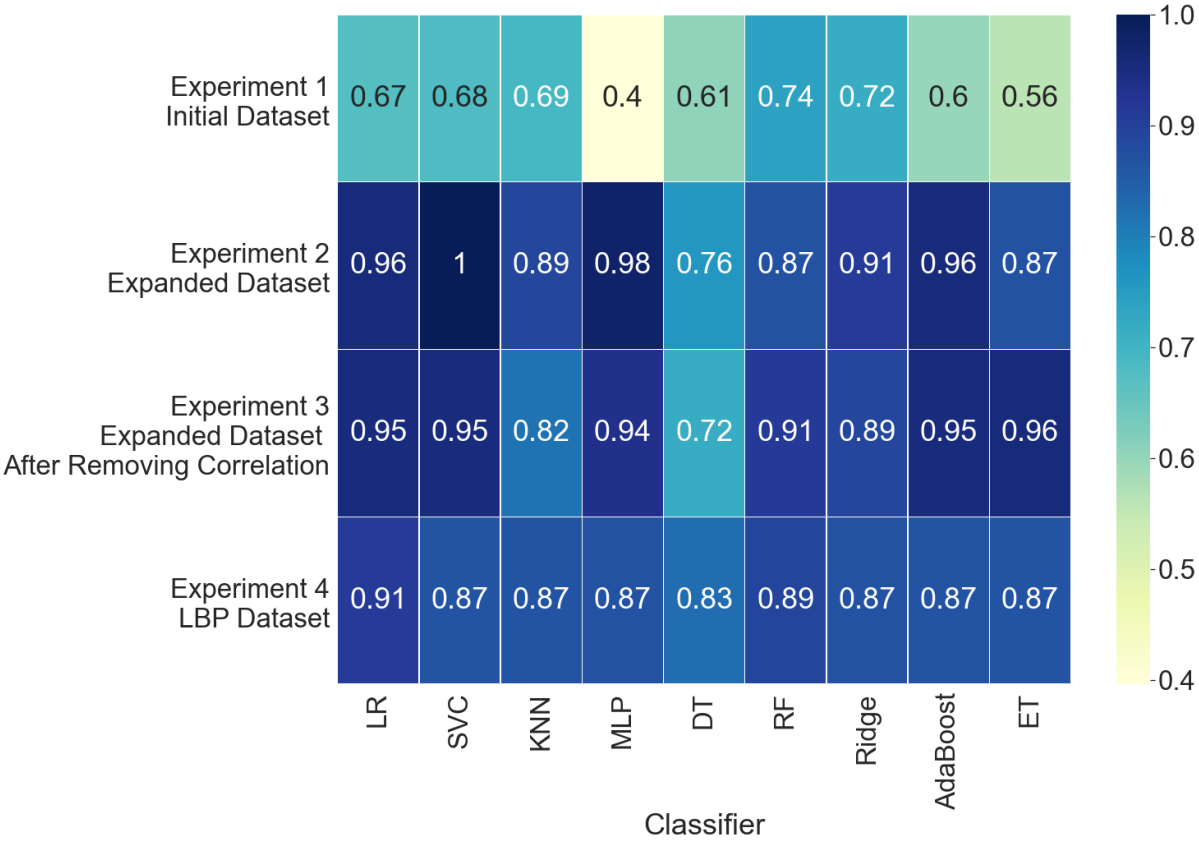


Figure 58. The overall heatmap shows the comparison between the performance of the classifiers based on the AUC score in four experiments on the thalamus datasets.

4.5 The Pallidum

4.5.1 Selected Features using RENT

In the pallidum experiments, we performed RENT one time without generating any polynomial features.

Selected Features in Experiment 1

In experiment 1, by applying RENT to the “initial dataset” (see Figure 18), we obtained a reduced dataset with 17 features out of 328 radiomics features, given in Table 18. Most of the selected features were texture features from the right side of the brain (Figure 59).

Table 18. Selected features attribute in experiment 1 on the “initial dataset” for the pallidum. Shape denotes the shape features. 128-bin and 64-bin refer to the texture features with,

respectively, 128 and 64 grey level discretisation. Right or Left indicate the right or left side of the brain, respectively.

	Feature Name	Side	Feature Type
1	Shape_MajorAxisLength_left	Left	Shape
2	Shape_MajorAxisLength_right	Right	Shape
3	Shape_Maximum2DDiameterColumn_right	Right	Shape
4	Shape_Maximum2DDiameterRow_right	Right	Shape
5	Shape_Maximum2DDiameterSlice_right	Right	Shape
6	128_ClusterProminence_d_1_left	Left	128-bin
7	128_JointAverage_d_1_left	Left	128-bin
8	128_SumAverage_d_1_left	Left	128-bin
9	128_MaximumProbability_d_1_right	Right	128-bin
10	128_ZoneEntropy_right	Right	128-bin
11	64_ClusterProminence_d_1_left	Left	64-bin
12	64_JointAverage_d_1_left	Left	64-bin
13	64_SumAverage_d_1_left	Left	64-bin
14	64_MaximumProbability_d_1_right	Right	64-bin
15	64_LowGrayLevelRunEmphasis_right	Right	64-bin
16	64_ShortRunLowGrayLevelEmphasis_right	Right	64-bin
17	64_LowGrayLevelEmphasis_right	Right	64-bin

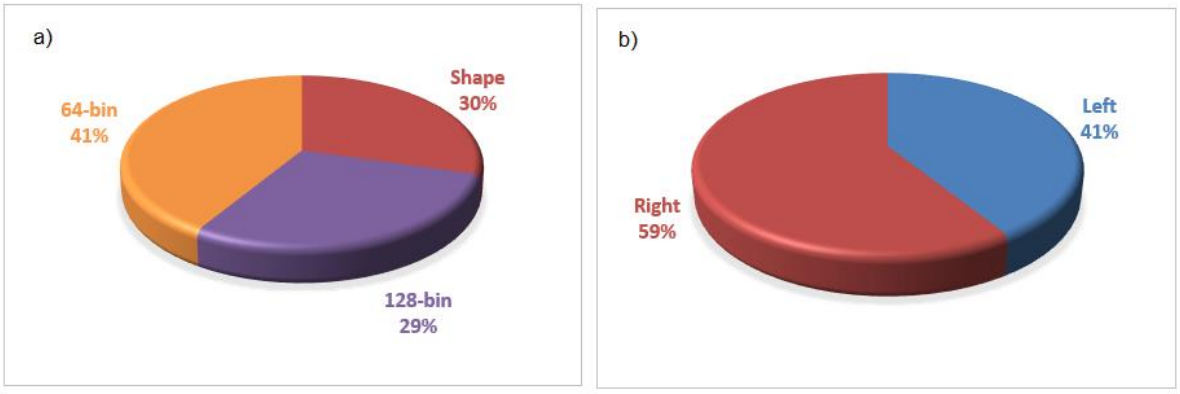


Figure 59. Pie charts show the distribution of selected features from the "initial dataset" in experiment 1 for the pallidum. a) the distribution of selected features based on the feature type. 128-bin and 64-bin refer to the texture features with, respectively, 128 and 64 grey level discretisation. Shape denotes the shape features. b) the distribution of features selected from the left or right sides of the brain.

Selected Features in Experiment 2

In this experiment, 14 features out of 348 radiomics features in the "expanded dataset" (see Figure 21) were selected by RENT (Table 19). Most of the selected features were LBP features (Figure 60) and from the left side of the brain.

Table 19. Selected features attribute in experiment 2 on the "expanded dataset" for the pallidum. Shape denotes the shape features. LBP corresponds to LBP features. 128-bin and 64-bin refer to the texture features with 128 and 64 grey level discretisation, respectively. Right or Left indicate the right or left side of the brain, respectively.

	Feature Name	Side	Feature Type
1	Shape_Maximum2DDiameterColumn_right	Left	Shape
2	Shape_Maximum2DDiameterRow_right	Left	Shape
3	Shape_Maximum2DDiameterSlice_right	Left	Shape
4	128_ClusterProminence_d_1_left	Left	128-bin
5	128_Imc1_d_1_right	Left	128-bin
6	64_ClusterProminence_d_1_left	Left	64-bin
7	LBP_111_left	Left	LBP
8	LBP_030_left	Left	LBP
9	LBP_021_left	Left	LBP
10	LBP_201_left	Left	LBP
11	LBP_003_left	Right	LBP
12	LBP_030_right	Right	LBP
13	LBP_201_right	Right	LBP
14	LBP_102_right	Right	LBP

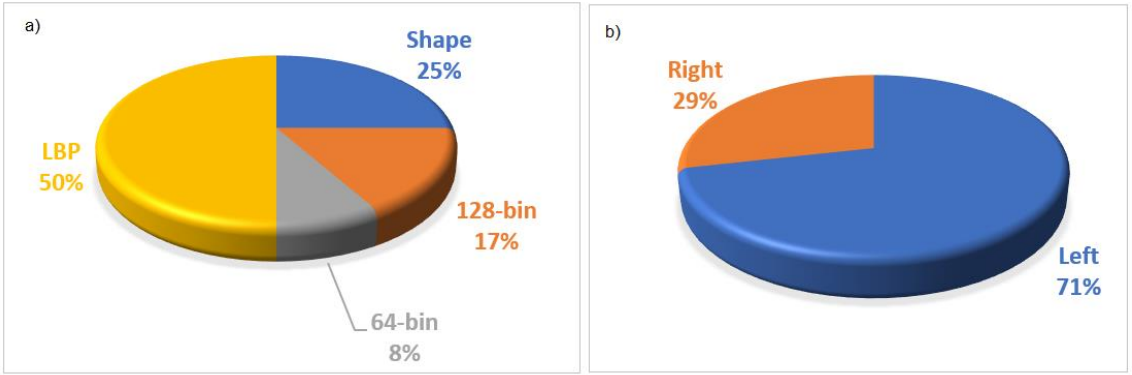


Figure 60. Pie charts show the characteristics of selected features from the "expanded dataset" in experiment 2 for the pallidum. a) the distribution of selected features based on the feature type. LBP corresponds to the LBP features. 128-bin and 64-bin refer to the texture features with, respectively, 128 and 64 grey level discretisation. Shape denotes the shape features. b) the distribution of features selected from the left or right sides of the brain.

Selected Features and Feature Correlation in Experiment 3

Features Collinearity

The correlation between the 14 features selected by RENT (from 348 radiomics features in the "expanded dataset", experiment 2) is shown in Figure 61.

The features with correlation above 70% were *64_ClusterProminence_d_1_left* and *128_ClusterProminence_d_1_left*.

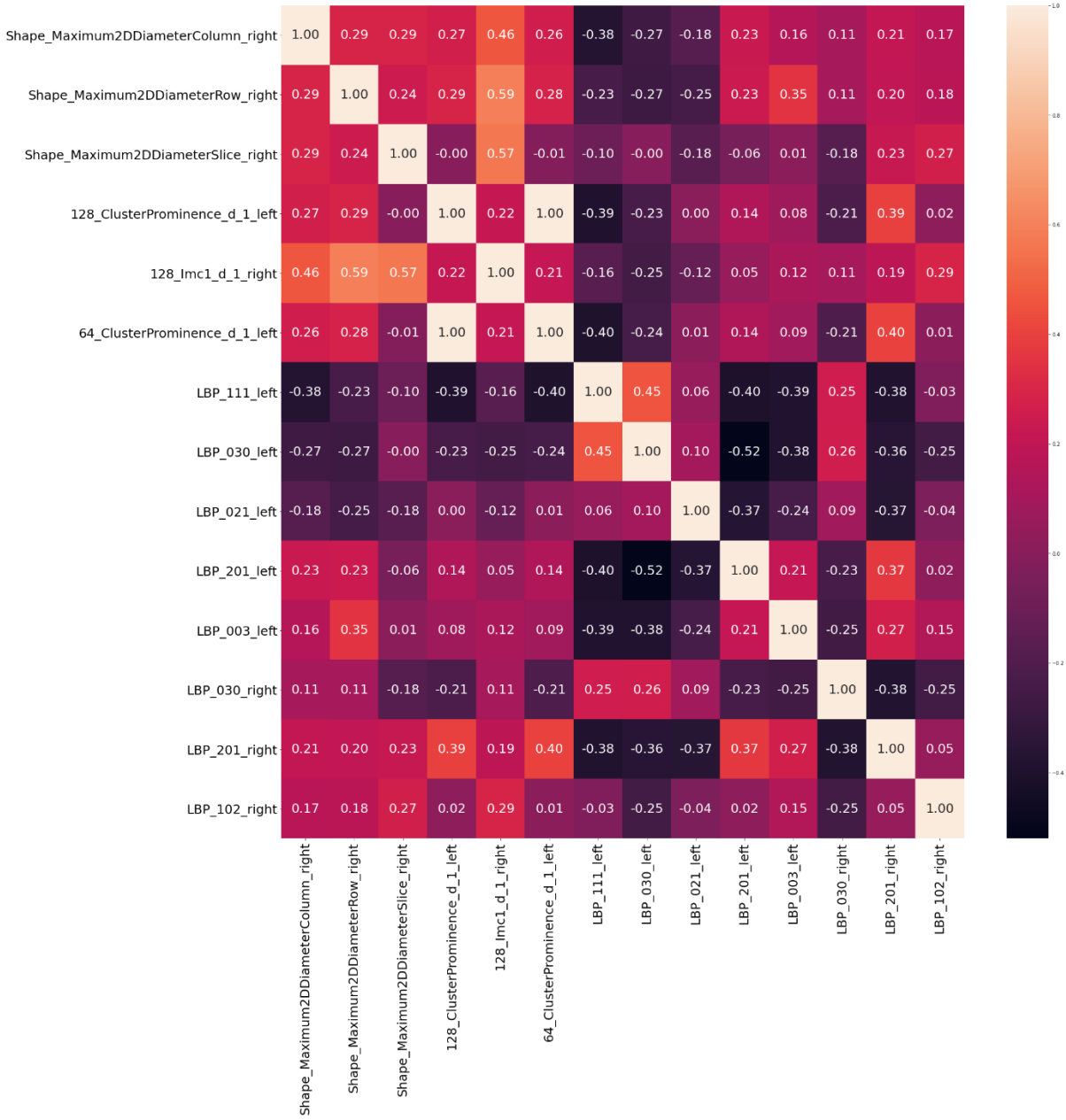


Figure 61. The correlation heatmap of features selected by RENT in experiment 2 for the pallidum. The values show the Spearman Correlation Coefficient between pairs of features.

Selected Features in Experiment 3

After removing the highly correlated features, we were left with 194 out of 348 features of the “expanded dataset”. Most of these selected features were texture features

(Figure 62a). About an equal number of features were selected from the right and left side of the brain (Figure 62b).

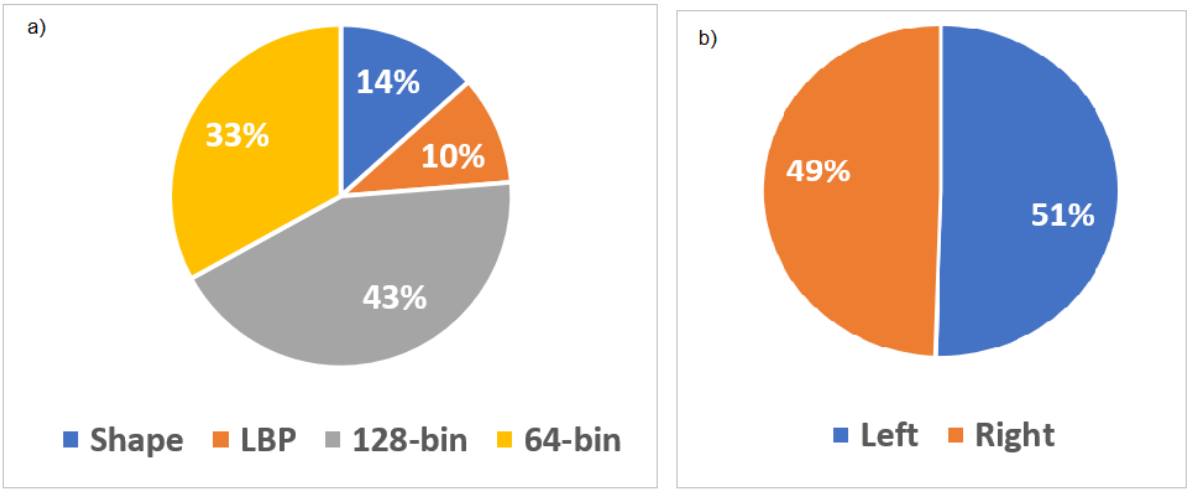


Figure 62. Pie charts show the distribution of various radiomics features in the dataset obtained after removing highly correlated features from the "expanded dataset" in experiment 3 for the pallidum. a) the distribution of features based on the feature type. 128-bin and 64-bin refer to the texture features, respectively, 128 and 64 grey level discretisation. Shape denotes the shape features. LBP corresponds to LBP features. b) the distribution of features from the left or right sides of the brain.

