FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Neuromonitoring and Brain Imaging as predictors of outcome in patients with Intracerebral hemorrhage

Diogo Miguel Borges Gomes

Mestrado em Engenharia Informática e Computação

Supervisor: Miguel Velhote Correia

Second Supervisor: Maria Celeste Dias

October 23, 2023

Neuromonitoring and Brain Imaging as predictors of outcome in patients with Intracerebral hemorrhage

Diogo Miguel Borges Gomes

Mestrado em Engenharia Informática e Computação

October 23, 2023

Resumo

O sangramento no tecido cerebral, também designado como Hemorragia Intracerebral (HIC), é um dos tipos mais comuns de AVC com risco de vida, no qual o resultado pode ser melhorado por cuidados intensivos. Notoriamente, a individualidade dos pacientes que sofrem desses incidentes torna-o um dos mais difíceis de tratar e estabelecer tetos terapêuticos corretos. Assim, a dissertação visa explorar e identificar soluções que melhorem a certeza dos tratamentos de reabilitação adotados, minimizando os recursos terapêuticos desnecessários.

Para o controlo e análise dos estados fisiológicos e comportamentos fisiopatológicos desses pacientes neurocríticos, o monitorização em tempo real de parâmetros multimodais específicos do cérebro em UCIs tornou-se, ao longo do tempo, uma estratégia essencial. Como tal, ferramentas como o ICM+ integraram funcionalidades de Monitoramento Contínuo Multimodal (MCM) oferecendo armazenamento de alta resolução e análise em tempo real de dados de neuromonitorização.

Além da MCM, a interpretação de tomografias computorizadas (TACs) cerebrais pode ser utilizada como uma ferramenta diagnóstica não invasiva para avaliar a gravidade da lesão neurológica dos pacientes com HIC e estimar o tratamento de recuperação a ser realizado.

Para resolver o problema identificado, este projeto desenvolve um conjunto de modelos de aprendizagem supervisada capazes de identificar padrões nos dados obtidos na primeira semana de admissão do paciente na unidade de cuidados intensivos, através dos quais o prognóstico e a prevenção de possíveis desfechos podem ser realizadas. Para implementar esses modelos, conjuntos de dados retrospectivos, contendo aspectos clínicos, demográficos, do histórico médico, de neuroimagem, recolhidos através de MCM e colecionados entre 16 de janeiro de 2015 e 28 de novembro de 2021, são fornecidos pela Unidade de Cuidados Neurocríticos do Centro Hospitalar de S. João e são posteriormente anonimizados. A partir desses conjuntos de dados, para cumprir os objetivos do projeto, o trabalho faz a implementação de técnicas avançadas de análise de dados de machine learning destinadas a estimar modelos preditivos que permitam uma melhor previsão de desfechos 6 meses após admissão nos cuidados intensivos, aproveitando o poder de conjuntos de dados multimodais, incorporando dados tabulares e de imagem para melhorar a eficácia da previsão.

Os resultados desta tese revelaram que essas combinações de diferentes modalidades de dados proporcionaram boas performances. Com essas abordagens, são alcançados AUCs iguais ou superiores a 87,5% até 100% numa variedade de modelos desenvolvidos.

Concluiu-se que, embora os nossos resultados sejam promissores, alcançando valores notáveis de AUC, eles são preocupantes devido à potencial existência de overfitting nos modelos provocada pelo tamanho pequeno do conjunto de dados. Para avançar nesta pesquisa, trabalhos futuros devem utilizar conjuntos de dados maiores e mais diversos e explorar técnicas inovadoras de modelos baseados em imagens e de fusão de modelos. No entanto, as nossas descobertas abrem portas para aplicações práticas em cenários da vida real, oferecendo informações valiosas para profissionais médicos na otimização dos cuidados aos pacientes com HIC.

Abstract

The bleeding into the brain tissue, also designated as Intracerebral Hemorrhage (ICH), is one of the most common kinds of life-threatening strokes, in which the outcome can be improved by intensive care. Notoriously, the individuality of patients who suffer from these incidents makes it one of the most difficult to treat while establishing correct therapeutic ceilings. Thus, this dissertation aims to explore and identify solutions that improve the certainty of the adopted treatments while keeping unnecessary therapy resources to a minimum.

For the control and analysis of these neurocritical patients' physiological states and pathophysiological behaviours, real-time monitoring of brain-specific multimodal parameters in intensive care units (ICUs) has become, over time, an essential strategy. As such, bedside sources such as ICM+ have integrated Multimodal Continuous Monitoring (MCM) functionalities by offering high-resolution collection and real-time analysis of neuromonitoring data.

Besides MCM, the interpretation of brain computerized tomography (CT) scans can be utilized as a non-invasive diagnostic tool for evaluating the severity of the ICH patients' neurological injury and estimating the recovery treatment to be carried out.

To resolve the identified problem, this project develops a set of supervised learning models capable of identifying patterns in the data obtained in the first week of a patient's ICU admission, through which the prognostication and prevention of possible outcomes can be done. For implementing these models, retrospective datasets containing clinical, demographic, MCM, medical history, neuroimaging and follow-up aspects of the patients and collected between January 16, 2015, and November 28, 2021, are provided by the Neurocritical Care Unit of the S. João Hospital Centre and subsequently anonymized. From these datasets, to fulfil the project's objectives, the work revolves around implementing advanced machine learning data analysis techniques intended to estimate predictive models that allow for better prediction of 6-month outcomes by harnessing the power of multimodal datasets, incorporating both tabular and imaging data to enhance predictive effectiveness.

The results from this thesis revealed that these combinations of different data modalities provided good performances. With these approaches, AUCs higher or equal to 87.5% up to 100% are achieved on the plethora of developed models.

It was concluded that while our results demonstrate promise, achieving remarkable AUC values, they are tempered by concerns of potential overfitting in the models due to the dataset's small size. To advance this research, future work should leverage larger, more diverse datasets and explore innovative imaging-based models and fusion techniques. Nonetheless, our findings open the door to practical applications in real-life scenarios, offering valuable insights for medical professionals in optimizing ICH patient care.

Acknowledgements

I would like to express my gratitude to all those who, in one way or another, made this thesis a reality.

To my supervisor, Professor Miguel Correia, I would like to thank for the valuable contributions throughout the entire process. His guidance made all the difference.

To the team at the Neurocritical Care Unit of the S. João Hospital Centre, for providing the necessary tools for the completion of this project.

To my parents and my sister, for their unwavering trust in my progress and their unconditional support, always encouraging and assisting me in all areas of my life.

To my aunt, Amélia Morim (in memoriam), for her support and help in realizing my dream of entering my chosen university.

I would also like to thank the Faculty of Engineering of the University of Porto and its faculty members, who have demonstrated a commitment to the expected quality and excellence of education.

My sincere thanks to all,

Diogo Gomes

"When you feel that what you are doing is just a drop in the ocean, remember that the ocean would be less because of that missing drop"

Madre Teresa De Calcutá

Contents

List of Figures

List of Tables

Abreviaturas e Símbolos

ABP	Arterial Blood Pressure
AI	Artificial Intelligence
ANN	Artificial Neural Network
AUC	Area Under the Curve
CBF	Cerebral Blood Flow
CNN	Convolutional Neural Network
CT	Computerized Tomography
DICOM	Digital Imaging and Communications in Medicine
ECG	Eletrocardiogram
EEG	Electroencephalogram
FP	False Positive
FPR	False Positive Rate
FN	False Negative
GBC	Gradient Boosting Classifier
GCS	Glasgow Coma Scale/Score
HU	Housenfield Units
ICH	Intracerebral Hemorrhage
ICP	Intracranial Pressure
ICU	Intensive Care Unit
KD	Knowledge Discovery
MCM	Multimodal Continuous Monitoring
MLP	Multilayer Perceptron
MRI	Magnetic Resonance Imaging
mRS	Modified Rankin Scale
NIfTI	Neuroimaging Informatics Technology Initiative
LOO	Leave-One-Out
PRx	Pressure Reactivity Index
RBF	Radial Basis Function
RF	Random Forest
ROC	Receiver Operator Characteristic
rSO2L	Lung Oxygenation
rSO2R	Cerebral Oxygen Saturation
SAH	Subarachnoid Hemorrhage
SAPS	Simplified Acute Physiology Score
SMOTE	Synthetic minority oversampling technique
SVM	Support Vector Machine
TBI	Traumatic Brain Injuries
TP	True Positive
TPR	True Positive Rate
TN	True Negative

Chapter 1

Introduction

1.1 Context

The brain is one of the most complex organs of the human body, which, despite the protection of the skull bone cavity, can be subject to traumatic brain injuries and spontaneous occurrences within. ICH, also known as intraparenchymal bleed, is the sudden and acute bleeding into the brain's tissues and is one of the more common primary neurological diagnoses after ICU admissions, only behind subarachnoid hemorrhages (SAHs) [39]. This pathology is the second most frequent cause of strokes (15-30%) [36], usually leading to highly debilitating or fatal consequences.

Given these factors, diagnosed patients admitted to the neurocritical care units are typically placed with bedside multimodal continuous brain monitoring (MCM) to avoid secondary harmful effects. S. João Hospital Centre Neurocritical Care Unit routinely applies these monitoring techniques through ICM+. This software tool offers online (real-time) data collection capabilities, with highly informative displays for a variety of signals (e.g., intracranial pressure, i.e., ICP) and allows for advanced offline statistical analysis of the real-time data [44]. These monitoring capabilities are highly advantageous for detecting physiological changes in these patients, analysing their pathophysiological behaviour and, overall, for keeping a closer inspection on highly volatile conditions.

CT scans are the current standard for initial imaging of patients with either head traumas or strokes [13]. These scans are procedures that use X-rays and computer technology to produce images of the inside of the body, e.g., axial images of the brain (also known as slices). These provide the possibility of determining image features to predict the likelihood of hemorrhage expansion in ICH patients (and other intracranial hemorrhage pathologies) [24]:

- Hemorrhage size
- Hemorrhage density
- Hemorrhage shape
- Ventricular hemorrhagic extension

1.2 Problem and Objectives

Considering the high mortality and individuality of patients of the ICH pathology, it is challenging for medical professionals to establish therapeutic ceilings, thus leading, in specific scenarios, to the use of excessive resources for rehabilitating these patients.