By performing RENT, we reduced the number of features to 15 (from 194 radiomics features), given in Table 20. Most of the selected features were LBP features (Figure 63a) and from the right side of the brain (Figure 63b).

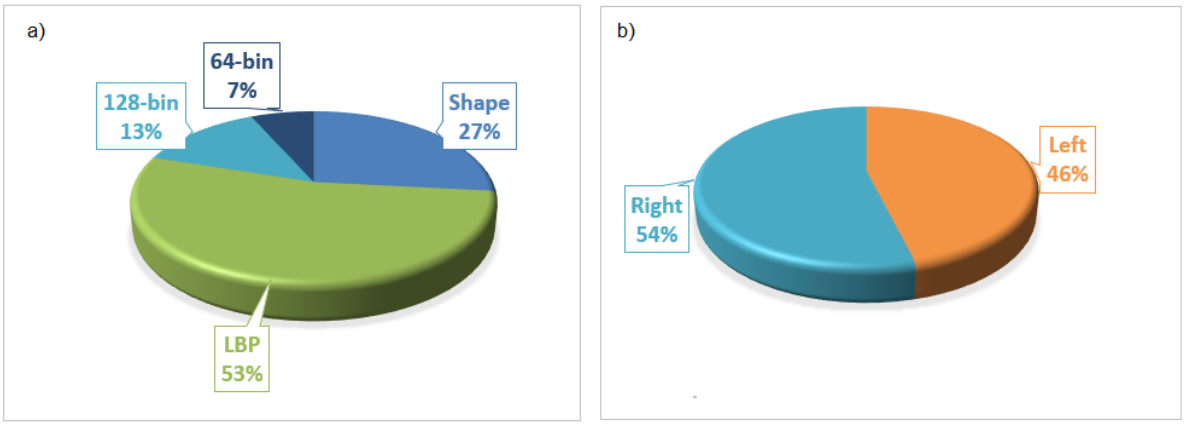


Figure 63. Pie charts show the characteristics of selected features from the dataset gained after removing highly correlated features from the "expanded dataset" in experiment 3 for the pallidum. a) the distribution of selected features based on the feature type. LBP corresponds to the LBP features. 128-bin and 64-bin refer to the texture features with, respectively, 128 and 64 grey level discretisation. Shape denotes the shape features. b) the distribution of features selected from the left or right sides of the brain.

Table 20. Selected features attribute in experiment 3 for the pallidum. Shape denotes the shape features. LBP corresponds to LBP features. 128-bin and 64-bin refer to the texture features with 128 and 64 grey level discretisation, respectively. Right or Left indicate the right or left side of the brain, respectively.

	Feature Name	Side	Feature Type
1	Shape_MinorAxisLength_left	Left	Shape
2	Shape_Maximum2DDiameterColumn_right	Right	Shape
3	Shape_Maximum2DDiameterRow_right	Right	Shape
4	Shape_Maximum2DDiameterSlice_right	Right	Shape
5	128_ClusterProminence_d_1_left	Left	128-bin
6	128_Imc1_d_1_right	Right	128-bin
7	64_LongRunLowGrayLevelEmphasis_right	Right	64-bin
8	LBP_111_left	Left	LBP
9	LBP_030_left	Left	LBP
10	LBP_021_left	Left	LBP
11	LBP_201_left	Left	LBP
12	LBP_003_left	Left	LBP
13	LBP_030_right	Right	LBP
14	LBP_201_right	Right	LBP
15	LBP_102_right	Right	LBP

4.5.2 Heatmap Comparison of the Experiments

The overall heatmap of the pallidum experiments is shown in Figure 64. We eliminated the result from LGBM as it had constantly poor results. The highest scores were obtained in experiments 1 and 3, closely followed by experiment 4. Note that the SVC classifier obtained a score of only 50% in experiments 2 and 3.

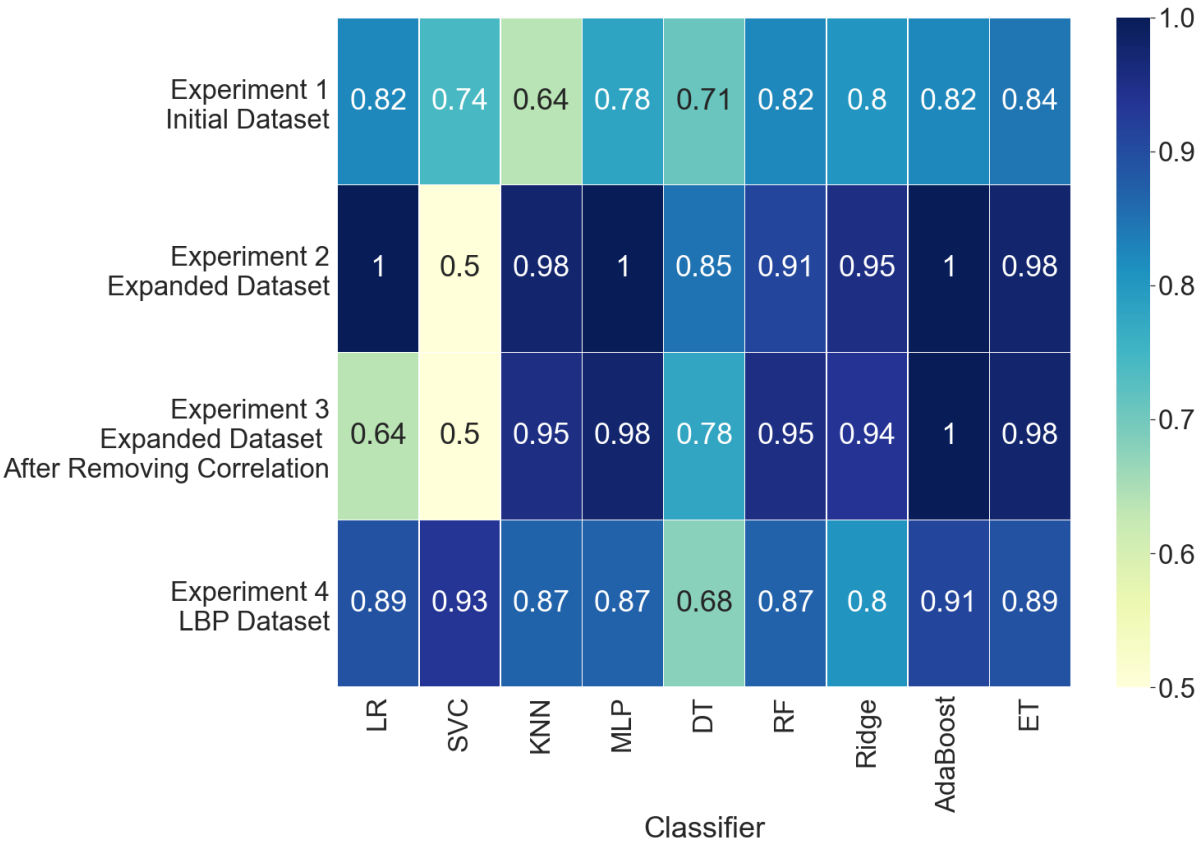


Figure 64. The overall heatmap compares classifiers' performance based on the AUC score in four experiments on pallidum datasets.

5 Discussion and Further work

The main objective of this study was to examine whether ADHD medication alters the structures in the grey matter of brains of male patients aged 10 to 12. The output of the classification experiments suggested detactable patterns in the five subcortical brain structures (hippocampus, caudate, pallidum, putamen, and thalamus) due to the assigned medication. These findings, along with the findings related to the secondary objectives, will be discussed in the first two sections of this chapter. At the end of this chapter, we will present some areas for future work.

5.1 Selected Features

By observing the overall results of selected features distribution (Table 21), one could observe that features from the right side of the brain were preferred to the features from the left side. Mainly, the right side of the hippocampus and caudate structures always prevailed over the left side of those structures. In a previous study by Grünbeck [13], similarly, the right-sided ROIs dominated among several feature selection methods. This was also mentioned in the papers review by Frodl et al. [84] that the right side of the brain structures was influenced by ADHD and perhaps by MPH medication.

Table 21. A comparison between selected features by RENT (input dataset for modelling step). This table shows which feature type and side (left or right) of the brain had the highest selection rate. LBP corresponds to LBP features. 128-bin and 64-bin refer to the texture features with 128 and 64 grey level discretisation, respectively.

	Experiment 1		Experiment 2		Experiment 3	
	Side	**Type**	**Side**	**Type**	**Side**	**Type**
Hippocampus	Right	128-bin	Right	LBP	Right	LBP
Caudate	Right	64-bin	Right	LBP & 64-bin	Right	LBP
Putamen	Right	64-bin	Left	LBP	Left	LBP
Thalamus	Right	128-bin	Left	LBP	Left	LBP
Pallidum	Right	64-bin	Left	LBP	Right	LBP

Furthermore, Table 21 shows that the texture features (including LBP, 128-bin and 64-bin features) were always preferred to the shape features. Therefore, it appears that the relevant information for discriminating between MPH and controls was found in the texture features, particularly LBP features and not in the shape of the structure. Similarly, the results of using various feature selectors in the study by Grünbeck [13] showed that most of the selected features were from the texture features and not the shape features.

The LBP descriptor is known for being straightforward and, at the same time, highly discriminative. It is less sensitive to the inhomogeneity of image and noise artefacts [47]. It has not been frequently used in radiomics studies. Because the medical images used in radiomics studies may be three dimensional, having a 3D LBP implementation in Python was necessary. Hence, we developed a 3D LBP feature extraction programme. The calculation of LBP was simpler and faster than other texture or shape features.

Table 21 demonstrates that whenever LBP features were presented in a dataset, they were the dominant feature type selected, showing that they capture more of the relevant texture than the other feature types. Even though the number of LBP features in the dataset before feature selection was much lower than the number of other texture features (20 versus 300), most selected features were still LBP features. Likewise, in the study by Peeken et al. [85], LBP features achieved better prediction results than shape features and other texture features.

5.2 Classification Performance

From the classification results in chapter 4, we observed that the AUC scores varied across different experiments showing improvements compared to the previous study done by Grünbeck [13]. In our study, most of the AUC scores were above 85%,

especially in experiments 2, 3 and 4, in which LBP features were included in addition to the standard radiomics features or stand-alone features. However, in the study by Grünbeck, where only the standard radiomics features were included, no brain structure obtained scores above 80%. The better performance achieved by experiments 2, 3 and 4 indicates that adding LBP features improved the model performance compared to the situations that only used standard radiomics features as demonstrated in the result of experiment 1 of this study and Grünbeck's study.

Furthermore, although removing highly correlated features improved the performance in some cases, there was no marked performance increase between experiment 2 using a dataset with correlated features and experiment 3 using a dataset where correlated features had been removed. This showed that RENT had selected robust features.

If we compare the classification results of experiment 1, in which we used the same dataset as Grünbeck [13], but with a different feature selection method, we observed that we attained higher classification performance for all the brain structures. Grünbeck [13] employed several feature selection algorithms such as Low Variance Threshold, Fisher Score, a modified version of Mutual Information Classifier and ReliefF, along with no feature selection. In her study, the brain structures showed AUC performance scores below 60%, and the AUC scores rarely surpassed 70%, while the AUC of experiment 1 (the same dataset as Grünbeck's study based on the dataset containing standard radiomics features) of our study mainly were close to 70% and more. In experiment 1, the highest AUC score was achieved for the hippocampus set (85%), followed by pallidum (84%), putamen (80%), caudate (78%) and thalamus (74%) compared to the results achieved by Grünbeck where the best score was obtained for the pallidum set (79%) followed by putamen (76%), hippocampus (71%), caudate (64%) and thalamus (64%). This can indicate the RENT's ability in selecting features that are more discriminant than the feature selection methods used in Grünbeck's study. Furthermore, it should be pointed out that RENT provides two validation studies to evaluate its performance with diagrams presented in Appendix F. These statistical tests showed that the selected features by RENT give significantly higher classification performance than randomly selected features or permuted test labels.

As we saw in the heatmaps of classifier performances in chapter 4, Ridge, AdaBoost and ET were the classifiers that showed AUC scores above 80% most of the time. If we have a closer look at the definition of these algorithms, we may know the reason for their higher performance relative to the other classifiers. Ridge is a regularisation algorithm. The regularisation techniques reduce the variance of the model and increase the model generalisability [86]. AdaBoost and ET are ensemble algorithms trying to combine several weak classifiers into one robust classifier [87], [88].

Ridge achieved acceptable results in the study by Langberg [18] in which the biomarkers related to disease free survival in head and neck cancers were examined. The AdaBoost algorithm has been used widely in classification studies based on medical images because of its robust and stable prediction performance [89]. For instance, Zhang et al. (2019) [90] examined arteriovenous malformation related hematomas using radiomics where the AdaBoost algorithm had superiority compared to other classifiers (such as DT, RF, LR, SVC, KNN). There are some studies in radiomics that used ET classifier and attained good results. For example, Gabryś et al. (2018) [91] studied the risk assessment of xerostomia by using radiomics and other methods. In their study, the ET algorithm outperformed SVM, LR and KNN classifiers. In Grünbeck's study [13], DT and ET showed relatively higher scores than other classifiers included in her study (Ridge, LGBM, SVC and LR).

However, in current research, SVC and DT showed the worst performance having prediction scores mostly below 80%. Despite the acceptable performance of the LGBM classifier in Grünbeck's study, in our study, this classifier had a poor performance by predicting all the labels as class 0, which led to the constantly poor score of 50% in all experiments.

The AUC scores of different experiments showed that the highest score for hippocampus was 98% and for putamen was 96%, while the best scores of thalamus, caudate, and palladium were 100% showing a possibility of overfitting. In 2017, Hoogman et al. [92] examined several brain structures, including the five structures used in this study. They reported reduced volumes in hippocampus, caudate and putamen in ADHD patients. In another study, Schrantee et al. (2016) [10] explored the effect of MPH on the dopaminergic system of ADHD children. They observed an impact of MPH treatment on caudate, putamen and thalamus.

It should be stressed that because of not having any independent validation dataset and having very few samples, our models were prone to overfit potentially. We used the nested cross-validation method to tackle the overfitting issue and visualise the model's behaviour on unseen data. However, because of few samples, the risk of overfitting still existed. The nested cross-validation was used in studies with few samples where hyperparameter tuning is required [93]–[96]. For instance, in the study by Smit et al. (2007) [93], nested cross-validation for modelling was used and evaluated on a dataset with very few samples. In another study, a review paper on cross-validation methods in neuroimaging, Varoquaux et al. (2017) [96] mentioned that the nested cross-validation could be the choice in case of the limited amount of data. According to Maleki et al. (2020) [95], when dealing with few samples, nested cross-validation is a better alternative than cross-validation, especially when the hyperparameter tunning process is included because it can provide a reliable

generalisation error versus the cross-validation, which estimates the error over-optimistically. Nested cross-validation can tackle the overfitting problem because, in nested cross-validation, the prediction takes place in the outer loop on the data, which is new to the predictive model [93].

Overall, the results presented suggest that there may be detectable changes in the brain structure due to MPH medication. This requires more examinations, gathering more samples or having external validation data to remove the possibility of overfitting.

5.3 Further Work

The high performance scores in almost every classification experiment and the comparison with the previous study by Grünbeck [13] on a similar dataset suggest differences between the two treated groups, connected to detectable changes due to MPH treatment.

One of the limitations of radiomics studies is having datasets with very few samples and a lack of unseen data. These limitations can make the prediction process suffer from overfitting. We tried to decrease this risk by using the nested cross-validation method and observing the train and validation curves. From these curves, we can get a better understanding of the classifier's prediction behaviour. Despite employing the nested cross-validation method to increase the generalisability of classification tasks, our predictive models can still be susceptible to overfitting. Thus, we suggest performing similar training and validation experiments by including more samples and validating unseen external data to confirm the promising results of this research.

Another limitation in radiomics studies is that the extracted features are not interpreted because many features are included and the radiomics features (especially texture features and higher-order features) are slightly non-understandable. We propose a further study on investigating the selected features' interpretation. Moreover, we suggest the inclusion of demographic information and other characteristics of the patients.

In our study, we also followed the idea of developing a 3D LBP feature extraction tool. We observed that LBP features were more informative than shape features and other texture feature. Our 3D LBP code can be upgraded by considering more neighbours, for example, the neighbours on the diagonal (considering 26 neighbours instead of 6 neighbours) or having user-defined values for the number of neighbouring nodes (P) and the distance of the neighbourhood cells from the centre node (R). In addition, we recommend using other extensions of the LBP method. For instance, the Local Ternary Pattern (LTP) descriptor [97] instead of the LBP operator. LTP is a

generalisation of LBP introduced by Tan et al. (2010). LTP encodes the surrounding voxels into three labels (less than zero, equal to zero and greater than zero). In contrast, LBP thresholds the intensity values into two class labels (to equal or greater than zero or less than zero). According to [97], LTP is less sensitive to noise and more discriminative in uniform and near-uniform areas than LBP. Also, LTP has the LBP advantage of computational efficiency [97].

6 Conclusion

The main goal of this thesis was to investigate changes to five brain structures caused by MPH medication in ADHD male children. Features of these brain structures were extracted from MR images acquired from the ePOD-MPH study [11]. We achieved very promising results, which suggest detectable changes to the five subcortical structures of the brain (hippocampus, caudate, pallidum, putamen, and thalamus). We achieved high performance scores for classifying the children into the MPH medication group or the placebo group.

Four classification experiments based on four different datasets were done. The results were very promising, with AUC scores mostly above 80% indicating improvement compared to a previous study by Grünbeck [13] on the same data. This and the relatively similar prediction performance achieved in experiment 2 (using a dataset contained LBP and other standard radiomics features with correlated features) and experiment 3 (using the dataset with the same feature types as in experiment 2 but excluding highly correlated features) showed the efficiency of RENT versus various feature selectors used in Grünbeck's study (Low Variance Threshold, Fisher Score, modified version of Mutual Information Classifier and ReliefF besides no feature selection step).

In the current study, we analysed a short-wide dataset utilising the radiomics pipeline. It is very common in radiomics studies to assess a short-wide dataset (high dimension with few samples). In this thesis, we explored a new feature selection tool, RENT which was claimed to be appropriate for this kind of dataset (short-wide). For tackling the issues of the lack of a validation set (unseen data), we used nested cross-validation and visualised classifiers behaviour by depicting train and validation curves to give the reader a clearer observation of the classification process. However, our models may be inclined to overfitting, particularly in the models with AUC scores above 95%.

The LBP features have shown their high discriminative power in computer vision studies over the years. They are not frequently used in radiomics studies. Therefore, another objective of this thesis was to develop a feature extraction tool for extracting 3D LBP features in the Python programming language. The code is available in Appendix A and added to the Biorad feature extraction module. In this research, whenever LBP was included in a dataset, they were preferred to the shape features and other texture features. Also, the classification experiments using LBP features (either as a stand-alone feature or in addition to the standard radiomics features) had very high prediction scores.

Bibliography

[1] H. Sun *et al.*, “Psychoradiologic Utility of MR Imaging for Diagnosis of Attention Deficit Hyperactivity Disorder: A Radiomics Analysis,” *Radiology*, vol. 287, no. 2, Art. no. 2, May 2018, doi: 10.1148/radiol.2017170226.

[2] T. Eslami, F. Almuqhim, J. S. Raiker, and F. Saeed, “Machine Learning Methods for Diagnosing Autism Spectrum Disorder and Attention- Deficit/Hyperactivity Disorder Using Functional and Structural MRI: A Survey,” *Front. Neuroinform.*, vol. 14, 2021, doi: 10.3389/fninf.2020.575999.

[3] V. Engert and J. C. Pruessner, “Dopaminergic and noradrenergic contributions to functionality in ADHD: the role of methylphenidate,” *Curr Neuropharmacol*, vol. 6, no. 4, Art. no. 4, Dec. 2008, doi: 10.2174/157015908787386069.

[4] C. Bouziane, O. G. Filatova, A. Schrantee, M. W. A. Caan, F. M. Vos, and L. Reneman, “White Matter by Diffusion MRI Following Methylphenidate Treatment: A Randomized Control Trial in Males with Attention-Deficit/Hyperactivity Disorder,” *Radiology*, vol. 293, no. 1, pp. 186–192, Oct. 2019, doi: 10.1148/radiol.2019182528.

[5] M. Klein *et al.*, “Brain imaging genetics in ADHD and beyond - Mapping pathways from gene to disorder at different levels of complexity,” *Neurosci Biobehav Rev*, vol. 80, pp. 115–155, Sep. 2017, doi: 10.1016/j.neubiorev.2017.01.013.

[6] M. Starck, J. Grünwald, and A. A. Schlarb, “Occurrence of ADHD in parents of ADHD children in a clinical sample
 ,” *NDT*, vol. 12, pp. 581–588, Mar. 2016, doi: 10.2147/NDT.S100238.

[7] K. B. Walhovd *et al.*, “Methylphenidate Effects on Cortical Thickness in Children and Adults with Attention-Deficit/Hyperactivity Disorder: A Randomized Clinical Trial,” *AJNR Am J Neuroradiol*, vol. 41, no. 5, Art. no. 5, May 2020, doi: 10.3174/ajnr.A6560.

[8] H. G. H. Tamminga *et al.*, “Do effects of methylphenidate on cognitive performance last beyond treatment? A randomized placebo-controlled trial in boys and men with ADHD,” *European Neuropsychopharmacology*, vol. 46, pp. 1–13, May 2021, doi: 10.1016/j.euroneuro.2021.02.002.

[9] T. Grund, K. Lehmann, N. Bock, A. Rothenberger, and G. Teuchert-Noodt, “Influence of methylphenidate on brain development--an update of recent animal experiments,” *Behav Brain Funct*, vol. 2, p. 2, Jan. 2006, doi: 10.1186/1744-9081-2-2.

[10] A. Schrantee *et al.*, “Age-Dependent Effects of Methylphenidate on the Human Dopaminergic System in Young vs Adult Patients With Attention-Deficit/Hyperactivity Disorder: A Randomized Clinical Trial,” *JAMA Psychiatry*, vol. 73, no. 9, pp. 955–962, Sep. 2016, doi: 10.1001/jamapsychiatry.2016.1572.

[11] M. A. Bottelier *et al.*, “The effects of Psychotropic drugs On Developing brain (ePOD) study: methods and design,” *BMC Psychiatry*, vol. 14, no. 1, Art. no. 1, Feb. 2014, doi: 10.1186/1471-244X-14-48.

[12] A. Schrantee, H. Mutsaerts, C. Bouziane, H. Tamminga, M. Bottelier, and L. Reneman, “The age-dependent effects of a single-dose methylphenidate challenge on cerebral perfusion in patients with attention-deficit/hyperactivity disorder,” *Neuroimage Clin*, vol. 13, pp. 123–129, Nov. 2016, doi: 10.1016/j.nicl.2016.11.021.

[13] I. A. Grünbeck, “The effects of methylphenidate on brain structures of ADHD-diagnosed children : explorative analyses using radiomic features,” Norwegian University of Life Sciences, Ås, 2020. Accessed: May 24, 2021. [Online]. Available: https://nmbu.brage.unit.no/nmbu-xmlui/handle/11250/2724853

[14] M. Avanzo, J. Stancanello, and I. El Naqa, “Beyond imaging: The promise of radiomics,” *Phys Med*, vol. 38, pp. 122–139, Jun. 2017, doi: 10.1016/j.ejmp.2017.05.071.

[15] M. E. Mayerhoefer *et al.*, “Introduction to Radiomics,” *J Nucl Med*, vol. 61, no. 4, Art. no. 4, Apr. 2020, doi: 10.2967/jnumed.118.222893.

[16] Z. Liu *et al.*, “The Applications of Radiomics in Precision Diagnosis and Treatment of Oncology: Opportunities and Challenges,” *Theranostics*, vol. 9, no. 5, Art. no. 5, 2019, doi: 10.7150/thno.30309.

[17] P. Afshar, A. Mohammadi, K. N. Plataniotis, A. Oikonomou, and H. Benali, “From Hand-Crafted to Deep Learning-based Cancer Radiomics: Challenges and Opportunities,” *IEEE Signal Process. Mag.*, vol. 36, no. 4, Art. no. 4, Jul. 2019, doi: 10.1109/MSP.2019.2900993.

[18] G. S. R. E. Langberg, “Searching for biomarkers of disease-free survival in head and neck cancers using PET/CT radiomics,” Norwegian University of Life Sciences, Ås, 2019. Accessed: May 24, 2021. [Online]. Available: https://nmbu.brage.unit.no/nmbu-xmlui/handle/11250/2641820

[19] A. Albuni, “Development of a user-friendly radiomics framework,” Norwegian University of Life Sciences, Ås, 2020. Accessed: May 24, 2021. [Online]. Available: https://nmbu.brage.unit.no/nmbu-xmlui/handle/11250/2721430

[20] J. J. M. van Griethuysen *et al.*, “Computational Radiomics System to Decode the Radiographic Phenotype,” *Cancer Res*, vol. 77, no. 21, Art. no. 21, Nov. 2017, doi: 10.1158/0008-5472.CAN-17-0339.

[21] “PyRadiomic Features,” May 24, 2021. https://pyradiomics.readthedocs.io/en/latest/features.html (accessed May 24, 2021).

[22] P. Giraud *et al.*, “Radiomics and Machine Learning for Radiotherapy in Head and Neck Cancers,” *Front. Oncol.*, vol. 9, 2019, doi: 10.3389/fonc.2019.00174.

[23] A. Jenul, S. Schrunner, K. H. Liland, U. G. Indahl, C. M. Futsaether, and O. Tomic, “RENT -- Repeated Elastic Net Technique for Feature Selection,” *arXiv:2009.12780 [cs, stat]*, Feb. 2021, Accessed: May 24, 2021. [Online]. Available: http://arxiv.org/abs/2009.12780

[24] L. B. Sollaci and M. G. Pereira, “The introduction, methods, results, and discussion (IMRAD) structure: a fifty-year survey,” *J Med Libr Assoc*, vol. 92, no. 3, Art. no. 3, Jul. 2004.

[25] K. Suzuki and Y. Chen, Eds., *Artificial Intelligence in Decision Support Systems for Diagnosis in Medical Imaging*. Springer International Publishing, 2018. doi: 10.1007/978-3-319-68843-5.

[26] R. J. Gillies, P. E. Kinahan, and H. Hricak, “Radiomics: Images Are More than Pictures, They Are Data,” *Radiology*, vol. 278, no. 2, pp. 563–577, Nov. 2015, doi: 10.1148/radiol.2015151169.

[27] A. J. Wong, A. Kanwar, A. S. Mohamed, and C. D. Fuller, “Radiomics in head and neck cancer: from exploration to application,” *Transl Cancer Res*, vol. 5, no. 4, pp. 371–382, Aug. 2016, doi: 10.21037/tcr.2016.07.18.

[28] S. Yousaf, S. Anwar, and H. RaviPrakash, *Brain Tumor Survival Prediction using Radiomics Features*. 2020.

[29] V. Kumar *et al.*, “Radiomics: the process and the challenges,” *Magn Reson Imaging*, vol. 30, no. 9, pp. 1234–1248, Nov. 2012, doi: 10.1016/j.mri.2012.06.010.
[30] G. Cinarer and B. G. Emiroglu, “Classification of brain tumours using radiomic features on MRI,” *GJPAAS*, no. 12, Art. no. 12, Apr. 2020, doi: 10.18844/gjpaas.v0i12.4989.
[31] C. Hassani, B. A. Varghese, J. Nieva, and V. Duddalwar, “Radiomics in Pulmonary Lesion Imaging,” *AJR Am J Roentgenol*, vol. 212, no. 3, pp. 497–504, Mar. 2019, doi: 10.2214/AJR.18.20623.
[32] J. E. van Timmeren, D. Cester, S. Tanadini-Lang, H. Alkadhi, and B. Baessler, “Radiomics in medical imaging—‘how-to’ guide and critical reflection,” *Insights into Imaging*, vol. 11, no. 1, p. 91, Aug. 2020, doi: 10.1186/s13244-020-00887-2.
[33] “Positron emission tomography scan - Mayo Clinic.” https://www.mayoclinic.org/tests-procedures/pet-scan/about/pac-20385078 (accessed May 24, 2021).
[34] “CT scan - Mayo Clinic.” https://www.mayoclinic.org/tests-procedures/ct-scan/about/pac-20393675 (accessed May 24, 2021).
[35] “MRI - Mayo Clinic.” https://www.mayoclinic.org/tests-procedures/mri/about/pac-20384768 (accessed May 24, 2021).
[36] I. Tsougos, A. Vamvakas, C. Kappas, I. Fezoulidis, and K. Vassiou, “Application of Radiomics and Decision Support Systems for Breast MR Differential Diagnosis,” *Comput Math Methods Med*, vol. 2018, p. 7417126, 2018, doi: 10.1155/2018/7417126.
[37] M. Hatt, C. C. Le Rest, F. Tixier, B. Badic, U. Schick, and D. Visvikis, “Radiomics: Data Are Also Images,” *J Nucl Med*, vol. 60, no. Suppl 2, pp. 38S-44S, Sep. 2019, doi: 10.2967/jnumed.118.220582.
[38] P. Lohmann *et al.*, “PET/MRI Radiomics in Patients With Brain Metastases,” *Front. Neurol.*, vol. 11, 2020, doi: 10.3389/fneur.2020.00001.
[39] N. Papanikolaou, C. Matos, and D. M. Koh, “How to develop a meaningful radiomic signature for clinical use in oncologic patients,” *Cancer Imaging*, vol. 20, no. 1, p. 33, May 2020, doi: 10.1186/s40644-020-00311-4.
[40] T. Ojala, M. Pietikäinen, and D. Harwood, “A comparative study of texture measures with classification based on featured distributions,” *Pattern Recognition*, vol. 29, no. 1, pp. 51–59, Jan. 1996, doi: 10.1016/0031-3203(95)00067-4.
[41] C. Montagne, A. Kodewitz, V. Vigneron, V. Giraud, and S. Lelandais, “3D Local Binary Pattern for PET Image Classification by SVM - Application to Early Alzheimer Disease Diagnosis,” May 2021, pp. 145–150. Accessed: May 25, 2021. [Online]. Available: https://www.scitepress.org/Link.aspx?doi=10.5220/0004226201450150
[42] M. Pietikäinen, A. Hadid, G. Zhao, and T. Ahonen, “Computer Vision Using Local Binary Patterns,” in *Computer Vision Using Local Binary Patterns*, M. Pietikäinen, A. Hadid, G. Zhao, and T. Ahonen, Eds. London: Springer, 2011, pp. E1–E2. doi: 10.1007/978-0-85729-748-8_14.
[43] Y. Zhao, W. Jia, R.-X. Hu, and H. Min, “Completed robust local binary pattern for texture classification,” *Neurocomputing*, vol. 106, pp. 68–76, Apr. 2013, doi: 10.1016/j.neucom.2012.10.017.
[44] T. Ojala, M. Pietikainen, and T. Maenpaa, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” *IEEE Transactions on*

Pattern Analysis and Machine Intelligence, vol. 24, no. 7, pp. 971–987, Jul. 2002, doi: 10.1109/TPAMI.2002.1017623.

[45] D. Huang, C. Shan, M. Ardabilian, Y. Wang, and L. Chen, “Local Binary Patterns and Its Application to Facial Image Analysis: A Survey,” *IEEE Trans. Syst., Man, Cybern. C*, vol. 41, no. 6, pp. 765–781, Nov. 2011, doi: 10.1109/TSMCC.2011.2118750.

[46] G. Thibault *et al.*, “Texture Indexes and Gray Level Size Zone Matrix Application to Cell Nuclei Classification,” presented at the 10th International Conference on Pattern Recognition and Information Processing, Nov. 2009.

[47] A. Larroza, V. Bodí, and D. Moratal, “Texture Analysis in Magnetic Resonance Imaging: Review and Considerations for Future Applications,” 2016. doi: 10.5772/64641.

[48] A. Destrero, S. Mosci, C. De Mol, A. Verri, and F. Odone, “Feature selection for high-dimensional data,” *Comput Manag Sci*, vol. 6, no. 1, Art. no. 1, Feb. 2009, doi: 10.1007/s10287-008-0070-7.