To resolve the problem at hand, the thesis aims to provide a solution that identifies with greater certainty which patients should benefit from increased care by envisioning the following:

- Development of algorithms for prognosticating and detecting future consequences resulting from the ICH.
- Finding of associations of the patients' functional statuses in the sixth subsequent month with the following factors:
  - First-day CT scan features.
  - Invasive and non-invasive signal parameters obtained through the ICM+ MCM tool on the first and second days post-ICU admission.
  - Clinical, follow-up and demographic data from the patients.

For these goals to be fulfilled, the work will apply advanced machine-learning techniques to a multimodal dataset.

1.3 Motivation

Although the application of machine learning techniques for the prediction of patients' outcomes is now commonly used in the medical field, it is still a rare adoption for the ICH pathology, especially with the use of a multimodal dataset (i.e., containing categorical, MCM and imaging data).

State-of-the-art predictive models have proven highly effective and beneficial for ensuring that medical professionals provide patients with the most optimal possible care [49, 29]. Even though these models already offer good predictive capabilities, this thesis is also motivated by the need for more knowledge and research on the usefulness of a multimodal dataset for predicting the functional statuses of ICH patients and identifying dominant factors for prognostication.

Existing formal prognostication models of mortality and the functional outcome, such as the ICH score (described in Section 2.1.2), have not yet been found to be beneficial, with the subjective judgement of medical professionals being, in some occasions, more accurate than these scales in predicting ICH outcomes [23].

1.4 Document Structure

In addition to the introduction, this dissertation contains five more chapters:

- Chapter 2: In this chapter, a background on the concepts relevant to this thesis study is given with a focus on ICH and machine learning notions.

- Chapter 3: This chapter focuses on reviewing state-of-the-art works within the domain of the thesis theme to answer four research questions.
- Chapter 4: Describes the data provided for this investigation, the proposed solution for the problem of the dissertation and the methodologies to accomplish it.
- Chapter 5: In this division of the document, the results of the proposed solution are illustrated and discussed.
- Chapter 6: Chapter where the conclusions from the carried work are drawn.

Chapter 2

Domain Concepts

This chapter provides a comprehensive overview of the main theoretical concepts necessary and intrinsic to the project's development in 5 sections. The first section explores the ICH pathology, where the definition of scales/scores (ICH score, GCS, SAPS II) typically used for the prognosis of patients is encountered, as well as additionally delving into the MCM practice and the valuable information that can be extracted from it. The second section focuses on the knowledge discovery (KD) process in data, highlighting the crucial steps in uncovering useful insights from data. The second section describes artificial intelligence, a field directly related to two subdomains (approached in the two final sections) that are highly linked to the thesis. The following section covers machine learning concepts, mostly emphasizing predictive classification methods and algorithms using imaging and tabular data. The final section covers some evaluation metrics for assessing classification models' performance.

2.1 ICH Related Concepts

As was defined in section 1.1, ICH is the sudden bleeding into the brain tissues, namely in the brain parenchyma. The main causes for occurrences like this are the following [36]:

- Hypertension: High blood pressure levels make the rupture of brain arteries more likely.
- Age: More probable after the age of 55.
- Gender: More common in men.
- Previous history of stroke increases risk 23 times. [36]
- Alcohol and drug abuse.
- Liver disease.

Imaging modalities such as CT, magnetic resonance imaging (MRI), and ultrasound are commonly used for the detection and evaluation of ICH [4]:

- **CT:** the preferred initial imaging modality for diagnosing ICH due to its availability and speed.
- **MRI:** more useful for the detailed assessment of brain structure and function.
- **Ultrasound (namely, the Transcranial Doppler ultrasound):** Provides real-time information on blood flow in the brain and may be useful in monitoring changes in cerebral blood flow in response to ICH.

Treatments typically focus on stopping the bleeding and relieving the intracranial pressure (ICP) to prevent further damage to the brain, which can result from increased pressure within the skull induced by the extra blood.

Patients with ICH usually undergo clinical assessments for estimating severity scores, which give a probability of mortality by evaluating a handful of independent characteristics of the patients that are highly related to the ICH's severity. The ICH score and the Simplified Acute Physiology Score (SAPS) II constitute ICH patients' most commonly used scores.

2.1.1 Glasgow Coma Scale/Score (CGS)

The GCS is a standard scale for the severity assessment of the coma and impaired consciousness in the critically ill (commonly used for patients who suffer from ICH). It is highly related to the functional outcome of the neurocritical patients [22, 49], being of high importance for predicting neurological outcomes. For the determination of the score, medical professionals evaluate the following three factors:

- Eye Opening Response (1-4)
- Verbal Response (1-5)
- Motor Response (1-6)

The lower the overall score (3-15), the deeper the coma state.

2.1.2 ICH score

The ICH score is an outcome risk stratification scale developed from a logistic regression model, which included the five following characteristics determined to be independent predictors of 30-day mortality [22]:

- GCS on initial presentation (or after resuscitation)
- Age ≥ 80
- ICH Volume ≥ 30 mL
- Presence of bleeding in brain ventricles, i.e., intraventricular hemorrhage

- Infratentorial origin of hemorrhage

Each of these factors was assigned different weights based on the strength of association with the outcome. The total ICH score is the sum of these weights (Table 2.1).

Factor	**Weight**		
GCS	3-4 (+2)	5-12 (+1)	13-15 (0)
Age $\geq$ 80	True (+1)	False (0)	
ICH Volume $\geq$ 30 mL	True (+1)	False (0)	
Intraventricular hemorrhage	True (+1)	False (0)	
Infratentorial origin of hemorrhage	True (+1)	False (0)	
Total ICH Score	0-6		

Table 2.1: ICH Score Estimation

The mortality of the patients rises as the ICH score estimation increases. The original study associates different 30-day mortality probabilities with the various scores (0-6).

2.1.3 Simplified Acute Physiology Score (SAPS) II

SAPS II is a classification system widely used for assessing the severity of illness in the ICU. It is calculated with the worst results measured within 24 hours of the patient's ICU admission and has a value from 0 to 163 and a predicted mortality likelihood (0 % to 100 %).

The medical professionals calculate this score from 12 variables associated with the admitted patient, which include:

- Age
- Heart rate
- Body temperature
- Arterial pH
- White blood cell count
- GCS score
- Mean arterial pressure
- Respiratory rate
- Presence or absence of mechanical ventilation

Each variable is given a weighted score based on its relative contribution to mortality, with higher scores indicating a greater likelihood of death.

While SAPS II was originally developed to be used in general ICU patients, it has also been found to be useful for predicting outcomes for ICH patients.

Overall, the SAPS II score can be a tool to provide important information for clinicians when determining the best course of treatment for patients with ICH, as well as for predicting their likelihood of survival.

2.1.4 Modified Rankin Scale (mRS)

The mRS is a scale that measures the functional status, more specifically, the degree of disability or dependence of people who have suffered a stroke or other causes of neurological disability. It ranks the patients' mRS scores from 0 to 6 according to the following [48]:

- **0:** No symptoms at all.
- **1:** Minor symptoms that result in no significant disability.
- **2:** Slight disability, which makes the patient unable to carry out all previous activities.
- **3:** Moderate disability (patient requires help, but not for walking).
- **4:** Moderately severe disability (unable to walk and attend to bodily needs).
- **5:** Severe disability (bedridden with constant nursing care needs).
- **6:** Dead.

2.1.5 Multimodal Continuous Monitoring (MCM)

As was said in Section 1.1, bedside MCM is now a common practice in neurocritical care units. Through tools like ICM+, several brain and systemic signals (invasive and non-invasive) are saved into raw files in real-time, such as:

- Intracranial Pressure (ICP)
- Lung Oxygenation (rSO2L)
- Cerebral Oxygen Saturation (rSO2R)
- Cerebral Blood Flow (CBF)
- Electrocardiogram (ECG)
- Arterial Blood Pressure (ABP)
- CO2 Levels
- Electroencephalogram (EEG)

Other variables can be calculated in real-time besides the signals monitoring. One example is the pressure reactivity index (PRx), calculated at low frequencies, used to monitor cerebral auto-regulation by setting an interval for the CBF, in which the brain is safe according to relative changes in ABP and ICP. This variable has been proven to strongly correlate with the outcome of patients with traumatic brain injuries [16].

Besides the real-time data collection capabilities, ICM+ offers a user interface that allows offline advanced statistical analysis and calculations, which help set valuable parameters for detecting patient anomalies and for a deeper analysis of the patient's physiological volatilities [44].

2.1.6 CT Scan Images

As delineated in Section 1.1, it is meaningful that CT emerges as the primary modality commonly acquired in the initial phases of patient assessment. CT scans possess distinctive attributes that hold the potential to predict the likelihood of hemorrhage expansions. These scans typically unveil hyperdense accumulations of blood, frequently accompanied by the presence of hypodense edema in the surrounding areas [24].

In medical imaging, particularly when considering CT scans, images typically appear in specific formats that facilitate storage, analysis, and interpretation. Digital Imaging and Communications in Medicine (DICOM) and Neuroimaging Informatics Technology Initiative (NIfTI) are two prevalent formats utilized for medical images. DICOM is a standard format extensively used for various medical imaging modalities, including CT scans. It contains image data and associated metadata, such as patient information and imaging parameters, ensuring data preservation [1]. On the other hand, NIfTI is primarily utilized in neuroimaging and is particularly relevant for imaging modalities like MRI. It stores volumetric data, allowing the representation of three-dimensional structures. These formats not only enable interoperability across different imaging devices and software but also provide a standardized means for researchers and clinicians to access, analyze, and share medical image data effectively [2].

Additionally, CT scans often provide the image pixel values in Hounsfield Units (HU), a scale that measures the radiodensity of tissues. HU assign values according to the tissue types depicted in Table 2.2. This standardized scale aids in interpreting CT images, allowing medical professionals to differentiate between various tissues based on their radiodensity levels.

Hounsfield units	**Tissue**
> 1000	Bone, calcium, metal
100 to 600	Iodinated CT contrast
30 to 500	Punctate calcifications
60 to 100	Intracranial hemorrhage
35	Gray matter
25	White matter
20 to 40	Muscle, soft tissue
0	Water

-30 to -70	Fat
< -1000	Air

Table 2.2: Hounsfield scale brain CT numerical ranges. Source: [27]

2.2 Knowledge Discovery Process

The exponential increase in data collection has heightened the need to extract valuable information. Consequently, extracting meaningful knowledge from this vast amount of data has become complex, making the development of standard process models necessary for knowledge extraction. According to [15], these models should serve as a roadmap for organizations, guiding them in planning and executing projects while fulfilling the following objectives:

- Result in a product that is useful and, for this purpose, prevents data dredging, i.e., the blind application of data mining tasks to unstructured data.
- Possess a clear structure and approach that decision-makers (such as individuals with access to the KD system) can understand.
- Align with already established models avoiding the unnecessary reinvention of the wheel.

In essence, the KD model can be defined as the non-trivial process that transforms raw data into valid and novel patterns that are valuable in a specific domain. To establish this model, a series of steps must be followed, as shown in Figure 2.1.

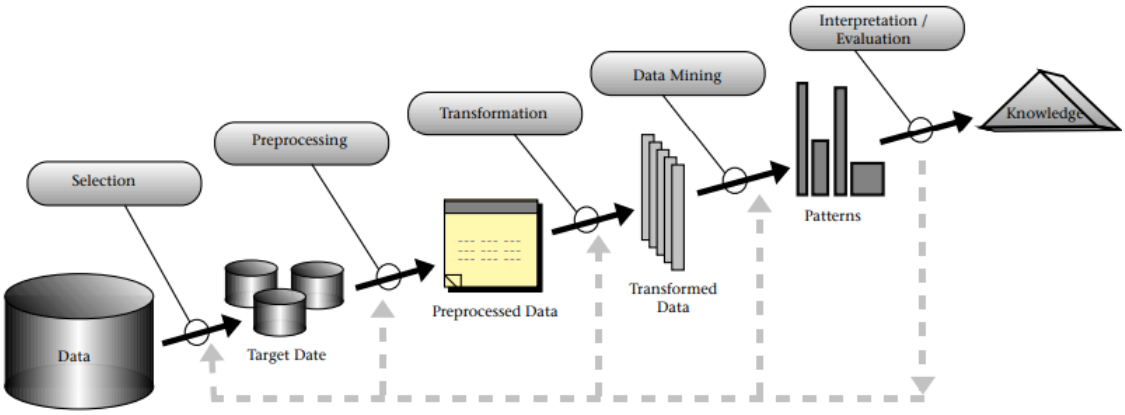


Figure 2.1: KD process model Source: [17]

The showcased model (by Fayyad et al. [17]) consists of the following 5 major steps:

- **Data selection** - In this phase, a target dataset is generated by selecting the relevant features (variables) according to the problem's domain knowledge.

- **Data preprocessing** - This step removes inconsistent data and deals with noise and missing values (data cleaning).
- **Data transformation** - Data is transformed into appropriate formats for the data mining phase (e.g., normalization and aggregation) and finds useful attributes (e.g., using feature selection).
- **Data mining** - Step where the patterns in the data are produced in the form of classification rules, clusters, etc. In the data mining task, an algorithm is picked, conforming to the goals defined for the problem at hand and the patterns that are being searched for.
- **Interpretation/Evaluation** - Phase where the better or most interesting patterns are selected based on specific metrics.

The previous model can end up not following a sequential order as loops can form between any two steps making the model versatile to changes in any of the phases. Upon following these steps, the discovered knowledge can then be incorporated into a system.

2.3 Artificial Intelligence

Artificial Intelligence (AI) is a rapidly growing field that is revolutionizing and is, in some cases, intrinsically contained in various domains, including healthcare. AI refers to the simulation of human behaviours in machines, which enable the performance of actions that usually require human cognitive capabilities:

- Perception (of the surrounding environment)
- Reasoning
- Learning
- Decision-making

AI can potentially transform the field of medicine by improving diagnosis and prognosis accuracy, treatments, and patient outcomes. In the context of this thesis, AI can help identify patterns and trends in the patients' data that may otherwise be difficult to detect for human clinicians. This can lead to earlier detection of hemorrhages, more accurate predictions of patients' outcomes and diagnoses, and personalized treatment plans based on patient individualities [38].

Within AI, diverse and interconnected fields that contribute to its advancement are encountered. Two notable subfields are machine learning and computer vision, which play key roles in harnessing the potential of AI technology.

2.4 Machine Learning Concepts

As already stated, machine learning is a field within the broader ecosystem of AI that holds immense significance in facilitating automated pattern recognition, prediction, and decision-making. By harnessing statistical techniques, machine learning algorithms can autonomously extract patterns from data, enabling the generation of models qualified with prediction and classification capacities. These are powerful tools for analysing and forecasting new data on acquired features. Furthermore, utilising advanced algorithms and statistical methodologies entrusts machine learning systems to adapt and better their performance over time, making them critical in domains reliant on data-driven insights and decision-making.

Within the healthcare field, machine learning has gained significant popularity, particularly in the analysis of medical images and patient data. For instance, it can be leveraged to train models on extensive datasets of brain images, enabling accurate classification into distinct categories. This exemplifies the profound impact of machine learning in aiding healthcare professionals with precise diagnoses and informed treatment decisions.

It is important to note that machine learning contains various subcategories, each with its applications. These categories widen the scope of machine learning and add versatility in solving diverse problems. Three of the main subcategories are supervised, unsupervised and deep learning.

2.4.1 Supervised Learning

Trains algorithms using labelled training data, allowing the generation of a model over time. The algorithm estimates its accuracy through a loss function by adjusting until the error is minimized, resulting in a function/model that receives input data and outputs a prediction. Supervised learning can be split into two different kinds of problems [7]:

- **Classification -** Aims to predict or classify the target variable, which is labelled with distinct values.
- **Regression -** The goal is to determine continuous values such as ages, prices, etc.

Some commonly utilized algorithms in supervised learning include the random forest (section 2.4.1.2), support vector machines (section 2.4.1.4), gradient boosting classifiers (section 2.4.1.3) and artificial neural networks (section 2.4.4), which are described in the following subsections. Notably, some of these algorithms use the ensemble learning (section 2.4.1.1) technique to enhance their predictive capabilities further.

2.4.1.1 Ensemble learning

Ensemble learning is a machine learning technique that combines multiple base learners (individual models) to improve the models' overall prediction accuracy and generalization performance. It bases itself on the idea that by merging the strengths of multiple models into a single one, the ensemble can achieve better results [11].

The main types of ensemble methods are the following:

- **Bagging:** Uses multiple base learners trained in different bootstrap samples (random samples with replacement) retrieved from the training data. Each base learner generates an individual prediction, and the final prediction is obtained through the aggregation of the forecasts of all base learners. The most common aggregation techniques comprise majority voting for classification tasks and averaging for regression.
- **Boosting:** Uses base learners trained sequentially, where each subsequent learner focuses more on the instances that were misclassified or have high errors by the previous learners. The predictions of all base learners are combined using weighted voting, where more weight is given to the predictions of more accurate learners.

2.4.1.2 Random Forest (RF)

The random forest algorithm is an ensemble learning technique that uses a collection of uncorrelated decision trees to make predictions through aggregations. This process is depicted in Figure 2.2.

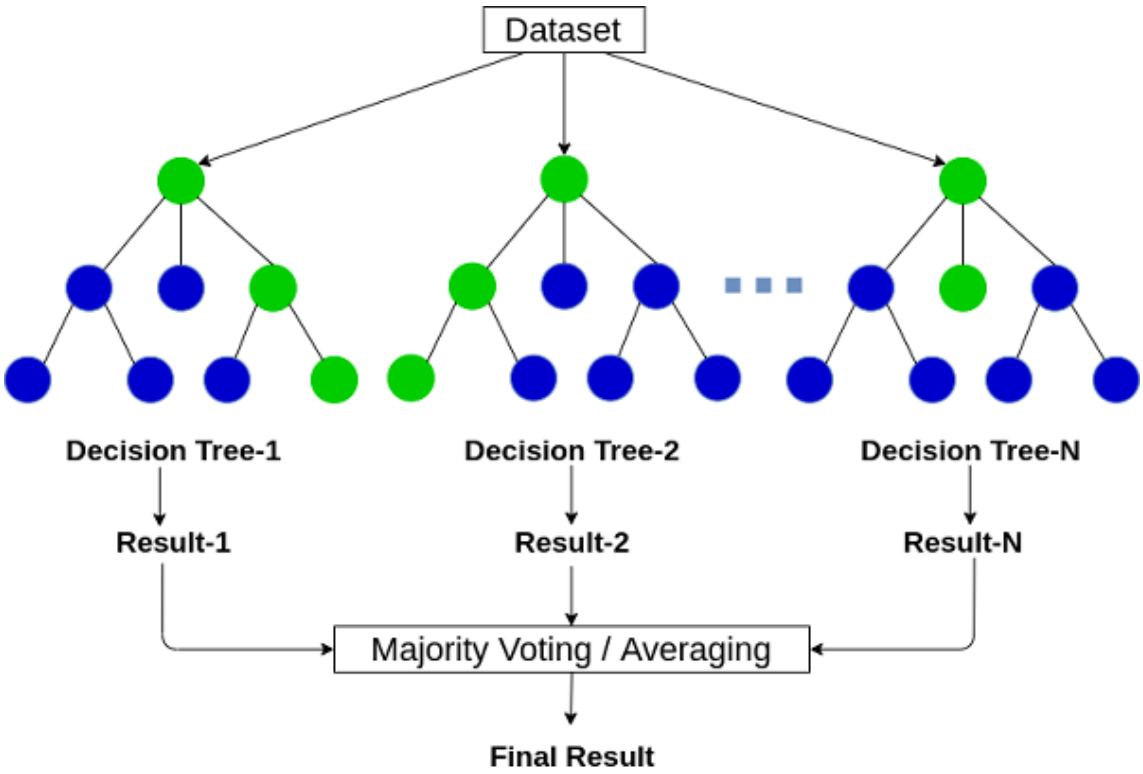


Figure 2.2: Random Forest algorithm illustration. Source: [43]

To create these decision trees, several steps are involved:

1. **Bootstrapping** - The algorithm randomly selects X data points from the dataset, allowing for replacement in each selection. As such, this strategy helps generate diverse random subsets of data.