[49] L. Yu and H. Liu, “Feature Selection for High-Dimensional Data: A Fast Correlation-Based Filter Solution,” in *Proceedings, Twentieth International Conference on Machine Learning*, Dec. 2003, pp. 856–863. Accessed: May 25, 2021. [Online]. Available: https://asu.pure.elsevier.com/en/publications/feature-selection-for-high-dimensional-data-a-fast-correlation-ba

[50] G. Chandrashekar and F. Sahin, “A survey on feature selection methods,” *Comput. Electr. Eng.*, vol. 40, no. 1, Art. no. 1, Jan. 2014, doi: 10.1016/j.compeleceng.2013.11.024.

[51] H. Liu and L. Yu, “Yu, L.: Toward Integrating Feature Selection Algorithm for Classification and Clustering. IEEE Transaction on Knowledge and Data Engineering 17(4), 491-502,” *IEEE Transactions on Knowledge and Data Engineering - TKDE*, vol. 17, pp. 491–502, Apr. 2005, doi: 10.1109/TKDE.2005.66.

[52] F. Bagherzadeh Khiabani, A. Ramezankhani, F. Azizi, F. Hadaegh, E. Steyerberg, and D. Khalili, “A tutorial on variable selection for clinical prediction models: Feature selection methods in data-mining could improve the results,” *Journal of clinical epidemiology*, vol. 71, Oct. 2015, doi: 10.1016/j.jclinepi.2015.10.002.

[53] H. Zou and T. Hastie, “Regularization and Variable Selection via the Elastic Net,” *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005.

[54] Van der Kooij, A.J. and Meulman, J.J., “Regularization with Ridge penalties, the Lasso, and the Elastic Net for Regressionwith Optimal Scaling Transformations,” 2006.

[55] “scikit-learn: machine learning in Python — scikit-learn 0.24.2 documentation,” May 24, 2021. https://scikit-learn.org/stable/ (accessed May 24, 2021).

[56] S. M. Saqlain *et al.*, “Fisher score and Matthews correlation coefficient-based feature subset selection for heart disease diagnosis using support vector machines,” *Knowl Inf Syst*, vol. 58, no. 1, Art. no. 1, Jan. 2019, doi: 10.1007/s10115-018-1185-y.

[57] Q. Gu, Z. Li, and J. Han, “Generalized Fisher Score for Feature Selection,” *arXiv:1202.3725 [cs, stat]*, Feb. 2012, Accessed: May 24, 2021. [Online]. Available: http://arxiv.org/abs/1202.3725

[58] E. Yılmaz, “An Expert System Based on Fisher Score and LS-SVM for Cardiac Arrhythmia Diagnosis,” *Computational and Mathematical Methods in Medicine*, vol. 2013, p. e849674, Jun. 2013, doi: 10.1155/2013/849674.

[59] J. Brownlee, “Recursive Feature Elimination (RFE) for Feature Selection in Python,” *Machine Learning Mastery*, May 24, 2020. https://machinelearningmastery.com/rfe-feature-selection-in-python/ (accessed May 24, 2021).

[60] P. M. Granitto, C. Furlanello, F. Biasioli, and F. Gasperi, “Recursive feature elimination with random forest for PTR-MS analysis of agroindustrial products,” *Chemometrics and Intelligent Laboratory Systems*, vol. 83, no. 2, Art. no. 2, Sep. 2006, doi: 10.1016/j.chemolab.2006.01.007.

[61] B. F. Darst, K. C. Malecki, and C. D. Engelman, “Using recursive feature elimination in random forest to account for correlated variables in high dimensional data,” *BMC Genetics*, vol. 19, no. 1, Art. no. 1, Sep. 2018, doi: 10.1186/s12863-018-0633-8.

[62] G. Langs *et al.*, “Machine learning: from radiomics to discovery and routine,” *Radiologe*, vol. 58, no. Suppl 1, Art. no. Suppl 1, 2018, doi: 10.1007/s00117-018-0407-3.

[63] S. Raschka and V. Mirjalili, *Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow 2, 3rd Edition*. Packt Publishing Ltd, 2019.

[64] E. Y. Boateng and D. A. Abaye, “A Review of the Logistic Regression Model with Emphasis on Medical Research,” *Journal of Data Analysis and Information Processing*, vol. 7, no. 4, Art. no. 4, Sep. 2019, doi: 10.4236/jdaip.2019.74012.

[65] S. Wan, Y. Liang, and M. Guizani, “Deep Multi-Layer Perceptron Classifier for Behavior Analysis to Estimate Parkinson’s Disease Severity Using Smartphones,” *IEEE Access*, vol. PP, pp. 1–1, Jul. 2018, doi: 10.1109/ACCESS.2018.2851382.

[66] L. Breiman, “Random Forests,” *Machine Learning*, vol. 45, no. 1, Art. no. 1, Oct. 2001, doi: 10.1023/A:1010933404324.

[67] M. Kuhn and K. Johnson, *Applied Predictive Modeling*. New York, NY: Springer New York, 2013. doi: 10.1007/978-1-4614-6849-3.

[68] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. New York, NY, USA: Springer New York Inc., 2001.

[69] P. Geurts, D. Ernst, and L. Wehenkel, “Extremely randomized trees,” *Mach Learn*, vol. 63, no. 1, Art. no. 1, Apr. 2006, doi: 10.1007/s10994-006-6226-1.

[70] G. Ke *et al.*, “LightGBM: A Highly Efficient Gradient Boosting Decision Tree,” in *Advances in Neural Information Processing Systems*, 2017, vol. 30. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf

[71] J. N. Basalyga, C. A. Barajas, M. K. Gobbert, and J. Wang, “Performance Benchmarking of Parallel Hyperparameter Tuning for Deep Learning based Tornado Predictions,” May 2020, doi: 10.13016/m2mm94-pnfu.

[72] J. Wu, X.-Y. Chen, H. Zhang, L.-D. Xiong, H. Lei, and S.-H. Deng, “Hyperparameter Optimization for Machine Learning Models Based on Bayesian Optimizationb,” *Journal of Electronic Science and Technology*, vol. 17, no. 1, Art. no. 1, Mar. 2019, doi: 10.11989/JEST.1674-862X.80904120.

[73] M. Claesen and B. De Moor, “Hyperparameter Search in Machine Learning,” *arXiv:1502.02127 [cs, stat]*, Apr. 2015, Accessed: May 24, 2021. [Online]. Available: http://arxiv.org/abs/1502.02127

[74] B. Koçak, E. Ş. Durmaz, E. Ateş, and Ö. Kılıçkesmez, “Radiomics with artificial intelligence: a practical guide for beginners,” *Diagn Interv Radiol*, vol. 25, no. 6, Art. no. 6, Nov. 2019, doi: 10.5152/dir.2019.19321.

[75] J. C. Peeken *et al.*, “Radiomics in radiooncology - Challenging the medical physicist,” *Phys Med*, vol. 48, pp. 27–36, Apr. 2018, doi: 10.1016/j.ejmp.2018.03.012.

[76] Y. Zhong, J. He, and P. Chalise, “Nested and Repeated Cross Validation for Classification Model With High-Dimensional Data,” *Rev. colomb. estad.*, vol. 43, no. 1, Art. no. 1, Jan. 2020, doi: 10.15446/rce.v43n1.80000.

[77] J. Wainer and G. Cawley, “Nested cross-validation when selecting classifiers is overzealous for most practical applications,” *arXiv:1809.09446 [cs, stat]*, Sep. 2018, Accessed: May 24, 2021. [Online]. Available: http://arxiv.org/abs/1809.09446

[78] M. Brett *et al.*, *nipy/nibabel: 3.2.1*. Zenodo, 2020. Accessed: May 24, 2021. [Online]. Available: https://zenodo.org/record/4295521#.YKwcp6GxVPY

[79] C. Spearman, “The Proof and Measurement of Association between Two Things,” *The American Journal of Psychology*, vol. 15, no. 1, Art. no. 1, 1904, doi: 10.2307/1412159.

[80] C. Albon, “Drop Highly Correlated Features,” Dec. 20, 2017. https://chrisalbon.com/machine_learning/feature_selection/drop_highly_correlated_features/ (accessed May 25, 2021).

[81] J. Hauke and T. Kossowski, “Comparison of Values of Pearson’s and Spearman’s Correlation Coefficients on the Same Sets of Data,” *Quaestiones Geographicae*, vol. 30, pp. 87–93, Jun. 2011, doi: 10.2478/v10117-011-0021-1.

[82] A. K. J and A. S, “Aspect-based opinion ranking framework for product reviews using a Spearman’s rank correlation coefficient method,” *Information Sciences*, vol. 460–461, pp. 23–41, Sep. 2018, doi: 10.1016/j.ins.2018.05.003.

[83] S. Kumar, “Nested Cross-Validation — Hyperparameter Optimization and Model Selection,” *Medium*, Sep. 17, 2020. https://towardsdatascience.com/nested-cross-validation-hyperparameter-optimization-and-model-selection-5885d84acda (accessed May 25, 2021).

[84] T. Frodl and N. Skokauskas, “Meta-analysis of structural MRI studies in children and adults with attention deficit hyperactivity disorder indicates treatment effects,” *Acta Psychiatrica Scandinavica*, vol. 125, no. 2, pp. 114–126, 2012, doi: https://doi.org/10.1111/j.1600-0447.2011.01786.x.

[85] J. C. Peeken *et al.*, “A CT-based radiomics model to detect prostate cancer lymph node metastases in PSMA radioguided surgery patients,” *Eur J Nucl Med Mol Imaging*, vol. 47, no. 13, pp. 2968–2977, Dec. 2020, doi: 10.1007/s00259-020-04864-1.

[86] P. Gupta, “Regularization in Machine Learning,” *Medium*, Nov. 16, 2017. https://towardsdatascience.com/regularization-in-machine-learning-76441ddcf99a (accessed May 30, 2021).

[87] A. Desarda, “Understanding AdaBoost,” *Medium*, Jan. 17, 2019. https://towardsdatascience.com/understanding-adaboost-2f94f22d5bfe (accessed May 30, 2021).

[88] D. Tunnicliffe, “Extra Trees, please.,” *Medium*, Feb. 10, 2021. https://towardsdatascience.com/extra-trees-please-cec916e24827 (accessed May 30, 2021).
[89] J. Yu *et al.*, “Noninvasive IDH1 mutation estimation based on a quantitative radiomics approach for grade II glioma,” *Eur Radiol*, vol. 27, no. 8, pp. 3509–3522, Aug. 2017, doi: 10.1007/s00330-016-4653-3.
[90] Y. Zhang *et al.*, “Radiomics features on non-contrast-enhanced CT scan can precisely classify AVM-related hematomas from other spontaneous intraparenchymal hematoma types,” *Eur Radiol*, vol. 29, no. 4, pp. 2157–2165, Apr. 2019, doi: 10.1007/s00330-018-5747-x.
[91] H. S. Gabryś, F. Buettner, F. Sterzing, H. Hauswald, and M. Bangert, “Design and Selection of Machine Learning Methods Using Radiomics and Dosiomics for Normal Tissue Complication Probability Modeling of Xerostomia,” *Front. Oncol.*, vol. 8, 2018, doi: 10.3389/fonc.2018.00035.
[92] M. Hoogman *et al.*, “Subcortical brain volume differences of participants with ADHD across the lifespan: an ENIGMA collaboration,” *Lancet Psychiatry*, vol. 4, no. 4, pp. 310–319, Apr. 2017, doi: 10.1016/S2215-0366(17)30049-4.
[93] S. Smit, M. J. van Breemen, H. C. J. Hoefsloot, A. K. Smilde, J. M. F. G. Aerts, and C. G. de Koster, “Assessing the statistical validity of proteomics based biomarkers,” *Analytica Chimica Acta*, vol. 592, no. 2, pp. 210–217, Jun. 2007, doi: 10.1016/j.aca.2007.04.043.
[94] E. Anderssen, K. Dyrstad, F. Westad, and H. Martens, “Reducing over-optimism in variable selection by cross-model validation,” *Chemometrics and Intelligent Laboratory Systems*, vol. 84, no. 1, pp. 69–74, Dec. 2006, doi: 10.1016/j.chemolab.2006.04.021.
[95] F. Maleki, N. Muthukrishnan, K. Ovens, C. Reinhold, and R. Forghani, “Machine Learning Algorithm Validation: From Essentials to Advanced Applications and Implications for Regulatory Certification and Deployment,” *Neuroimaging Clin N Am*, vol. 30, no. 4, pp. 433–445, Nov. 2020, doi: 10.1016/j.nic.2020.08.004.
[96] G. Varoquaux, P. R. Raamana, D. A. Engemann, A. Hoyos-Idrobo, Y. Schwartz, and B. Thirion, “Assessing and tuning brain decoders: Cross-validation, caveats, and guidelines,” *NeuroImage*, vol. 145, pp. 166–179, Jan. 2017, doi: 10.1016/j.neuroimage.2016.10.038.
[97] X. Tan and B. Triggs, “Enhanced Local Texture Feature Sets for Face Recognition Under Difficult Lighting Conditions,” *IEEE Transactions on Image Processing*, vol. 19, no. 6, Art. no. 6, Jun. 2010, doi: 10.1109/TIP.2010.2042645.

Appendix

A. Code of 3D LBP feature extraction

```
# -*- coding: utf-8 -*-
"""
LBP feature extraction from 3D image
"""
__author__ = "Nasibeh Mohammadi"
__email__ = "nasibeh.mohammadi@gmail.com"
__date__ = "10 March 2021"

import numpy as np
import pandas as pd
import six
from collections import Counter
import nibabel as nib

class LBPFeature():
    """
    Extract LBP features of a 3d image with regards to the given mask.

    For fast computation, first we make shift vectors including six vectors
    such as :
        x = 1, y=0, z=0
        x = -1, y=0, z=0
        x = 0, y=1, z=0
        x = 0, y=-1, z=0
        x = 0, y=0, z=1
        x = 0, y=0, z=-1
    By shifting the image with these shift vectors we can compare each cell
    with its direct neighbours

    In this code P=6 and R=1, the neighbours on diagonal were not considered.

    After calculating the LBP number for each cell in the masked area of image,
    the LBP feature vector to be used in radiomics study was constructed.

    :param image_name: the image file name and its corresponding path
    :param mask_name: the mask file name and its corresponding path

    """

    def __init__(self,image_name, mask_name):

        # load image and mask.
        image_load = nib.load(image_name)
        mask_load =  nib.load(mask_name)
        # get the array of image and mask .
        image=image_load.get_fdata()
        mask=mask_load.get_fdata()

        self.image=image
        self.mask=mask
        n,m,k=self.image.shape
        self.n=n
        self.m=m
        self.k=k
        # read the pattern  file which is based on rotation invariant concept.
        self.pattern= pd.read_csv('rotation_invariant_pattern.txt', sep = "\t",
                        converters={'rotation_invariant': lambda x: str(x)})

    def shift(self,dx,dy,dz):
        # extend the zone to all directions with zeros.
        extended_square=np.zeros((3*self.n,3*self.m,3*self.k),self.image.dtype)
```

```
        extended_square[self.n:self.n+self.n,self.m:self.m+self.m,
                        self.k:self.k+self.k]=self.image
        x=self.n+dy
        y=self.m-dx
        z=self.k-dz
        return extended_square[x:x+self.n,y:y+self.m,z:z+self.k]

    def feature_vector(self):
        # get the permutation of three numbers -1, 0, 1 to make shift vectors.
        x = [-1,1,0]
        y = [-1,1,0]
        z=[-1,1,0]
        x = pd.DataFrame(data={'x': x}, index=np.repeat(0, len(x)))
        y = pd.DataFrame({'y': y}, index=np.repeat(0, len(y)))
        z = pd.DataFrame({'z': z}, index=np.repeat(0, len(z)))
        # get all permutations stored in a new df.
        shift_vec =pd.merge(x, (pd.merge(y, z, left_index=True,
                        right_index=True)),left_index=True, right_index=True)

        # becuase we only need direct neighbors, we only want the shift vectors
        # that have only one value (either 1 or -1) and the two other values
        # should be zero.
        shift_vec=shift_vec[shift_vec.abs().sum(axis=1) == 1]

        # shift list are intensity values of cells when we shifted the image
        # based on the corresponding shift vector in the extended square that
        # we made earlier.
        shift_list = []
        for i in range(len(shift_vec)):
           dx= shift_vec.iloc[i]['x']
           dy=shift_vec.iloc[i]['y']
           dz=shift_vec.iloc[i]['z']
           shift_list.append(self.shift(dx,dy,dz))

        # calculate LBP for each cell
        result=np.zeros((self.n,self.m,self.k),self.image.dtype)
        for x_center in range(self.n):
            for y_center in range(self.m):
                for z_center in range(self.k):
                    # only calculating LBP for the mask region has value 1
                    if self.mask[x_center,y_center,z_center] == 1:
                        decimal_center=0
                        for j in range(len(shift_list)):
                        # Comparing the image with corresponding shifted area.
                            b=np.sign(
                                shift_list[j][x_center][y_center][z_center] -
                                self.image[x_center][y_center][z_center])
                            # assign 0 or 1 to the corresponding neighbour.
                            if b == -1:
                                sign=0
                            else:
                                sign= 1

                            decimal_center+=sign*(2**j)

                        # mapp each decimal value to the corresponding pattern
                        # based on rotation invariant concept.
                        if np.where((
                                self.pattern['original'] == decimal_center
                                ).values==True):
```

```
                    result[x_center][y_center][z_center]=self.pattern.loc[
                        self.pattern['original'] ==
                                decimal_center]['minvalue'].values[0]
    # remove the zero regions.
    mask_result= result[self.mask==1]

    # calculate each pattern frequency.
    frequency = Counter(mask_result)

    # make the feature vector
    freq_total = sum(frequency.values())
    lbp_feature_vect_dict ={}
    for (label, val) in six.iteritems(frequency):
        if np.where((self.pattern['original'] == label).values==True):
            key=self.pattern.loc[self.pattern['original']==
                                 label]['rotation_invariant'].values[0]
            freq_scaled = val/freq_total
            lbp_feature_vect_dict[key] = freq_scaled

    return lbp_feature_vect_dict
```