2. **Unique tree training** - Each decision tree is trained distinctly. At each node of the tree, rather than choosing the best split, a subset of predictors (variables) is randomly sampled. The best split from this subset is then chosen.

3. **Prediction Aggregation**: The algorithm combines the predictions from individual trees using either majority voting for classification or averaging for regression. The aggregation result then yields the final prediction.

Since the RF follows a process of first performing bootstrapping and subsequently aggregating the different tree models' results into one, it can be considered an extension of the ensembling bagging method.

Other than the prediction value of the random forest, according to [28], extra information can be extracted from it:

- **Feature Importance** - It is estimated for each variable by verifying the sensibility of the prediction error when out-of-bag data (not included in the bootstrap sample) for that variable is changed while others remain unchanged. [9]

- **Proximity Measure** - Similarity between two data points. It can be useful in unsupervised learning scenarios with random forests or in identifying underlying structures within the data.

The RF algorithm stands out for its capacity to address overfitting issues, handle high-dimensional datasets by employing uncorrelated decision trees, and exploit the potential supplementary information. These distinct qualities have driven its adoption as a choice for classification and regression applications across various domains, including healthcare.

2.4.1.3 Gradient Boosting Classifier (GBC)

Gradient boosting models are techniques widely used for classification and regression tasks. They are an ensemble learning method that combines multiple weak models, typically decision trees, to create a strong predictive model [18]:

$$P(X) = \sum_{i=1}^{N} p_i(X) \tag{2.1}$$

where:

P = Gradient boosting model prediction
X = Features
p_i = Prediction function of the ith weak model

N = Amount of weak models

These models use the boosting ensemble technique by learning some prediction as a sum of a group of weak learners (e.g., decision trees) trained with progressive learning from mistakes made by previous learners.

From a broad perspective, GBC applies the ensemble boosting process by iteratively fitting new models to the errors (residuals) of the previous models. This methodology begins with a weak model, for instance, a decision tree containing only one leaf with the average of the values to be predicted (for regression), and succeeding models are built to correct and minimize the errors made by previous models on the overall prediction.

The use of a loss function is a fundamental aspect of these algorithms as it measures the discrepancies between the predicted and actual values:

$$L(y, p) \tag{2.2}$$

where:

L = Loss Function
y = Truth
p = Prediction

This function should be differentiable, as the gradient of the loss function gives each subsequent weak learner valuable information on which data point should be focused according to its impact on the overall value of the loss (negative of the residuals), which can be calculated as shown in Equation 2.3. This allows for harder-to-predict instances (with higher errors) to be emphasized.

$$\nabla L_n = -(r_{ni}, r_{n(i+1)}, \ldots, r_{nM}) \tag{2.3}$$

where:

r_{ni} = residual of the ith data point on the nth weak model
M = amount of data points

To build each new model, a technique called gradient descent is utilized. It computes the residuals with respect to the predicted values and uses this information to update the model parameters in a direction that minimizes the loss. The learning rate parameter controls the contribution of each model, preventing overfitting and ensuring convergence to minimal loss.

To exemplify how the final prediction with all the weak models is made, the following equation showcases how it can be obtained with 3 weak models:

$$P(X) = p_1(X) + \gamma_2 p_2(X) + \gamma_3 p_3(X) \tag{2.4}$$

where:

P = Overall gradient boosting prediction
p_i = Prediction of the ith weak model
γ_i = Learning rate of ith weak model
X = Features

The final prediction from a gradient-boosting model is obtained by aggregating the predictions of all the individual models. Typically, the predictions of these models are combined using weighted averaging or summation, as seen in Equation 2.4.

2.4.1.4 Support Vector Machine (SVM)

A support vector machine is a supervised learning method used to solve classification and regression problems. Its primary objective is to find the optimal hyperplanes that separate different classes of the target into distinct and disjoint regions (Figure 2.3).

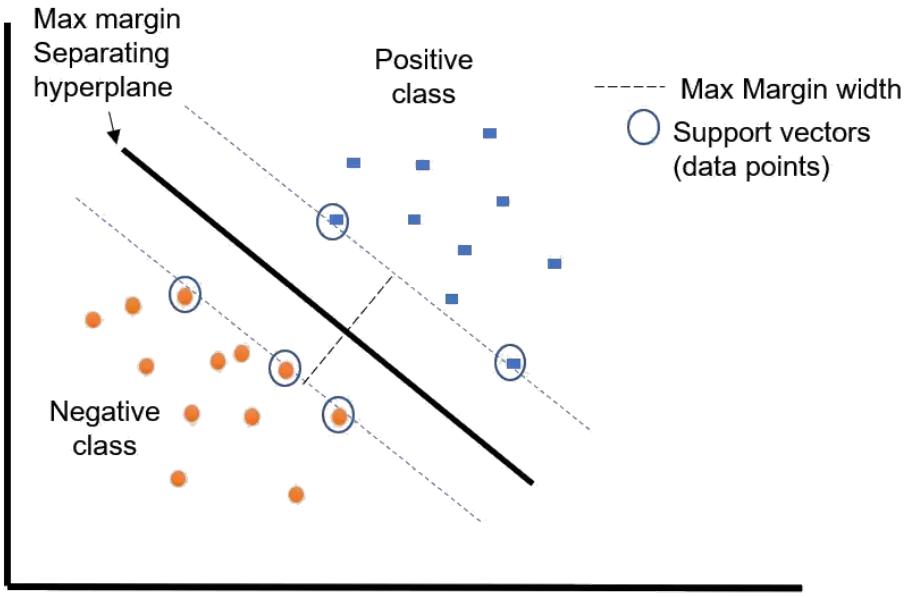


Figure 2.3: SVM model with optimal hyperplane and support vectors (data points at the same distance and closer to the hyperplane) highlighted. Adapted from: [8]

In this algorithm, the separation is achieved by identifying a hyperplane that maximizes the margin, i.e., the distance between the hyperplane and the nearest data points from each class. This plane acts as a decision boundary that can categorize new data points based on their position relative to the hyperplane.

Using a kernel function allows SVMs to efficiently handle complex nonlinear relationships between the input features and the target variable as they map these associations into separable

high-dimension spaces. Some examples of popular kernel functions include the linear, polynomial and radial basis function (RBF) kernels.

Overall, although SVMs can offer challenges with outlier data points and could overfit in higher-dimensional datasets under certain kernels, they are widely employed as efficient and robust algorithms [51, 19].

2.4.2 Unsupervised Learning

Trains models on an unlabeled dataset, where the algorithm finds patterns and relationships independently without predetermined categories or labels. Two common tasks in unsupervised learning are clustering and dimensionality reduction.

2.4.3 Deep Learning

A subset of the machine learning field that enables learning and decision-making through neural networks composed of multiple layers ("deep" in deep learning refers to the depth of the neural networks of this machine learning field). Deep learning models should be trained on large amounts of data to be effective, as their accuracy is improved over time while they continue to learn from more data. One advantage of deep learning is that it can be used for supervised and unsupervised learning tasks. Additionally, deep learning neural networks can be used for various tasks, such as computer vision, natural language processing, speech recognition, etc. [50]

2.4.4 Artificial Neural Network (ANN)

The ANN is a specific model utilized within the machine learning ecosystem. It draws inspiration from the structural and operational characteristics of biological neural networks, resembling the complex network of neurons connected through synapses in the human brain [21].

In constructing these neural networks, a group of interconnected nodes, called artificial neurons, are organized into layers. Each of these neurons receives input signals, applies weights and biases, and passes the result through an activation function to produce an output. The whole process of obtaining the artificial neuron output is depicted in Figure 2.4.

The architecture of ANNs is capable of learning from labelled examples and discovering patterns within data. Through the training process, which uses algorithms such as backpropagation, ANNs adapt their internal weights and biases to minimize the disparity between the predicted and real outputs. The iterative adjustment enables ANNs to generalize their learning and make accurate predictions on new data.

ANNs constitute a group of machine learning algorithms, such as the Multilayer perceptron (MLP) (Section 2.4.4.1) and the convolutional neural network (CNN) (Section 2.4.4.2).

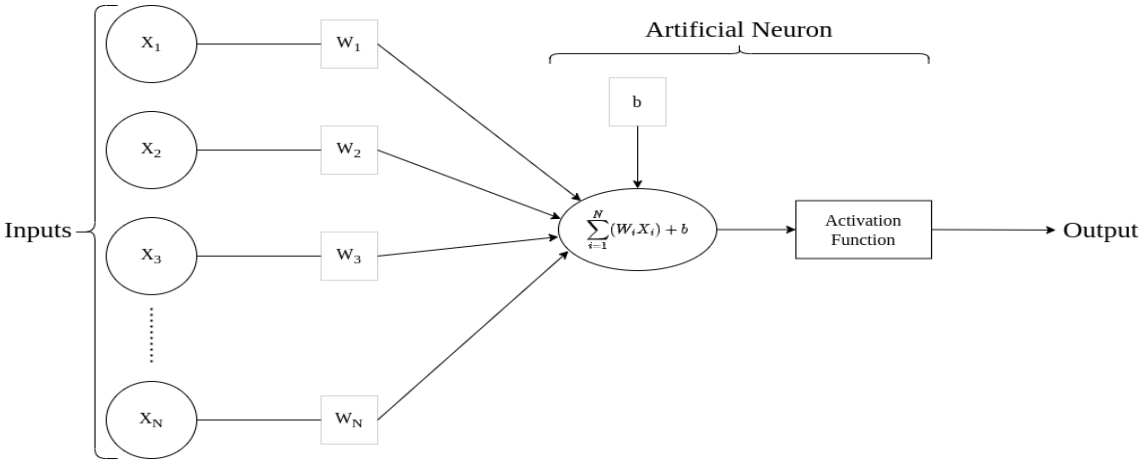


Figure 2.4: Artificial neuron structure. Adapted from: [21]

2.4.4.1 Multi-layer perceptron (MLP)

The MLP is an architecture within the domain of ANNs (represented in Figure 2.5). It is a feed-forward neural network (without loops between neuron connections) consisting of multiple layers of interconnected neurons, known as perceptrons. A perceptron is a single-layer neural network similar to what was represented in Figure 2.4, which is limited to classifying linearly separable patterns as referred in [21].