B. Rotation Invariant Table

The following table provides the content of the file named "rotation_invariant_pattern.txt" imported in the 3D LBP code. This file was constructed based on the rotation invariant concept showing each decimal number belongs to which pattern group. This file is used for labelling of the LBP features in the code (see Appendix A). "Original" denotes the decimal LBP value.

original	minvalue	Rotation_Invariant	original	minvalue	Rotation_Invariant
0	0	300	32	1	210
1	1	210	33	5	120
2	1	210	34	5	120
3	3	201	35	7	111
4	1	210	36	5	120
5	5	120	37	21	030
6	5	120	38	21	030
7	7	111	39	23	021
8	1	210	40	5	120
9	5	120	41	21	030
10	5	120	42	21	030
11	7	111	43	23	021
12	3	201	44	7	111
13	7	111	45	23	021
14	7	111	46	23	021
15	15	102	47	31	012
16	1	210	48	3	201
17	5	120	49	7	111
18	5	120	50	7	111
19	7	111	51	15	102
20	5	120	52	7	111
21	21	030	53	23	021
22	21	030	54	23	021
23	23	021	55	31	012
24	5	120	56	7	111
25	21	030	57	23	021
26	21	030	58	23	021
27	23	021	59	31	012
28	7	111	60	15	102
29	23	021	61	31	012
30	23	021	62	31	012
31	31	012	63	63	003

C. Modifications of Biorad feature extraction module

The modifications made in Biorad feature extraction [19] are surrounded by a red box.

```
# -*- coding: utf-8 -*-
#
# comparison_schemes.py
#

"""
Features extraction script.
"""

__author__ = "Ahmed Albuni"
__email__ = "ahmed.albuni@gmail.com"

__modifiedby__ = "Nasibeh Mohammadi"
__email__ =  "nasibeh.mohammadi@gmail.com"
__modifydate__ = "10 March 2021"

from LBP3d import LBPFeature

import argparse
import logging
from csv import DictWriter
from os import listdir
from os.path import isfile, join

import numpy as np
import pandas as pd
import SimpleITK as sitk
import six
from radiomics import (
    firstorder,
    glcm,
    gldm,
    glrlm,
    glszm,
    ngtdm,
    shape,
    shape2D,
)
from tqdm import tqdm

parser = argparse.ArgumentParser(description="Features extraction")
parser.add_argument(
    "-file", type=str, help="CSV parameters file name and " "path"
)
parser.add_argument(
    "-glcm_distance",
    type=str,
    help="list of distances, " "comma separated. " "default: 1",
)
parser.add_argument(
    "-ngtdm_distance",
    type=str,
    help="list of distances, " "comma separated. " "default 1",
)
parser.add_argument(
    "-gldm_distance",
    type=str,
    help="list of distances, " "comma separated. " "default 1",
)
parser.add_argument(
    "-gldm_a", type=int, help="Cutoff value for dependence, " "default: 0"
```

```
)

#  List of features groups available in pyradiomics package
#  This list match the input csv parameters file
FEATURES_LIST = (
    "shape",
    "first_order",
    "glszm",
    "glrlm",
    "ngtdm",
    "gldm",
    "glcm",
    "LBP"
)


def extract_radiomics_features(
    features_list,
    bin_width,
    images_path,
    masks_path=None,
    glcm_distance=None,
    ngtdm_distance=None,
    gldm_distance=None,
    gldm_a=0,
    output_file_name="output",
):
    """
    :param features_list: list of features to be extracted
    :param bin_width:
    :param images_path: The path that contains the images
    :param masks_path: The path of the masks, masks name should match the
    images names
    :param glcm_distance: A list of distances for GLCM calculations,
    default is [1]
    :param ngtdm_distance: List of integers. This specifies the distances
     between the center voxel and the neighbor, for which angles should be
      generated.
    :param gldm_distance: List of integers. This specifies the distances
     between the center voxel and the neighbor, for which angles should be
      generated.
    :param gldm_a:  integer, α cutoff value for dependence.
    A neighbouring voxel with gray level j is considered
    dependent on center voxel with gray level i if |i−j|≤α
    :param output_file_name: Name of the output csv file
    :return:
        """
    if glcm_distance is None:
        glcm_distance = [1]
    if ngtdm_distance is None:
        ngtdm_distance = [1]
    if gldm_distance is None:
        gldm_distance = [1]

    list_of_images = [
        f for f in listdir(images_path) if isfile(join(images_path, f))
    ]

    for i, img in tqdm(
        enumerate(list_of_images), total=len(list_of_images), unit="files"
    ):
```

```
image_name = images_path + img
image = sitk.ReadImage(image_name, sitk.sitkUInt8)

row = dict()
row["Name"] = img

#  If the mask is not available we create a dummy mask here that
#  covers the whole image
if masks_path is None:
    mask = np.zeros((image.get_fdata()).shape, int) + 1
    mask = sitk.GetImageFromArray(mask)

else:
    mask_name = masks_path + img
    mask = sitk.ReadImage(mask_name, sitk.sitkUInt8)

    # Shape features applied only when the mask is provided
    if "shape" in features_list:
        if len((sitk.GetArrayFromImage(image)).shape) == 2:
            shape_2d_f = shape2D.RadiomicsShape2D(
                image, mask, binWidth=bin_width
            )
            row.update(get_selected_features(shape_2d_f, "shape_2d"))
        else:
            shape_f = shape.RadiomicsShape(
                image, mask, binWidth=bin_width
            )
            row.update(get_selected_features(shape_f, "shape"))

if "first_order" in features_list:
    f_o_f = firstorder.RadiomicsFirstOrder(
        image, mask, binWidth=bin_width
    )
    row.update(get_selected_features(f_o_f, "first_order"))
if "glszm" in features_list:
    glszm_f = glszm.RadiomicsGLSZM(image, mask, binWidth=bin_width)
    row.update(get_selected_features(glszm_f, "glszm"))
if "glrlm" in features_list:
    glrlm_f = glrlm.RadiomicsGLRLM(image, mask, binWidth=bin_width)
    row.update(get_selected_features(glrlm_f, "glrlm"))
if "ngtdm" in features_list:
    for d in ngtdm_distance:
        ngtdm_f = ngtdm.RadiomicsNGTDM(
            image, mask, distances=[d], binWidth=bin_width
        )
        row.update(
            get_selected_features(
                ngtdm_f, "ngtdm", additional_param="_d_" + str(d)
            )
        )
if "gldm" in features_list:
    for d in gldm_distance:
        gldm_f = gldm.RadiomicsGLDM(
            image,
            mask,
            distances=[d],
            gldm_a=gldm_a,
            binWidth=bin_width,
        )
        row.update(
            get_selected_features(
```

```
                    gldm_f, "gldm", additional_param="_d_" + str(d)
                )
            )
        if "glcm" in features_list:
            for d in glcm_distance:
                glcm_f = glcm.RadiomicsGLCM(
                    image, mask, distances=[d], binWidth=bin_width
                )
                row.update(
                    get_selected_features(
                        glcm_f, "glcm", additional_param="_d_" + str(d)
                    )
                )
        if "LBP" in features_list:
                lbp_f=LBPFeature(image_name=image_name ,mask_name=mask_name).feature_vector()

                row.update(
                    get_selected_features(
                        lbp_f,"LBP"
                    )
                )

        if i == 0:
            create_output_file(output_file_name + ".csv", row.keys())
        add_row_of_data(output_file_name + ".csv", row.keys(), row)


def create_output_file(file_name, columns):
    with open(file_name, "w", newline="") as f:
        writer = DictWriter(f, fieldnames=columns)
        writer.writeheader()


def add_row_of_data(file_name, columns, row):
    with open(file_name, "a", newline="") as f:
        writer = DictWriter(f, fieldnames=columns)
        writer.writerow(row)


def get_selected_features(selected_feature, category, additional_param=None):

    data = {}
    if category=="LBP":
        for (key, val) in six.iteritems(selected_feature):
            key = category + "_" + str(key)
            data[key] = val
    else:
        selected_feature.execute()
        for (key, val) in six.iteritems(selected_feature.featureValues):
            key = category + "_" + key
            if additional_param is not None:
                key = key + additional_param
            data[key] = val

    return data


if __name__ == "__main__":

    logging.disable(logging.CRITICAL)
```

```
args = parser.parse_args()
glcm_d = args.glcm_distance
if glcm_d is not None:
    glcm_d = glcm_d.split(",")
ngtdm_d = args.ngtdm_distance
if ngtdm_d is not None:
    ngtdm_d = ngtdm_d.split(",")
gldm_d = args.gldm_distance
if gldm_d is not None:
    gldm_d = gldm_d.split(",")

gldm_a = args.gldm_a
if gldm_a is None:
    gldm_a = 0

f_list = pd.read_csv(args.file)

for index, row in f_list.iterrows():
    print("Output file: ", row["output_file_name"])
    feature = []
    for f in FEATURES_LIST:
        if row[f] == 1:
            feature.append(f)
    if type(row["mask_dir"]) is not str:
        mask_path = None
    else:
        mask_path = row["mask_dir"]
    extract_radiomics_features(
        feature,
        row["bin_width"],
        row["image_dir"],
        mask_path,
        output_file_name=row["output_file_name"],
        glcm_distance=glcm_d,
        ngtdm_distance=ngtdm_d,
        gldm_distance=gldm_d,
        gldm_a=gldm_a,
    )
```

D. Code for Removing Correlated Features

The code for removing highly correlated features [80] is as follows.

```
__author__ = "Chris Albon"
__email__ = "cralbon@gmail.com"

import pandas as pd
import numpy as np

df = pd.read_excel(file_name,index_col='ID')
traindata = df.drop('Label', axis=1)
labels= np.asarray(df['Label'])

corr_spearsman = traindata.corr(method='spearman')

# Create correlation matrix for the main dataframe to drop features
corr_matrix = corr_spearsman.abs()
# Select upper triangle of correlation matrix
upper = corr_matrix.where(np.triu(np.ones(corr_matrix.shape), k=1).astype(np.bool))
# Find index of feature columns with correlation greater than %95
to_drop = [column for column in upper.columns if any(upper[column] > 0.95)]

df_clean = traindata.drop(traindata[to_drop], axis=1)
```

E. RENT Configuration

❖ Hippocampus Experiment 1

```
# C parameters you would like to try
my_C_params =  [0.01, 0.1, 1.0, 10, 100]
# l1-strengths you would like to try
my_l1_ratios  =  np.arange(0, 1.1, 0.1)
# Apply RENT to whole dataset
analysis = RENT_Classification(data=traindata,
                               target=labels,
                               feat_names=traindata.columns,
                               C=my_C_params,
                               poly='OFF',
                               autoEnetParSel=False,
                               scoring='accuracy',
                               classifier='logreg',
                               testsize_range=(0.2, 0.2),
                               K=100,
                               l1_ratios = my_l1_ratios ,
                              random_state=0,
                               verbose = 0)
analysis.train()
```

```
selected_features = analysis.selectFeatures(tau_1_cutoff=0.2, tau_2_cutoff=0.2, tau_3_cutoff=0.975)
```

```
# C parameters you would like to try
my_C_params =  [0.01, 0.1, 1.0, 10, 100]
# l1-strengths you would like to try
my_l1_ratios  =  np.arange(0, 1.1, 0.1)
# Apply RENT to whole dataset
analysis = RENT_Classification(data=red_data,
                               target=labels,
                               feat_names=red_data.columns,
                               C=my_C_params,
                               poly='ON',
                               autoEnetParSel=False,
                               scoring='accuracy',
                               classifier='logreg',
                               testsize_range=(0.2, 0.2),
                               K=100,
                               l1_ratios = my_l1_ratios ,
                              random_state=0,
                               verbose = 0)
analysis.train()
```

```
select_feature = analysis.selectFeatures(tau_1_cutoff=0.9, tau_2_cutoff=0.9, tau_3_cutoff=0.975)
```

❖ Hippocampus Experiment 2

```
# C parameters you would like to try
my_C_params =  [0.01, 0.1, 1.0, 10, 100]
# l1-strengths you would like to try
my_l1_ratios  =  np.arange(0, 1.1, 0.1)
# Apply RENT to whole dataset
analysis = RENT_Classification(data=traindata,
                               target=labels,
                               feat_names=traindata.columns,
                               C=my_C_params,
                               poly='OFF',
                               autoEnetParSel=False,
                               scoring='accuracy',
                               classifier='logreg',
                               testsize_range=(0.2, 0.2),
                               K=100,
                               l1_ratios = my_l1_ratios ,
                              random_state=0,
                               verbose = 0)
analysis.train()
```

```
selected_features = analysis.selectFeatures(tau_1_cutoff=0.2, tau_2_cutoff=0.2, tau_3_cutoff=0.975)
```

```
# C parameters you would like to try
my_C_params =  [0.01, 0.1, 1.0, 10, 100]
# l1-strengths you would like to try
my_l1_ratios  =  np.arange(0, 1.1, 0.1)
# Apply RENT to whole dataset
analysis = RENT_Classification(data=red_data,
                               target=labels,
                               feat_names=red_data.columns,
                               C=my_C_params,
                               poly='ON',
                               autoEnetParSel=False,
                               scoring='accuracy',
                               classifier='logreg',
                               testsize_range=(0.2, 0.2),
                               K=100,
                               l1_ratios = my_l1_ratios ,
                              random_state=0,
                               verbose = 0)
analysis.train()
```

```
select_feature = analysis.selectFeatures(tau_1_cutoff=0.5, tau_2_cutoff=0.5, tau_3_cutoff=0.975)
```

❖ Hippocampus Experiment 3

```
# C parameters you would like to try
my_C_params =  [0.01, 0.1, 1.0, 10, 100]
# l1-strengths you would like to try
my_l1_ratios  =  np.arange(0, 1.1, 0.1)
# Apply RENT to whole dataset
analysis = RENT_Classification(data=df_clean,
                               target=labels,
                               feat_names=df_clean.columns,
                               C=my_C_params,
                               poly='OFF',
                               autoEnetParSel=False,
                               scoring='accuracy',
                               classifier='logreg',
                               testsize_range=(0.2, 0.2),
                               K=100,
                               l1_ratios = my_l1_ratios ,
                              random_state=0,
                               verbose = 0)
analysis.train()
```

```
selected_features = analysis.selectFeatures(tau_1_cutoff=0.1, tau_2_cutoff=0.1, tau_3_cutoff=0.97
```

```
# C parameters you would like to try
my_C_params =  [0.01, 0.1, 1.0, 10, 100]
# l1-strengths you would like to try
my_l1_ratios  =  np.arange(0, 1.1, 0.1)
# Apply RENT to whole dataset
analysis = RENT_Classification(data=red_data,
                               target=labels,
                               feat_names=red_data.columns,
                               C=my_C_params,
                               poly='ON',
                               autoEnetParSel=False,
                               scoring='accuracy',
                               classifier='logreg',
                               testsize_range=(0.2, 0.2),
                               K=100,
                               l1_ratios = my_l1_ratios ,
                              random_state=0,
                               verbose = 0)
analysis.train()
```

```
select_feature = analysis.selectFeatures(tau_1_cutoff=0.8, tau_2_cutoff=0.8, tau_3_cutoff=0.975)
```