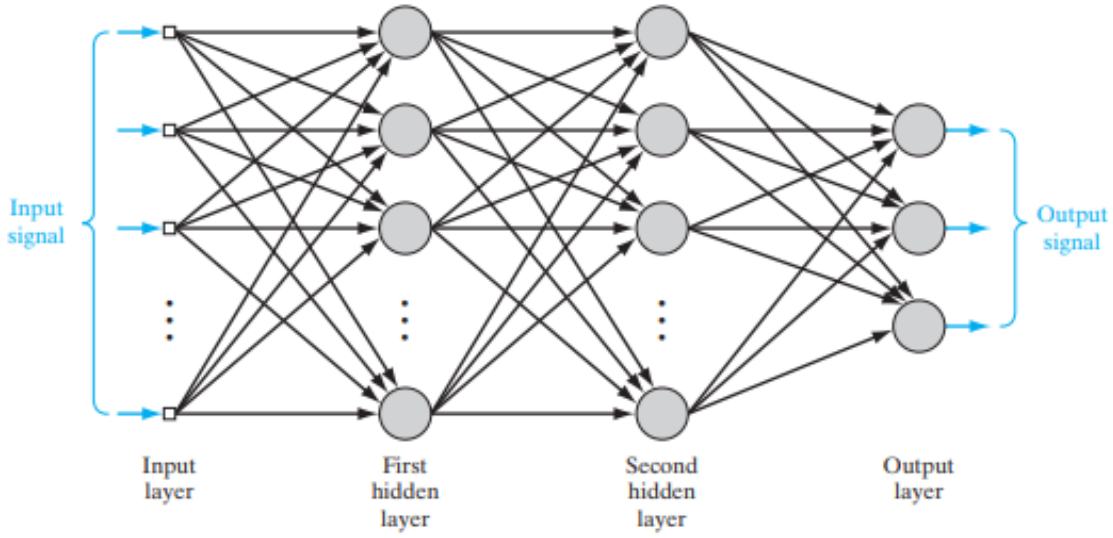


Figure 2.5: MLP architecture. Source: [21]

The MLP, in contrast, overcomes this limitation with the following three points that highlight its approach [21]:

- **Nonlinear differentiable activation functions** - Each neuron uses a nonlinear activation function. Popular choices include the sigmoid function, hyperbolic tangent function, rectified linear unit, etc.

- **Multiple hidden layers** - The network contains one or more hidden layers between the input and output layers.
- **High degree of connectivity** - Each layer's neuron (except the input layer) is connected to every one of the previous layer's neurons. This characteristic facilitates the flow of information and the propagation of signals through the network, allowing for extracting more abstract features from the input data.

A forward and backward phase exists in training these networks, commonly known as forward and back propagation [21].

In the forward phase, the input data is fed through the network, with the output of each neuron being produced by combining the weighted sum of inputs with the activation function. These outputs serve as inputs to the neurons of the subsequent layer, propagating the input signals through the network layer by layer until the output.

During the backpropagation phase, the network's output is compared to the desired one, and the difference quantified using a loss function, is used to compute the gradients of the network's parameters with respect to the loss. These gradients are then used to update the weights and biases of the network, iteratively refining its performance. These updates are successively done through the propagation of the gradients performed, layer by layer, in the backward direction.

2.4.4.2 Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) are a specific type of architecture found within the ANN model that has revolutionized various domains, particularly in computer vision, which employs tasks designated to extract and learn spatial features from imaging data. These types of networks offer effective solutions to find patterns in structured data. [26]

Unlike traditional neural networks, CNNs utilize convolutional layers that scan input data using small filters, also known as kernels. The network can extract local patterns and spatial relationships by convolving these kernels with the input, thus capturing features at different levels of abstraction. The convolution process is generally followed by pooling layers, which further aggregate and downsample the extracted features, reducing the dimensionality of the data.

Using the localized receptive regions and weight sharing allows CNNs to capture spatial correlations efficiently. By applying the same weights to kernels in different input regions, these networks can detect similar features in different image positions, reducing the number of learnable parameters compared to fully connected networks.

Similarly to MLPs CNNs use non-linear activation functions and train themselves with forward and backpropagation.

For image processing, training deep networks (deep learning) is essential. Nevertheless, as stated by Szeliski in [45], the most crucial component in these networks is trainable multi-layer convolutions. Since these can recognize patterns in the images, they can, ultimately, be used to classify them into different preconceived categories.

2.4.5 Algorithms evaluation metrics

The efficacy of machine learning algorithms hinges on their ability to accurately predict outcomes. This efficiency is quantified using a repertoire of evaluation metrics, each tailored to probe distinct facets of a model's performance. These metrics transform abstract notions of accuracy, precision, recall, and other performance dimensions into tangible figures. Some examples of evaluation metrics are the confusion matrix, the Receiver Operating Characteristic Area Under the Curve (ROC-AUC) and the F1-score.

2.4.5.1 Confusion Matrix

The confusion matrix is a tabular representation of the model's performance. It outlines true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions done by the model, offering a comprehensive view of the classification distribution [46].

- **TP** - Number of instances correctly predicted as positive.
- **FP** - Number of negative instances predicted as positive.
- **TN** - Number of instances correctly predicted as negative.
- **FN** - Number of positive instances predicted as negative.

2.4.5.2 Accuracy

Accuracy is an evaluation metric that measures the percentage of correct predictions on the total number of classifications. It gauges the overall correctness of the model's forecasts by estimating the ratio between the correct predictions and the total amount:

$$Accuracy = \frac{\#CorrectPredictions}{\#CorrectPredictions + \#WrongPredictions} \tag{2.5}$$

While the accuracy metric is valuable for evaluating model performance, it can introduce challenges, particularly when dealing with highly unbalanced datasets. In such scenarios, accuracy may yield misleading outcomes. Consider a situation where 99 out of 100 patients experience good outcomes. If a predictive model categorizes all patients as having favourable outcomes, the accuracy could be 99 %, suggesting exceptional performance. However, this seemingly positive result belies that the model might struggle to predict bad outcomes accurately. In nature, accuracy alone may not completely depict the model's effectiveness in capturing the patterns of different classes, highlighting the need for additional metrics that address these limitations.

2.4.5.3 F1-score

The F1-score is a metric that estimates the model's performance class-wise rather than an overall performance as done by accuracy. This is done by calculating the weighted average or harmonic

mean of the precision and recall measures [25]:

$$F1 = 2 \cdot \frac{Precision \times Recall}{Precision + Recall} \tag{2.6}$$

- **Precision -** Determines the ratio between correctly predicted positive cases to all the ones that were classified as positive (Equation 2.7).

$$Precision = \frac{TP}{TP + FP} \tag{2.7}$$

- **Recall -** Percentage of correctly classified positive cases (Equation 2.8).

$$Recall = \frac{TP}{TP + FN} \tag{2.8}$$

Even though the F1-Score mitigates the impact of disproportionate class sizes by considering false positives and negatives, it still has some sensitivity to class distribution since it does not treat all classes equally, offering a more skewed evaluation that accounts for the relative significance of different classes. It is a metric that can effectively capture the quality of the model's class-specific predictions.

2.4.5.4 ROC-AUC

The ROC-AUC quantifies the model's ability to discriminate between classes across different decision limits. For that effect, the ROC curve plots the True Positive Rate (TPR) as a function of the False Positive Rate (FPR), considering different classification thresholds.

- The **TPR**, like recall, measures the proportion of actual positive instances correctly predicted as positive by the model (Equation 2.8. It is also known as sensitivity.
- The **FPR**, also known as specificity, quantifies the ratio of negative samples wrongfully classified as positive by the model:

$$FPR = \frac{FP}{FP + TN} \tag{2.9}$$

Each point on a ROC chart corresponds to a different threshold for deciding on the classification score. The resulting curve is a tool that identifies a model's ability to distinguish negative instances from positive ones across different sensitivity and specificity values.

The AUC of the ROC curve (ROC-AUC) summarizes the overall performance across all possible thresholds. Its values range from 0 to 1, where a higher AUC corresponds to a better discriminatory power. An AUC of 0.5 suggests random guessing, while an AUC of 1 represents a perfect predictor [32].

Given its balanced consideration of TPR and FPR, the ROC-AUC metric is particularly well-suited for evaluating models in scenarios involving imbalanced datasets. This quality assures that

the metric accurately captures a model's ability to effectively discriminate between classes, even when one class vastly outnumbers the other in instances. Additionally, the ROC-AUC metric's representation of the model's performance across a spectrum of decision thresholds offers valuable insights into its behaviour under varying circumstances. Consequently, the ROC-AUC is a competent criterion to identify the most suitable model within this study's classification tasks. Thus, the selection of the optimal models in several studies is underpinned by the primacy of the ROC-AUC as a guiding metric.

Chapter 3

Literature Review

In this chapter, building upon the concepts presented in the preceding chapter (Chapter 2), the literature review is delivered, considering the theme of the dissertation. The aim is not only to showcase the viability of a machine learning-based prognosis solution for ICH patients but also to address pertinent research inquiries.

The structure of this chapter is as follows: Section 3.1 delineates the approach undertaken to conduct the state-of-the-art review, highlighting the research questions pursued, as well as the criteria and tools for selecting relevant studies. Subsequently, Section 3.2 delves into the investigated solutions, each tailored to address the research questions.

3.1 Approach

As noted in Chapter 1, the project's primary objective is to develop algorithms/models capable of prognosticating patients' outcomes utilizing data collected during the initial days of admission to the neurocritical care unit. This dataset encompasses both tabular and imaging modalities, necessitating the versatility of the proposed solution to accommodate multimodal data.

With the investigation presented in this chapter, the gathering of knowledge about different existing approaches is desired. For that purpose, having into consideration the objective of implementing the described prognostication tool, the articles have been incorporated into the state-of-the-art study based on explicit measures. Specifically, articles were included if they satisfied the following prerequisites:

1. They featured studies involving human subjects;
2. The articles included a prognostic tool designed to predict patients' outcomes;
3. The scope of the studies included either tabular patient data, imaging data, or both.

Notably, studies centered around patients with ICH were prioritised in the state-of-the-art study's inclusion process.

The selection process involved the application of these inclusion criteria to the abstracts during the initial review. Abstracts aligned with these underwent a more comprehensive and in-depth analysis for their subsequent presentation within this section.

Through the literature review, the aim was to uncover insights addressing the four following research questions:

- What are approaches for prognostic scoring systems and scales commonly used for assessing outcomes in patients with ICH?
- What successes have been reported in implementing machine learning-based predictive models for patient functional outcomes?
- What are the latest AI-driven solutions strategies for predicting patient functional outcomes in neurological medical conditions?
- How are predictive models combining data from medical imaging and clinical records to provide accurate predictions of patient functional outcomes?

With these in mind, to look at the scientific work done on the subject, the following search engines were utilized:

- Google Scholar [1]
- Elsevier [2]
- B-On (Biblioteca de Conhecimento Online) [3]
- ResearchGate [4]

Additionally, a Zotero [5] repository has been provided, containing a compilation of articles centered around investigative endeavours within the domain of neurocritical pathologies. Prior researchers and investigators have utilized this repository in their academic pursuits.

3.2 Related work

During the literature review, multiple projects in the medical realm have been found, each harnessing the potential of machine learning to predict patient outcomes. Upon examination of these initiatives, a set of recurrent methodologies was found.

The most common among these methodologies is the approach of developing models through deep learning (Section 2.4.3) and supervised learning (Section 2.4.1) techniques. These models

[1] https://scholar.google.com/
[2] https://www.sciencedirect.com/
[3] https://www.b-on.pt/
[4] https://www.researchgate.net/
[5] https://www.zotero.org/

are then applied to various datasets containing clinical records, medical images, or both to anticipate outcomes linked closely to a targeted variable encapsulating the patients' functional statuses. Within the solutions, the knowledge discovery process generally passes through the phases represented in Section 2.2.

To elucidate these methodologies, the subsequent three subsections expound upon the pertinent solutions. These subdivisions distinguish from each other in terms of the data modalities utilized in training distinct models for forecasting patients' outcomes, thus showcasing the varied strategies utilized for their development. The initial subsection outlines projects that produce models reliant solely on patient non-imaging data (e.g., demographic, clinical, laboratory data). The second subsection encapsulates models constructed from imaging modalities, such as CT scans and MRIs. Lastly, the third and final one presents models implemented by amalgamating patterns retrieved from imaging and tabular data through fusion techniques to develop a multimodal solution.

3.2.1 Tabular Data-Based Studies for Outcome Prediction

Regarding approaches using numerical and categorical patient data, several noteworthy studies have been conducted in stroke outcome prediction, employing machine learning methodologies to anticipate patient prognoses. For instance, Lin et al. [29] embarked on a comprehensive investigation into the viability of machine learning techniques for stroke outcome prediction. By leveraging a dataset comprising 58493 data instances from the Taiwan Nationwide Stroke Registry, which included both ischemic and hemorrhagic patients, they aimed to forecast 90-day stroke outcomes.

Key to their approach was the utilization of supervised learning algorithms such as SVM, RF, ANN and a hybrid artificial neural network (HANN). Employing an evaluation strategy involving 10-fold cross-validation, the models were systematically trained and assessed. Before modelling, critical data pre-processing steps were undertaken, including:

- Binarization of the outcome variable based on mRS cutoffs, distinguishing good (mRS ≤ 2) and bad (mRS > 2) outcomes.
- Feature selection utilizing the extra-trees algorithm to reduce the possibility of overfitting and emphasise informative attributes.

Overall, this solution showed robust results for ischemic and hemorrhagic stroke patients. Notably, the model achieved an AUC of 94% when incorporating preadmission and inpatient data. The introduction of follow-up data further improved the model's accuracy, elevating the AUC to 97%. These findings emphasise the utility of machine learning methodologies for prognosticating stroke patient outcomes.

More focused on the domain of ICH, another significant study by Wang et al. [49] was dedicated to developing a predictive framework for verifying the functional statuses of ICH patients at the 1st and 6th months post-event. Relying on a dataset spanning demographic traits, laboratory findings, and imaging results of 333 ICH patients, the study adopted a similar approach to Lin et al., utilizing supervised machine learning models grounded in the same binary mRS-based label

(good outcome: $mRS \leq 2$; bad outcome: $mRS > 2$). Noteworthy in their pre-processing was the application of the synthetic minority oversampling technique (SMOTE) to rectify class imbalances and the presence of features obtained from CT imaging on their data. The outcome was a model that exhibited substantial efficacy. The random forest model, in particular, demonstrated the best performance, achieving an AUC of 89.9% for the 1st-month prediction and further elevating to 91.7% for the 6th-month prediction.

In an alternate approach, Matsuo et al. [30] conducted a study to predict in-hospital mortality or poor outcomes in patients with traumatic brain injuries (TBIs). Their investigation utilised a dataset comprising 232 individuals. To validate their models, they used a combination of a train-test dataset split along with a 5-fold cross-validation strategy. This methodology allowed them to assess the models' generalizability and test their performance on unseen data.

Across their analyses, the RF model emerged as the top-performing model for poor outcome prediction, yielding an AUC of 89.5%. In contrast, the Ridge Classifier demonstrated the highest performance for predicting mortality, with an AUC of 87.5%. During the cross-validation phase, the SVM with a Radial Basis Function (RBF) kernel exhibited superior performance for poor outcome prediction, achieving an AUC of 89.4%. Also, for mortality prediction, the RF model excelled during cross-validation, attaining an AUC of 96%.

Overviewing, the tabular data-based models presented in this section exemplify effective strategies for addressing the core problem of this thesis. These techniques build models that accentuate the potential inherent in harnessing tabular data containing essential information, including patient demographics, clinical profiles, and laboratory specifics. By applying supervised learning algorithms to these structured attributes, these models stand as compelling demonstrations of the capacity to achieve efficient prognostications concerning outcomes related to stroke and ICH.

3.2.2 Imaging Data-Based Studies for Outcome Prediction

This subsection focuses on projects that harness imaging data to construct predictive models for patient outcomes. The multidimensional nature of medical imaging data offers insights into the patient's condition's structural and functional aspects. These studies utilize a range of deep learning architectures to extract pertinent features from imaging data and correlate them with clinical outcomes. While not exclusively focused on ICH, these investigations exemplify the potential for prognosticating patient outcomes across different medical conditions using images.

The application of deep learning neural networks, namely through 3D CNNs, has emerged as a prevalent technique for predicting patient prognoses based on CT scans. This trend is increasingly evident across diverse medical domains. [12, 37, 42, 35].

In the specific context of stroke prognosis, an advanced approach was pursued by Pérez del Barrio et al. [37]. Their study involved developing both tabular and image-only models, which were subsequently integrated into a comprehensive multimodal solution for prognosticating patients with intracranial hemorrhages. Concentrating on the image-only model, the researchers extracted a dataset of 262 distinct DICOM volumes derived from sequential head CT scans. These volumes encompass image dimensions of 512x512 pixels.

To enhance the efficacy of model training, several pre-processing steps were implemented. These involved rescaling pixel values to Hounsfield units to a range between 15 and 100, and downsampling images to a size of 128x128 pixels. Subsequently, a 3D CNN was used to extract features from the volumetric data.

In terms of outcomes, the image-only model demonstrated a somewhat notable performance. It achieved an AUC of 76.3%, affirming its capability to predict patient prognoses using solely imaging data.

Another study, contrary to the previous one, exclusively utilized imaging data to predict the severity of neurological outcomes was conducted by Zeng et al. [52]. This project leveraged, again, a deep learning model trained with a 3D-CNN to assess the impairment of patients who had experienced ischemic strokes, both upon admission and on the seventh day of hospitalization. The medical imaging data, in this case, was derived from MRIs. During the pre-processing pipeline, the DICOM images were transformed in the NIfTI format to facilitate data handling, and various image sizes (256x256x64 and 128x128x32 voxels) were used. Overall, the model showed superior results when using a voxel size of 256x256x64 upon admission (AUC of 84.6%) and a voxel size of 128x128x32 on the seventh day of hospitalization (AUC of 90.5%).

Collectively, these studies underscore the potential of advanced image analysis techniques for enhancing stroke prognostication methodologies, offering valuable insights into the feasibility of image-based models for the prognostication tool to be implemented in this thesis.

3.2.3 Multimodal Data-Based Studies for Outcome Prediction

This subsection explores models that amalgamate imaging and tabular information to develop predictive frameworks for patient outcomes. Researchers combine these distinct data modalities to enhance predictive accuracy and robustness. These projects provide insights into the potential for complementary benefits between clinical and imaging data.

In a study conducted by Nawabi et al. [34], the feasibility of employing machine learning models trained on non-enhanced CT scans of patients with ICH was explored to predict their functional outcomes. This research has the particularity of integrating these image-based predictions with the established ICH score prognostication model. This integration aimed to provide a more comprehensive forecast of outcomes categorized by different mRS cutoffs.

First, the image-based model was assembled following the workflow depicted in Fig. 3.1.

Then, the probability of death estimated by the ICH score is combined with the probability of that outcome given by the image-based model to arrive at a final decision. The fusion is done by averaging these probabilities.

This solution gave the outcomes for the different cutoffs of mRS for the imaging-only model. For its integration with the ICH score, only the survivability prediction was evaluated, as the ICH score only gives its probability. The results are presented in Table 3.1.

For the study conducted by Pérez del Barrio et al. [37], already approached in the previous section, the fusion of the tabular model with the imaging model was done by passing the output of both the last convolutional block of the image fed CNN and the output of a feed-forward network

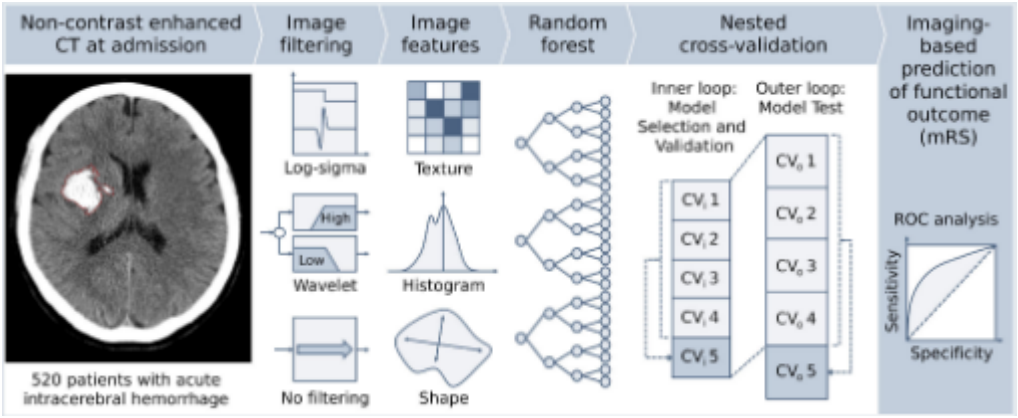


Figure 3.1: Nawabi et al. imaging-based model pipeline. Source: [34]

mRS cutoff	Image-based model	Imaging + ICH score model
2	80%	-
3	80%	-
4	79%	-
5	80%	84%

Table 3.1: Nawabi et al. models ROC-AUCs. Adapted from: [34]

inputted with tabular data to a last dense layer with a sigmoid and activation function. This resulted in an AUC of 92.4 % in the hybrid model, which is a jump from the results from the image and tabular models, which had a respective AUC of 76.3% and 74.6%.