❖ Caudate Experiment 1

```
# C parameters you would like to try
my_C_params =  [1.0, 10, 100]
# l1-strengths you would like to try
my_l1_ratios  =  np.arange(0, 1.1, 0.1)
# Apply RENT to whole dataset
analysis = RENT_Classification(data=traindata,
                               target=labels,
                               feat_names=traindata.columns,
                               C=my_C_params,
                               poly='OFF',
                               autoEnetParSel=False,
                               scoring='accuracy',
                               classifier='logreg',
                               testsize_range=(0.2, 0.2),
                               K=100,
                               l1_ratios = my_l1_ratios ,
                              random_state=0,
                               verbose = 0)
analysis.train()
```

```
selected_features = analysis.selectFeatures(tau_1_cutoff=0.9, tau_2_cutoff=0.9, tau_3_cutoff=0.975)
```

❖ Caudate Experiment 2

```
# C parameters you would like to try
my_C_params =  [1.0, 10, 100]
# l1-strengths you would like to try
my_l1_ratios  =  np.arange(0, 1.1, 0.1)
# Apply RENT to whole dataset
analysis = RENT_Classification(data=traindata,
                               target=labels,
                               feat_names=traindata.columns,
                               C=my_C_params,
                               poly='OFF',
                               autoEnetParSel=False,
                               scoring='accuracy',
                               classifier='logreg',
                               testsize_range=(0.2, 0.2),
                               K=100,
                               l1_ratios = my_l1_ratios ,
                              random_state=0,
                               verbose = 0)
analysis.train()
```

```
selected_features = analysis.selectFeatures(tau_1_cutoff=0.2, tau_2_cutoff=0.2, tau_3_cutoff=0.975)
```

❖ Caudate Experiment 3

```
# C parameters you would like to try
my_C_params =  [1.0, 10, 100]
# l1-strengths you would like to try
my_l1_ratios  =  np.arange(0, 1.1, 0.1)
# Apply RENT to whole dataset
analysis = RENT_Classification(data=df_clean,
                               target=labels,
                               feat_names=df_clean.columns,
                               C=my_C_params,
                               poly='OFF',
                               autoEnetParSel=False,
                               scoring='accuracy',
                               classifier='logreg',
                               testsize_range=(0.2, 0.2),
                               K=100,
                               l1_ratios = my_l1_ratios ,
                              random_state=0,
                               verbose = 0)
analysis.train()
```

```
selected_features = analysis.selectFeatures(tau_1_cutoff=0.2, tau_2_cutoff=0.2, tau_3_cutoff=0.975)
```

❖ Pallidum Experiment 1

```
# C parameters you would like to try
my_C_params =  [1.0, 10, 100]
# l1-strengths you would like to try
my_l1_ratios  =  np.arange(0, 1.1, 0.1)
# Apply RENT to whole dataset
analysis = RENT_Classification(data=traindata,
                               target=labels,
                               feat_names=traindata.columns,
                               C=my_C_params,
                               poly='OFF',
                               autoEnetParSel=False,
                               scoring='accuracy',
                               classifier='logreg',
                               testsize_range=(0.2, 0.2),
                               K=100,
                               l1_ratios = my_l1_ratios ,
                              random_state=0,
                               verbose = 0)
analysis.train()
```

```
selected_features = analysis.selectFeatures(tau_1_cutoff=0.70, tau_2_cutoff=0.7, tau_3_cutoff=0.975)
```

❖ Pallidum Experiment 2

```
# C parameters you would like to try
my_C_params =  [1.0, 10, 100]
# l1-strengths you would like to try
my_l1_ratios  =  np.arange(0, 1.1, 0.1)
# Apply RENT to whole dataset
analysis = RENT_Classification(data=traindata,
                               target=labels,
                               feat_names=traindata.columns,
                               C=my_C_params,
                               poly='OFF',
                               autoEnetParSel=False,
                               scoring='accuracy',
                               classifier='logreg',
                               testsize_range=(0.2, 0.2),
                               K=100,
                               l1_ratios = my_l1_ratios ,
                              random_state=0,
                               verbose = 0)
analysis.train()
```

```
selected_features = analysis.selectFeatures(tau_1_cutoff=0.6, tau_2_cutoff=0.6, tau_3_cutoff=0.975)
```

❖ Pallidum Experiment 3

```
# C parameters you would like to try
my_C_params =  [1.0, 10, 100]
# l1-strengths you would like to try
my_l1_ratios  =  np.arange(0, 1.1, 0.1)
# Apply RENT to whole dataset
analysis = RENT_Classification(data=df_clean,
                               target=labels,
                               feat_names=df_clean.columns,
                               C=my_C_params,
                               poly='OFF',
                               autoEnetParSel=False,
                               scoring='accuracy',
                               classifier='logreg',
                               testsize_range=(0.2, 0.2),
                               K=100,
                               l1_ratios = my_l1_ratios ,
                              random_state=0,
                               verbose = 0)
analysis.train()
```

```
selected_features = analysis.selectFeatures(tau_1_cutoff=0.6, tau_2_cutoff=0.6, tau_3_cutoff=0.975)
```

❖ Putamen Experiment 1

```
# C parameters you would like to try
my_C_params =  [0.01, 0.1, 1.0, 10, 100]
# l1-strengths you would like to try
my_l1_ratios  =  np.arange(0, 1.1, 0.1)
# Apply RENT to whole dataset
analysis = RENT_Classification(data=traindata,
                               target=labels,
                               feat_names=traindata.columns,
                               C=my_C_params,
                               poly='OFF',
                               autoEnetParSel=False,
                               scoring='accuracy',
                               classifier='logreg',
                               testsize_range=(0.2, 0.2),
                               K=100,
                               l1_ratios = my_l1_ratios ,
                              random_state=0,
                               verbose = 0)
analysis.train()
```

```
selected_features = analysis.selectFeatures(tau_1_cutoff=0.9, tau_2_cutoff=0.9, tau_3_cutoff=0.975)
```

❖ Putamen Experiment 2

```
# C parameters you would like to try
my_C_params =  [1.0, 10, 100]
# l1-strengths you would like to try
my_l1_ratios  =  np.arange(0, 1.1, 0.1)
# Apply RENT to whole dataset
analysis = RENT_Classification(data=traindata,
                               target=labels,
                               feat_names=traindata.columns,
                               C=my_C_params,
                               poly='OFF',
                               autoEnetParSel=False,
                               scoring='accuracy',
                               classifier='logreg',
                               testsize_range=(0.2, 0.2),
                               K=100,
                               l1_ratios = my_l1_ratios ,
                              random_state=0,
                               verbose = 0)
analysis.train()
```

```
selected_features = analysis.selectFeatures(tau_1_cutoff=0.5, tau_2_cutoff=0.5, tau_3_cutoff=0.975)
```

❖ Putamen Experiment 3

```
# C parameters you would like to try
my_C_params =  [1.0, 10, 100]
# l1-strengths you would like to try
my_l1_ratios  =  np.arange(0, 1.1, 0.1)
# Apply RENT to whole dataset
analysis = RENT_Classification(data=df_clean,
                               target=labels,
                               feat_names=df_clean.columns,
                               C=my_C_params,
                               poly='OFF',
                               autoEnetParSel=False,
                               scoring='accuracy',
                               classifier='logreg',
                               testsize_range=(0.2, 0.2),
                               K=100,
                               l1_ratios = my_l1_ratios ,
                              random_state=0,
                               verbose = 0)
analysis.train()
```

```
selected_features = analysis.selectFeatures(tau_1_cutoff=0.5, tau_2_cutoff=0.5, tau_3_cutoff=0.975)
```

❖ Thalamus Experiment 1

```
# C parameters you would like to try
my_C_params =  [1.0, 10, 100]
# l1-strengths you would like to try
my_l1_ratios  =  np.arange(0, 1.1, 0.1)
# Apply RENT to whole dataset
analysis = RENT_Classification(data=traindata,
                               target=labels,
                               feat_names=traindata.columns,
                               C=my_C_params,
                               poly='OFF',
                               autoEnetParSel=False,
                               scoring='accuracy',
                               classifier='logreg',
                               testsize_range=(0.2, 0.2),
                               K=100,
                               l1_ratios = my_l1_ratios ,
                              random_state=0,
                               verbose = 0)
analysis.train()
```

```
selected_features = analysis.selectFeatures(tau_1_cutoff=0.45, tau_2_cutoff=0.45, tau_3_cutoff=0.975)
```

❖ Thalamus Experiment 2

```
# C parameters you would like to try
my_C_params =  [1.0, 10, 100]
# l1-strengths you would like to try
my_l1_ratios  =  np.arange(0, 1.1, 0.1)
# Apply RENT to whole dataset
analysis = RENT_Classification(data=traindata,
                               target=labels,
                               feat_names=traindata.columns,
                               C=my_C_params,
                               poly='OFF',
                               autoEnetParSel=False,
                               scoring='accuracy',
                               classifier='logreg',
                               testsize_range=(0.2, 0.2),
                               K=100,
                               l1_ratios = my_l1_ratios ,
                              random_state=0,
                               verbose = 0)
analysis.train()
```

```
selected_features = analysis.selectFeatures(tau_1_cutoff=0.4, tau_2_cutoff=0.4, tau_3_cutoff=0.975)
```

❖ Thalamus Experiment 3

```
# C parameters you would like to try
my_C_params =  [1.0, 10, 100]
# l1-strengths you would like to try
my_l1_ratios  =  np.arange(0, 1.1, 0.1)
# Apply RENT to whole dataset
analysis = RENT_Classification(data=df_clean,
                               target=labels,
                               feat_names=df_clean.columns,
                               C=my_C_params,
                               poly='OFF',
                               autoEnetParSel=False,
                               scoring='accuracy',
                               classifier='logreg',
                               testsize_range=(0.2, 0.2),
                               K=100,
                               l1_ratios = my_l1_ratios ,
                              random_state=0,
                               verbose = 0)
analysis.train()
```

```
selected_features = analysis.selectFeatures(tau_1_cutoff=0.3, tau_2_cutoff=0.3, tau_3_cutoff=0.975)
```

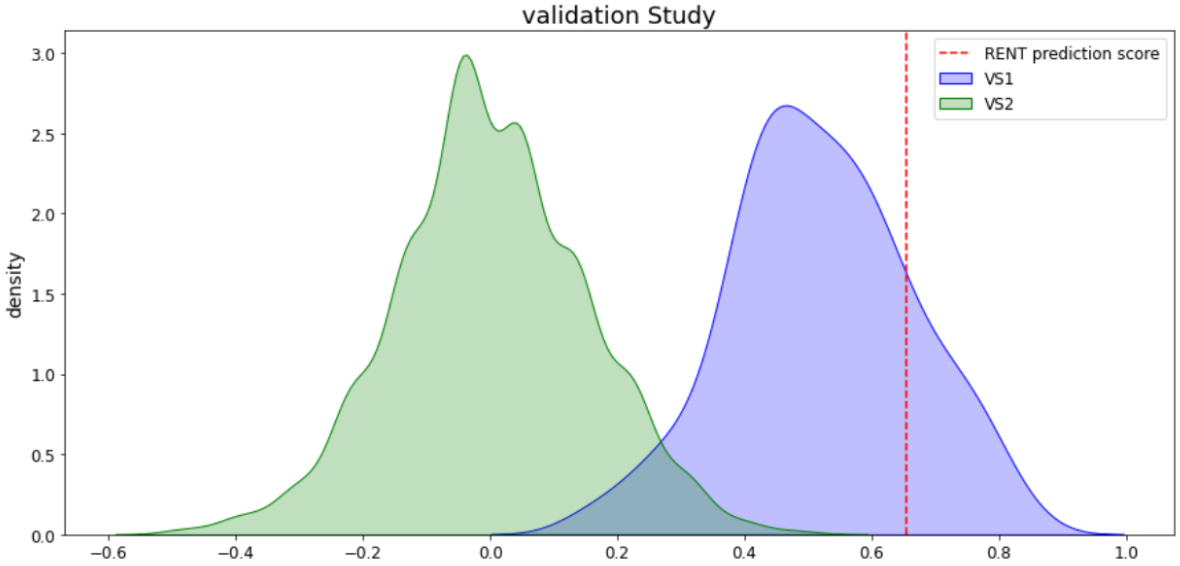
F. RENT Validation Study

❖ Hippocampus Experiment 1

```
mean VS1 0.5180824484545513
VS1: p-value for average score from random feature drawing:  1.9198797476744764e-16
With a significancelevel of  0.05  H0 is rejected.

------------------------------------------------------------

Mean VS2 0.0005871212121212112
VS2: p-value for score from permutation of test labels:  0.0
With a significancelevel of  0.05  H0 is rejected.
```

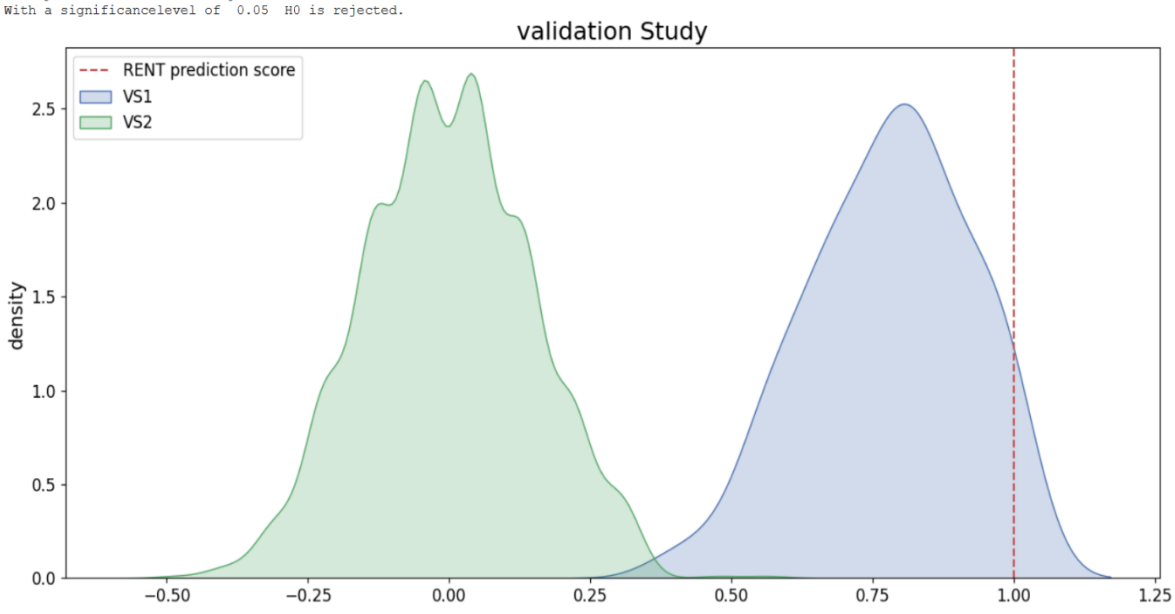


❖ Hippocampus Experiment 2

```
mean VS1 0.777728252829263
VS1: p-value for average score from random feature drawing:  1.4294739751429226e-28
With a significancelevel of  0.05  H0 is rejected.

------------------------------------------------------------

Mean VS2 -0.0018939393939393938
VS2: p-value for score from permutation of test labels:  0.0
With a significancelevel of  0.05  H0 is rejected.
```