In the study conducted by Pérez del Barrio et al. [37], previously discussed in the preceding section, the integration of the tabular model with the imaging model was realized through a fusion strategy. The output from the last convolutional block of the image-driven CNN and the output from a feed-forward network, which was supplied with tabular data, were combined. This amalgamation culminated in a final dense layer with a sigmoid activation function. Consequently, the hybrid model achieved an AUC value of 92.4%. This represents a significant improvement compared to the individual outcomes of the image-based and tabular models, which yielded AUC values of 76.3% and 74.6%, respectively.

Another intriguing fusion approach is the one depicted in the project of Sadasivuni et al. [40], who united patient electronic medical records and physiological sensor data to predict the early risk of sepsis. In this case, a meta-classifier was trained on the prediction scores of two individual models to integrate information from two sources. This late fusion strategy proved remarkably effective, yielding an accuracy of 93% and an AUC of 99% using a linear SVM classifier.

In summary, these studies showcase diverse fusion strategies that harness the power of integrating imaging and tabular data to propel predictive capabilities. This multimodal approach not only enhances the precision of prognostication but exemplifies the synergy between clinical and imaging data sources.

This fusion-based approach underscores the potential synergy between distinct data modalities for improving predictive accuracy in the context of intracranial hemorrhage prognostication.

Chapter 4

Solution & Methodology

In this chapter, the first section (Section 4.1) describes the data used to accomplish the proposed solution for the problem stated in Section 1.2. This solution is introduced in the subsequent section (Section 4.2), in which the technologies employed to achieve it are also described. To conclude, the final section (Section 4.3) delineates the methods used to achieve the proposed solution.

4.1 Data Description

This study analyses a retrospective dataset composed of records of patients who suffered an ICH and were admitted into the Neurocritical Care Unit of the S. João Hospital Center between January 16, 2015, and November 28, 2021. A critical consideration was the protection of patient privacy and confidentiality, achieved through an anonymization process that ensured the removal of sensitive personal information.

The dataset contains a rich array of information, including numerical, categorical, and imaging-related data, collected from 65 individuals who experienced ICH.

4.1.1 Tabular data

The numeric and categorical side of the database is structured in a tabular format and was made available through an Excel (.xlsx) file. The table features 111 variables, collectively representing clinical, demographic, MCM, medical histories and follow-up aspects of the patients' course of the medical condition.

During the variable selection process, certain measures were established to determine the suitability of each variable. Given that the focus of this study revolves around predictions made within a one-week timeframe, variables that extended beyond this temporal range were excluded from the training dataset (except the target variable). For instance, variables such as the mRS scores obtained at ICU discharge were intentionally omitted. While these scores provide valuable information about patients' post-ICU functional status, they are typically acquired over a longer duration. As such, they may not capture the immediate effects pertinent to the study's predictive objectives. Overall, seven variables were removed, resulting in 104 selected features.

Concerning patient exclusions, the ones with a high threshold of NULLs (> 60%) on the 104 selected features were removed. This resulted in drawing only one patient from the tabular dataset, ending up with a sample size of 64 patients.

Exploring some of the demographic features of the individuals, it was revealed that the ages of the patients spanned a range from 29 to 86 years old, following the distribution represented in Figure 4.1.

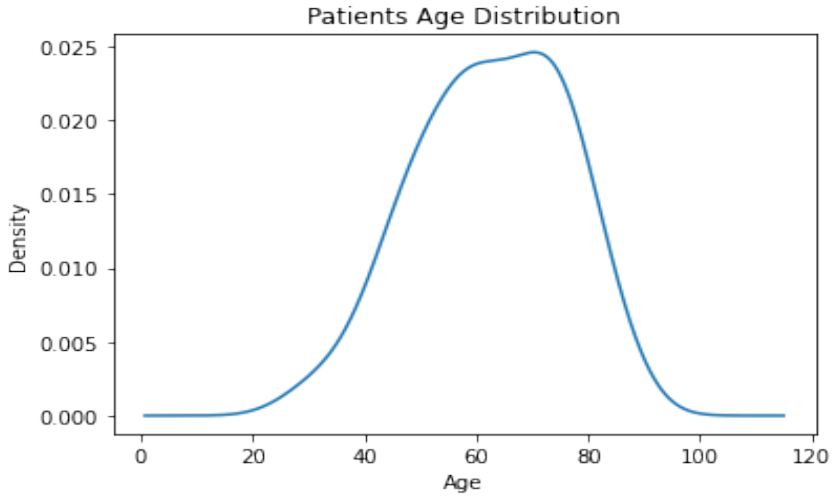


Figure 4.1: Patients age distribution

Regarding gender distribution, the dataset indicated that 24 patients were identified as female and 41 as male.

As stated in section 1.1, the ICM+ tool is highly employed for neurocritical patients. That is the case for the ICH patients who, from admission until discharge, were monitored resorting to ICM+. The monitoring data regarding each patient is organized into the following file types:

- **Raw files** containing different signal values monitored over time
- **Summary files** holding the offline estimations obtained from the monitored signals present in the raw files

Through these files, the tabular MCM features were retrieved by using the ICM+ software offline estimation capabilities. These consisted of signal aggregations at the first 24 and 48 hours after the admissions into the ICU (MCM variables - Appendix A).

About the remaining dataset, it is imperative to underscore the inclusion of variables encompassing the mRS (Section 2.1.4) assessments before ICU admission, at the point of ICU discharge, and six months after ICU admission. Additionally, notable variables include the ICH (Section 2.1.2) scores, the SAPS II (Section 2.1.3) measurements, and the GCS (Section 2.1.1) values, all of which exhibit direct correlations with patient outcomes.

The characteristics of some of the dataset's demographical, clinical and follow-up variables are presented in Table 4.1.

Variable	Value
Number of patients	64
Number of Males	41 (64.1 %)
Number of Females	23 (35.9 %)
Age	62.2 ± 13.1
ICH score	2.1 ± 0.9
GCS	10.1 ± 3.3
SAPS II Score	43 ± 12
Patients with mRS = 0 after 6 months	0 (0 %)
Patients with mRS = 1 after 6 months	1 (1.6 %)
Patients with mRS = 2 after 6 months	2 (3.1 %)
Patients with mRS = 3 after 6 months	13 (20.3 %)
Patients with mRS = 4 after 6 months	20 (31.3 %)
Patients with mRS = 5 after 6 months	6 (9.4 %)
Patients with mRS = 6 after 6 months	22 (34.4 %)

Table 4.1: Characteristics of a subset of demographical, clinical and follow-up features present in the 64 patients database

4.1.2 Imaging Data

The tabular database contains seven variables extracted by medical professionals from the first brain CT scans of the patients after ICU admission. These variables are considered with the other tabular features (a subset of the 104 variables) and are described in Table 4.2.

Variable	Description
ICHLocationSupraInfra	ICH location (**0** = Supratentorial; **1** = Infratentorial)
ICHside	ICH side (**0** = Right; **1** = Left)
Location_Origin	Location origin of the ICH (**0** = Lobar; **1** = Lenticulocapsular; **2** = Talamocapsular; **3** = PurelyVentricular; **4** = Cerebelar; **5** = Brain-Stem)
IVH	Intraventricular Hemorrhage (**0** = No; **1** = Yes)
VolA	Greater anteroposterior diameter
VolB	Greater mediolateral diameter
VolC	Greater superiorinferior diameter
ICHVolume	Intraparenquima ICH volume
IVHVolume	Intraventricular ICH volume

Table 4.2: Tabular features extracted from axial brain CT scans

The imaging side of the database consists of a total of 3576 images in DICOM format taken, as noted previously, from the first axial brain CT scans (e.g., Fig. 4.2) after ICU admission:

- Each patient has multiple DICOM images representing, each with a 512x512 resolution, an axial slice of the brain scan.
- Each patient has an average of 55 slices.

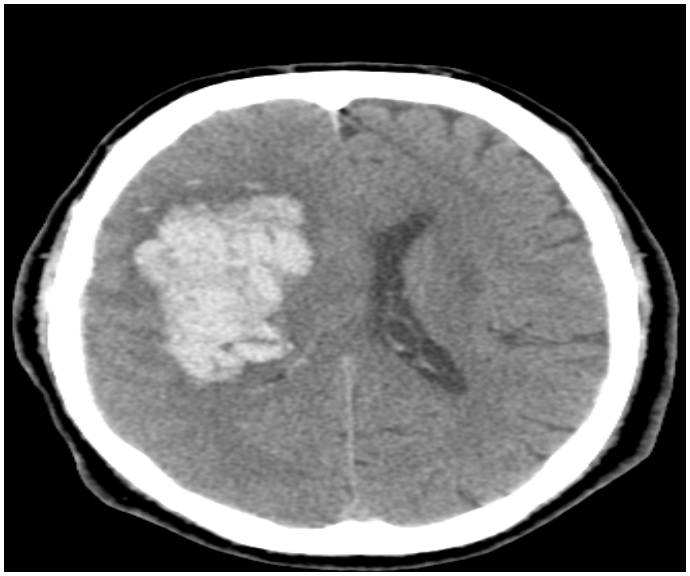


Figure 4.2: CT scan slice

4.2 Proposed Solution

As stated in section 1.2, the problem asks for the use of advanced machine learning algorithms with predictive capabilities for detecting long-term consequences in ICH patients, and, hence, be a tool for the prognosis, detection and prevention of outcomes in these patients.