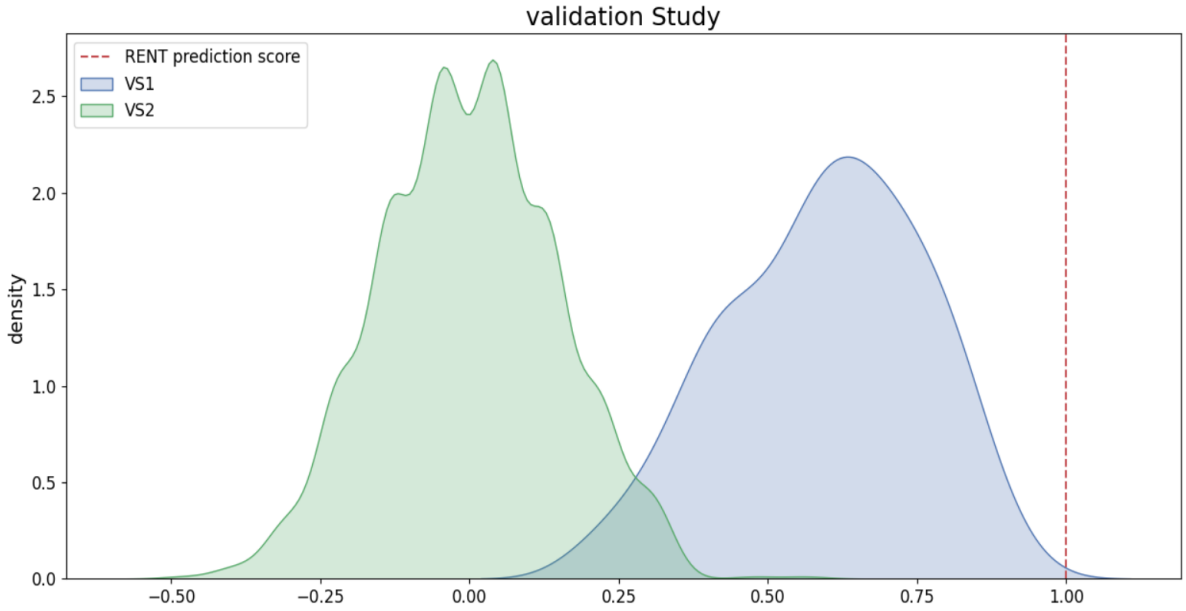


❖ Hippocampus Experiment 3

```
mean VS1 0.594749413648055
VS1: p-value for average score from random feature drawing:  1.089659133913559e-44
With a significancelevel of  0.05  H0 is rejected.

------------------------------------------------------------

Mean VS2 -0.0018939393939393938
VS2: p-value for score from permutation of test labels:  0.0
With a significancelevel of  0.05  H0 is rejected.
```

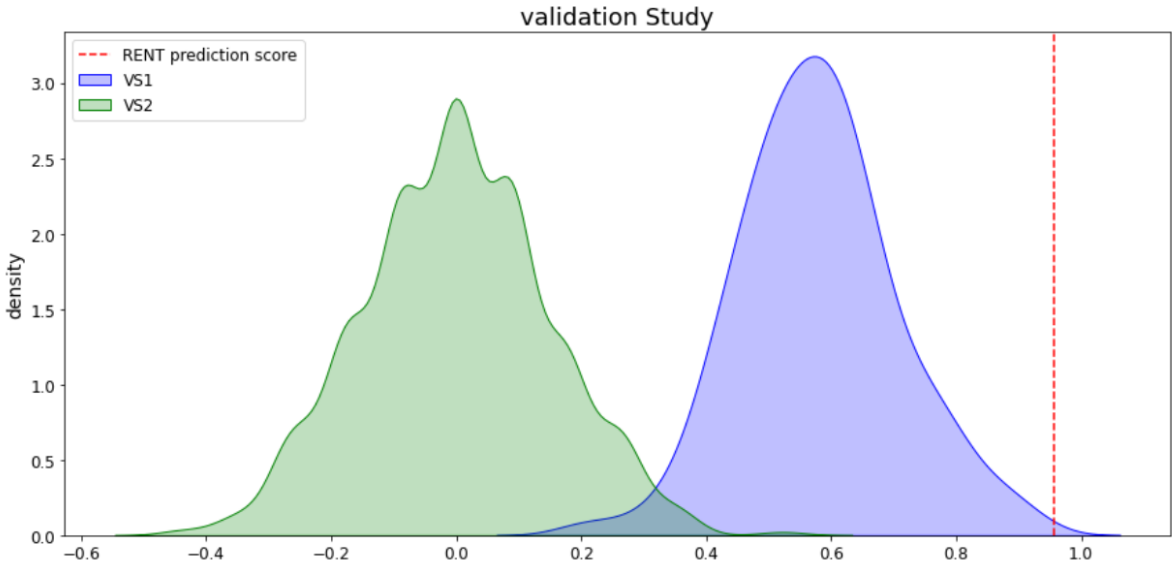


❖ Caudate Experiment 1

```
mean VS1 0.5784066259881445
VS1: p-value for average score from random feature drawing:  3.478564822365872e-52
With a significancelevel of  0.05  H0 is rejected.

------------------------------------------------------------

Mean VS2 -0.0002611164839335467
VS2: p-value for score from permutation of test labels:  0.0
With a significancelevel of  0.05  H0 is rejected.
```

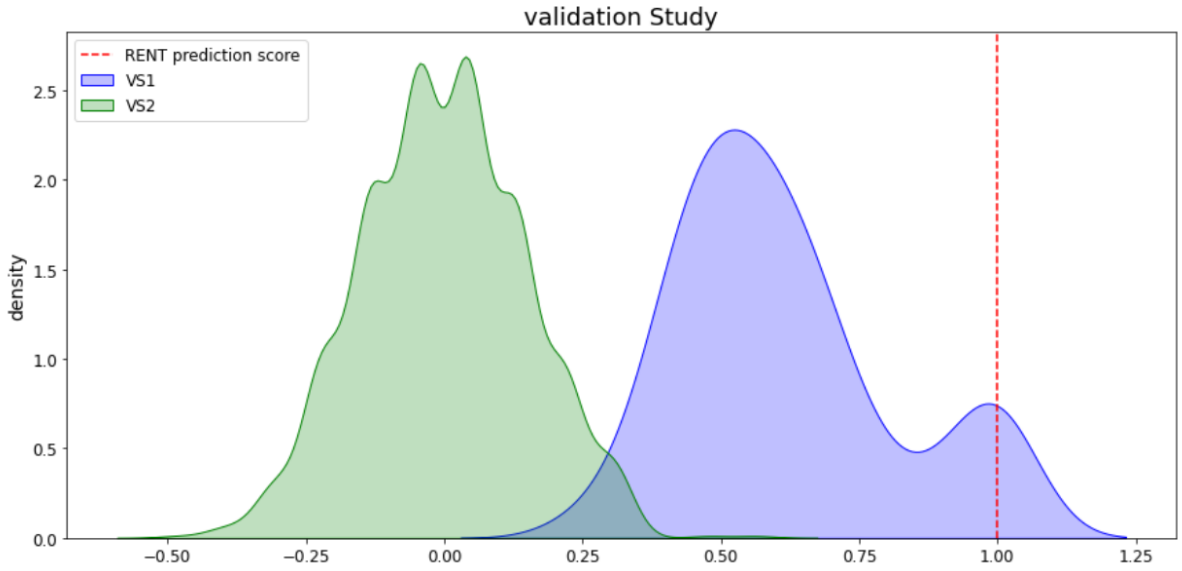


❖ Caudate Experiment 2

```
mean VS1 0.614081483372917
VS1: p-value for average score from random feature drawing:  1.1585279403771773e-36
With a significancelevel of  0.05  H0 is rejected.

------------------------------------------------------------

Mean VS2 -0.0018939393939393938
VS2: p-value for score from permutation of test labels:  0.0
With a significancelevel of  0.05  H0 is rejected.
```

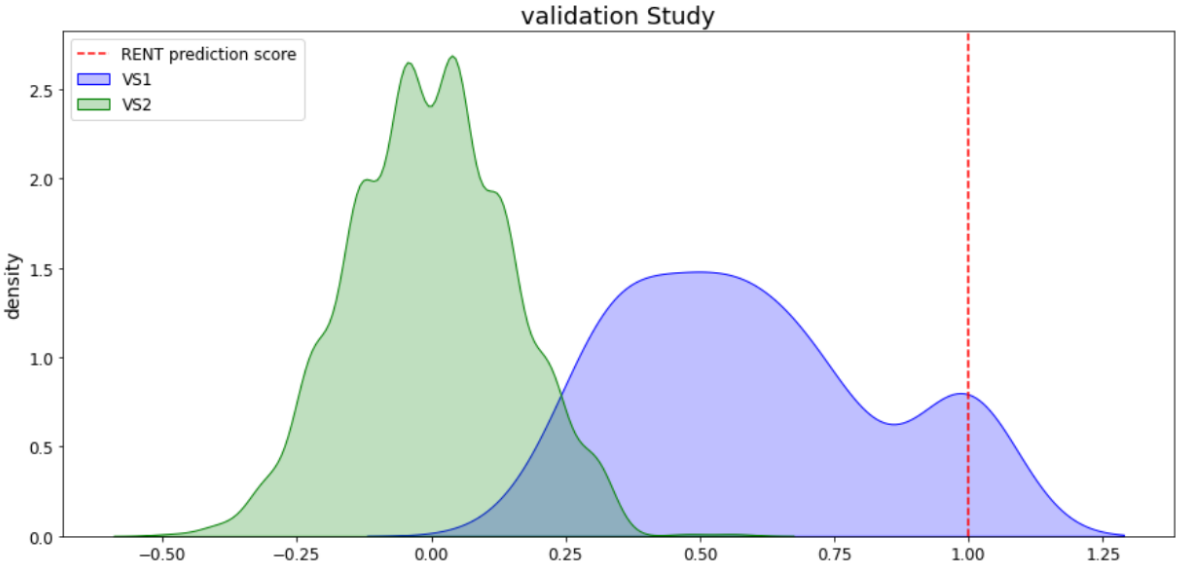


❖ Caudate Experiment 3

```
mean VS1 0.5970778728694368
VS1: p-value for average score from random feature drawing:  7.872541796034287e-31
With a significancelevel of  0.05  H0 is rejected.

------------------------------------------------------------

Mean VS2 -0.0018939393939393938
VS2: p-value for score from permutation of test labels:  0.0
With a significancelevel of  0.05  H0 is rejected.
```

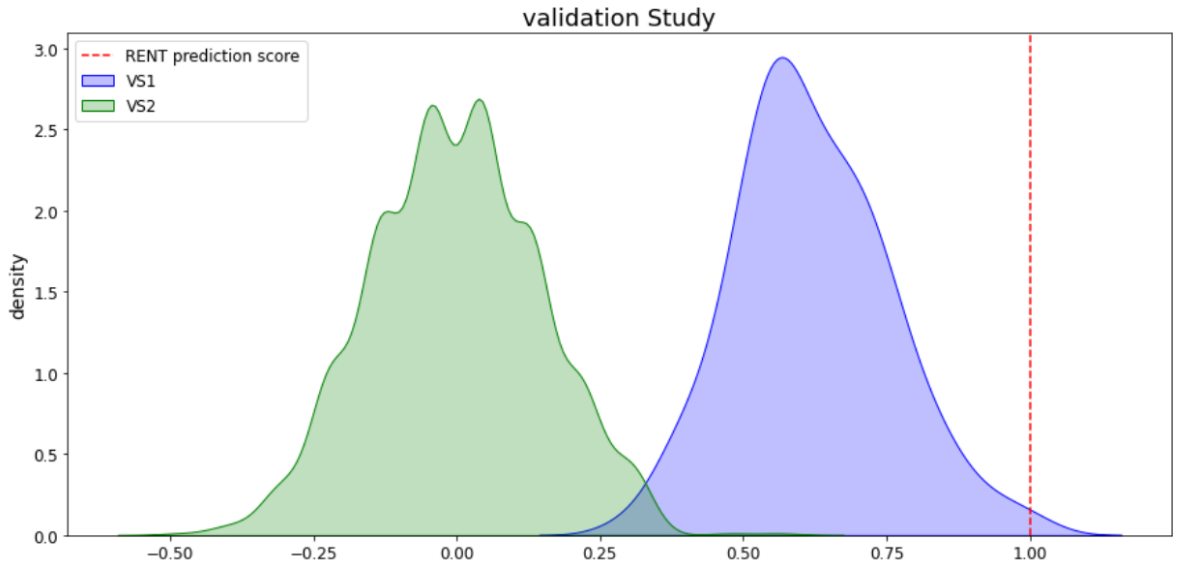


❖ Pallidum Experiment 1

```
mean VS1 0.6192402767754452
VS1: p-value for average score from random feature drawing:  7.346873693372442e-50
With a significancelevel of  0.05  H0 is rejected.

------------------------------------------------------------

Mean VS2 -0.0018939393939393938
VS2: p-value for score from permutation of test labels:  0.0
With a significancelevel of  0.05  H0 is rejected.
```

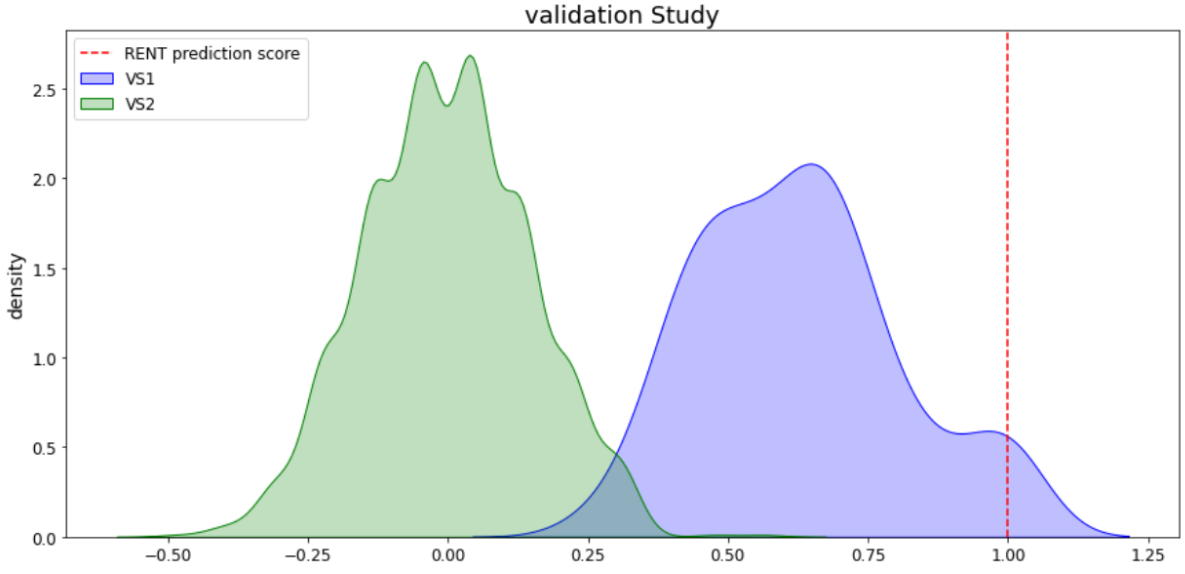


❖ Pallidum Experiment 2

```
mean VS1 0.6261694962656988
VS1: p-value for average score from random feature drawing:  3.7875153894758035e-38
With a significancelevel of  0.05  H0 is rejected.

------------------------------------------------------------

Mean VS2 -0.0018939393939393938
VS2: p-value for score from permutation of test labels:  0.0
With a significancelevel of  0.05  H0 is rejected.
```

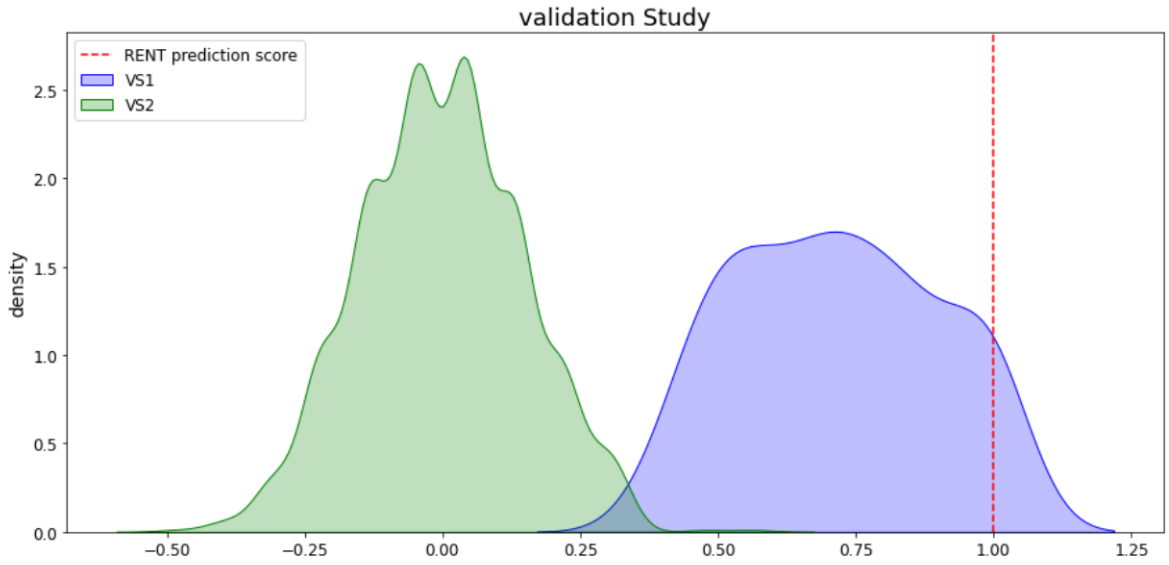


❖ Pallidum Experiment 3

```
mean VS1 0.7162836908382505
VS1: p-value for average score from random feature drawing:  1.355635847263875e-28
With a significancelevel of  0.05  H0 is rejected.

------------------------------------------------------------

Mean VS2 -0.0018939393939393938
VS2: p-value for score from permutation of test labels:  0.0
With a significancelevel of  0.05  H0 is rejected.
```

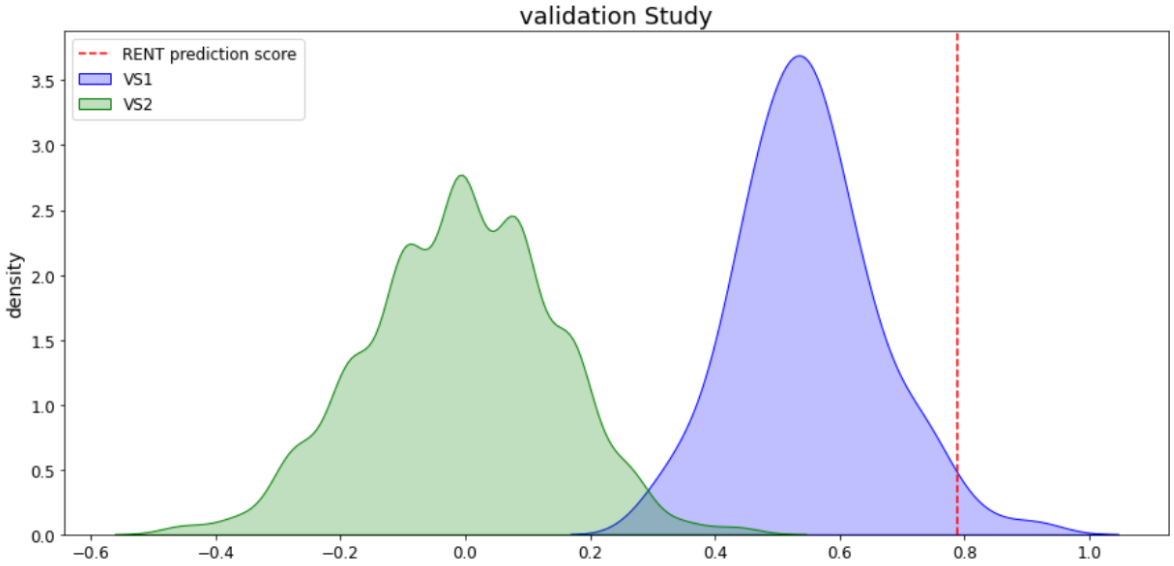


❖ Putamen Experiment 1

```
mean VS1 0.5471286822073967
VS1: p-value for average score from random feature drawing:  1.9549554032451516e-39
With a significancelevel of  0.05  H0 is rejected.

------------------------------------------------------------

Mean VS2 -0.007774102002228164
VS2: p-value for score from permutation of test labels:  0.0
With a significancelevel of  0.05  H0 is rejected.
```

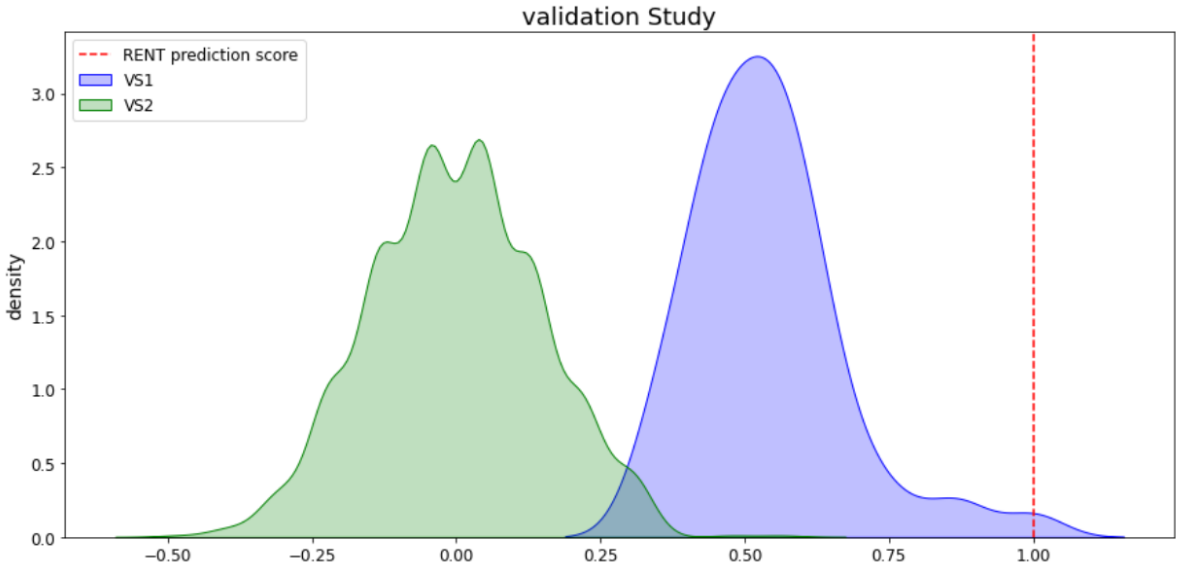


❖ Putamen Experiment 2

```
mean VS1 0.5376989705433316
VS1: p-value for average score from random feature drawing:  4.998161826006854e-58
With a significancelevel of  0.05  H0 is rejected.

-----------------------------------------------------------

Mean VS2 -0.0018939393939393938
VS2: p-value for score from permutation of test labels:  0.0
With a significancelevel of  0.05  H0 is rejected.
```

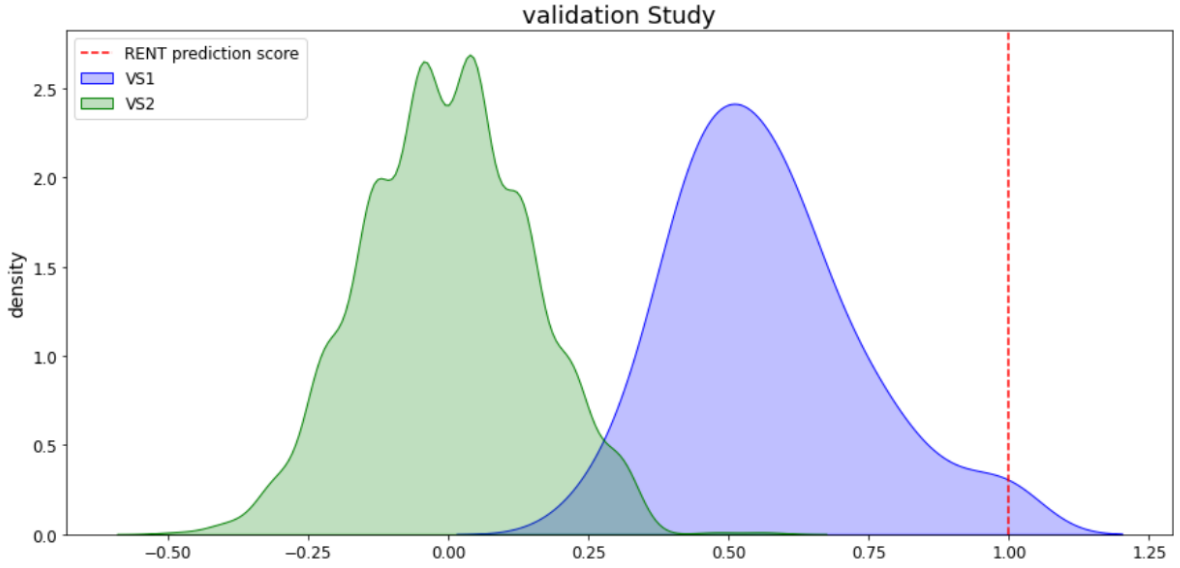


❖ Putamen Experiment 3

```
mean VS1 0.5675516407736545
VS1: p-value for average score from random feature drawing:  9.92352957905429e-46
With a significancelevel of  0.05  H0 is rejected.

-----------------------------------------------------------

Mean VS2 -0.0018939393939393938
VS2: p-value for score from permutation of test labels:  0.0
With a significancelevel of  0.05  H0 is rejected.
```

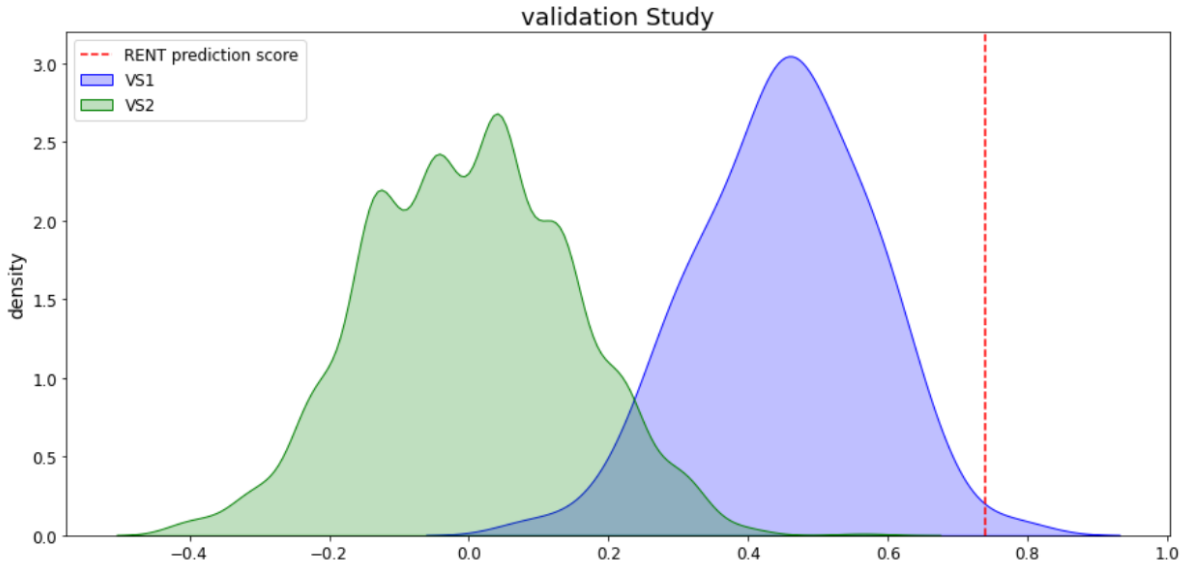


❖ Thalamus Experiment 1

```
mean VS1 0.4502088945168824
VS1: p-value for average score from random feature drawing:  4.324665682122485e-42
With a significancelevel of  0.05  H0 is rejected.

------------------------------------------------------------

Mean VS2 0.00019696969696969725
VS2: p-value for score from permutation of test labels:  0.0
With a significancelevel of  0.05  H0 is rejected.
```

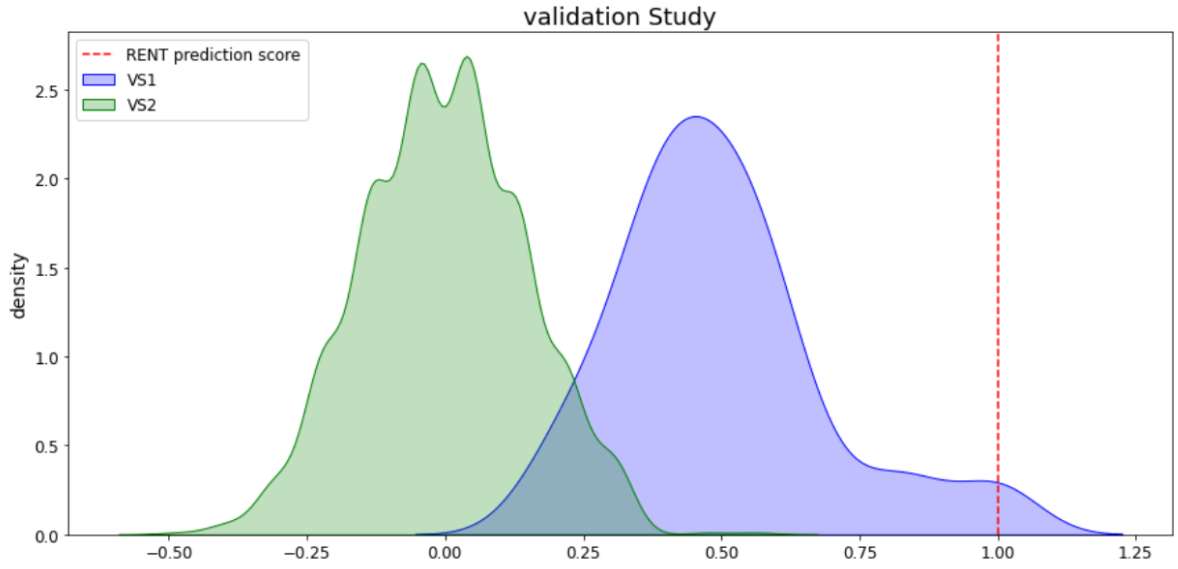


❖ Thalamus Experiment 2

```
mean VS1 0.49545674765812636
VS1: p-value for average score from random feature drawing:  2.046599211048232e-47
With a significancelevel of  0.05  H0 is rejected.

------------------------------------------------------------

Mean VS2 -0.0018939393939393938
VS2: p-value for score from permutation of test labels:  0.0
With a significancelevel of  0.05  H0 is rejected.
```

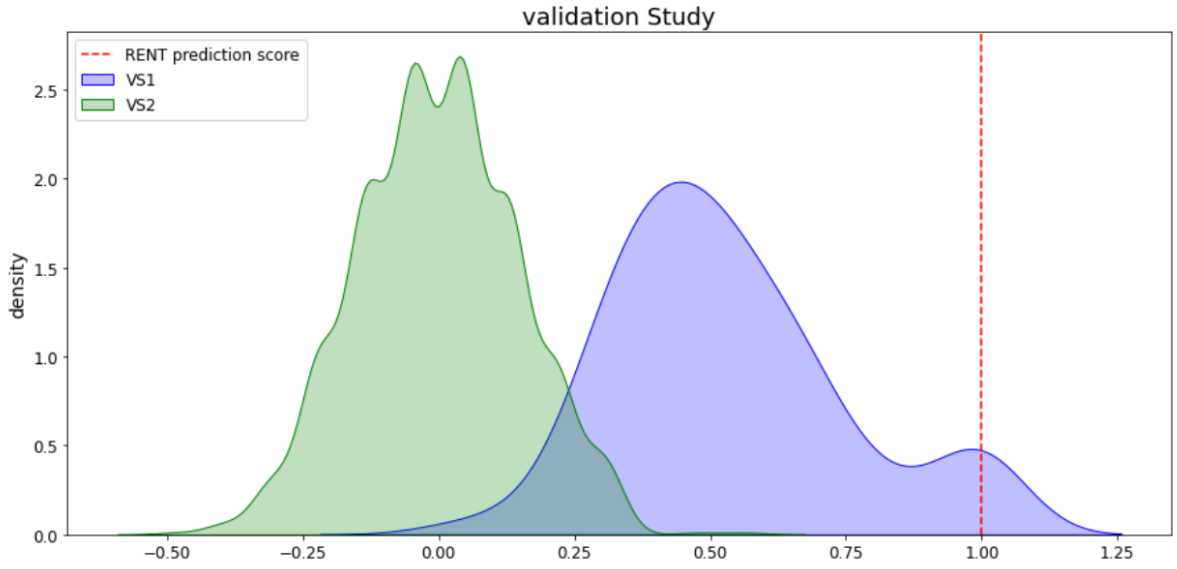


❖ Thalamus Experiment 3

```
mean VS1 0.5319023347461741
VS1: p-value for average score from random feature drawing:  7.421110230663567e-40
With a significancelevel of  0.05  H0 is rejected.

------------------------------------------------------------

Mean VS2 -0.0018939393939393938
VS2: p-value for score from permutation of test labels:  0.0
With a significancelevel of  0.05  H0 is rejected.
```



Norges miljø- og biovitenskapelige universitet
Noregs miljø- og biovitskapelege universitet
Norwegian University of Life Sciences

Postboks 5003
NO-1432 Ås
Norway