With this in mind, as the training data is labelled (6th-month mRS), the solution will use the supervised learning paradigm to create various models that output predictions on the mRS score at the end of the 6th-month, similar to some examples represented in Chapter 3.

Overall, multiple models are created for:

- Predicting if the patients have good or bad outcomes, which are classified as such, by binarizing the mRS score (thus, reducing the output complexity in the final model).
- Forecasting the outcomes of the patients with multilabel predictions on the mRS score variable.

Using only the tabular data to achieve the predictive models, the development will pursue the four following steps:

- The dataset will first be explored and studied through statistical analysis and then be prepared.
- Then, the predictor models will be trained by implementing and experimenting with different algorithmic approaches and preprocessing pipelines to predict the patient's functional status.
- While experimenting with the different algorithms, the parameters will be tuned, and the generated models will be validated using techniques such as cross-validation.
- To end, after applying the models, a feature importance study will be carried out to unveil associations between some parameters and the functional outcome of patients.

For the analysis of the CT scans, the implementation will construct tri-dimensional convolutional networks for classifying the images accordingly, using auxiliary frameworks.

The final predictive model (Figure 4.3) will result from combining the image-only and tabular data models using a late fusion method that trains a meta-classifier through the outputs of the models. A prediction of the patients' state can then be acquired by fitting these outputs to the final model.

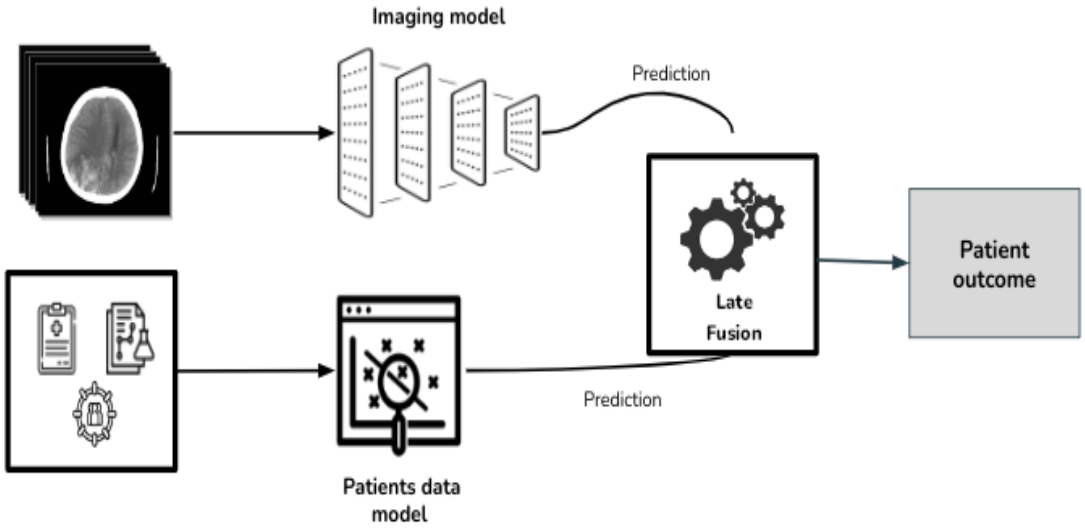


Figure 4.3: Proposed multimodal predictive model

4.2.1 Technologies

The programming language employed throughout all stages of the developmental continuum for the models was Python [47]. To expedite the implementation process, a selection of libraries were harnessed, amongst which the subsequent entities played a critical role:

- **Pandas and NumPy** - Python libraries used to analyze and manipulate data.

  - These libraries are operated mainly in the project's data preparation (e.g., high NULL threshold patient removal) phase.
  - Pandas provides tools that facilitate accessing and manipulating tabular data [31].
  - NumPy allows array programming with a "powerful, compact and expressive syntax for accessing, manipulating and operating on data in vectors" [20].

- **Pycaret** - Open-source library that automates the machine learning workflows. It is effectively a Python wrapper around other machine-learning libraries and frameworks [3].
  - Using this tool, certain pre-processing tasks can be automated, and specific algorithms can be trained and tuned to create predictive models.

- **Keras** - Deep Learning API running on top of TensorFlow that allows for the building and training models (such as CNNs) [14].

- **NiBabel** - Library for accessing neuroimaging file formats, image data and metadata [10].

4.3 Methodology

This section describes the methodology followed to produce the proposed solution. Since the solution revolves around developing two major models (tabular and imaging-based) and merging them into one multimodal model, these will be explained in three subsections: Tabular-Data Models (Section 4.3.2); Imaging-Data Models (Section 4.3.3) and Multimodal Models (Section 4.3.4). These subsections are preceded by a section that details the models' common aspects of their development across the different data modalities.

4.3.1 Common Methodological Aspects

4.3.1.1 Target Variable Balance and Developed Models

As outlined in the solution proposal section, the models are constructed with a specific focus on the mRS variable, measured six months after the ICU admission of patients who had experienced ICH. The proposed approach involves the development of both multilabel and binary classification models. To ensure the integrity of the modelling process and to avoid potential biases, an examination of data balance was undertaken. This analysis aimed to identify appropriate data binarization and partitioning strategies that would facilitate the creation of models capable of treating all classes impartially, including highly prevalent ones. A comprehensive visualization of the distribution of mRS values is provided in Figure 4.4.

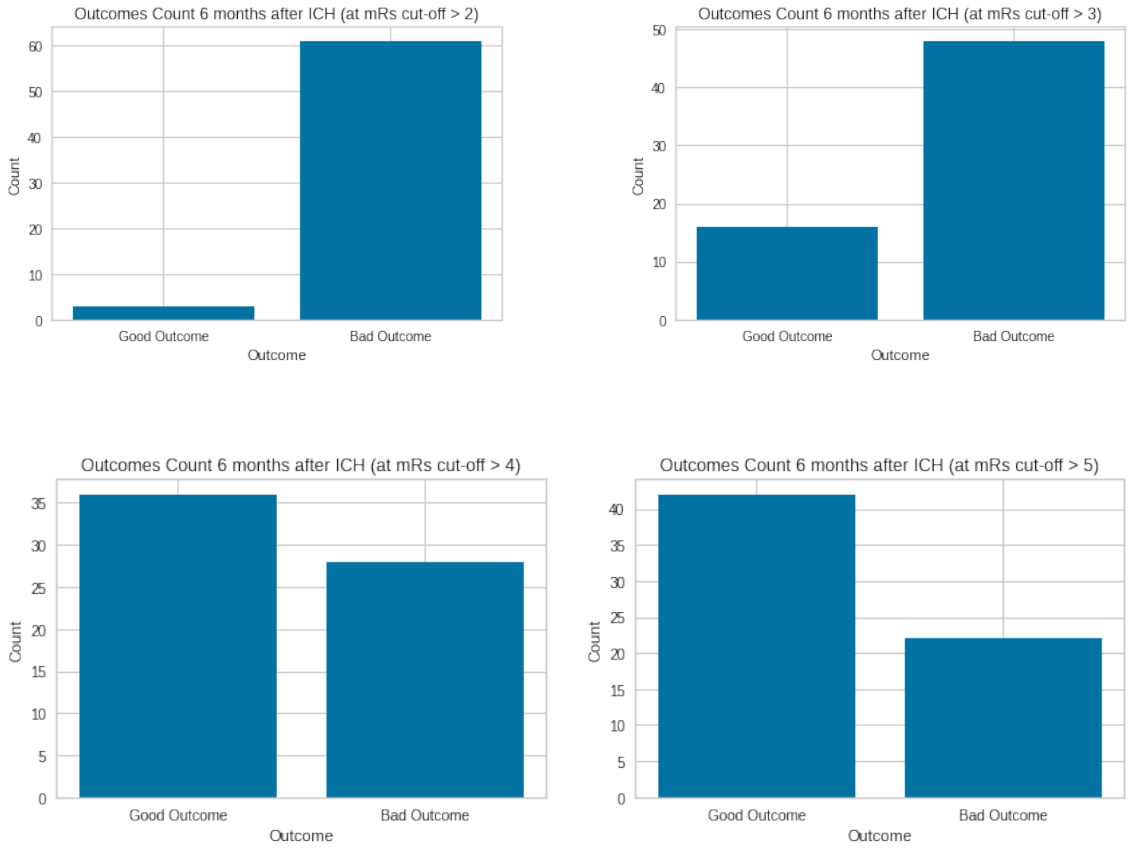


Figure 4.4: Class balance of the mRS variable at different cutoffs

In pursuing effective model construction, various threshold values were considered to binarize patient outcomes into favourable and unfavourable categories. Specifically, the threshold values of mRS = 2, mRS = 3, mRS = 4, and mRS = 5 were evaluated. However, it was observed that the distribution for the mRS = 2 threshold was exceedingly imbalanced, with only three instances out of a total of 64 falling into this category. Given the limitations posed by such a severe imbalance, the decision was made not to implement models based on the mRS = 2 threshold.

For the multilabel model, the optimal strategy involved aggregating classes to facilitate multi-label classification. This aggregation was performed as follows:

- **Class 0**: This class was defined by patients with mRS values encompassing 0, 1, 2, and 3, resulting in 16 patients.
- **Class 1**: Patients with mRS values of 4 and 5 were grouped into this class, amounting to 26 patients.
- **Class 2**: Patients with an mRS value of 6 formed this class, comprising 22 patients.

This approach to aggregate classes was chosen to address the imbalanced nature of the mRS variable while maintaining a meaningful grouping that aligns with the clinical understanding of patient outcomes. By consolidating the mRS values in this manner, the multilabel model could

effectively capture the nuances of patient recovery courses while enhancing the model's capability to generalize across different levels of functional outcomes.

4.3.1.2 Train Test Split

To train the described models, the datasets were partitioned into the training and test sets. The training set was assigned the dual role of serving as an assessment and validation tool for evaluating algorithm performance and being employed to train the classification models. On the other hand, the test set was exclusively reserved to evaluate whether the generated models exhibit signs of overfitting, which can be proven by performances that aren't on par with the training results. This approach facilitates the evaluation of the model's capacity to generalise to unseen examples distinct from those encountered during the training phase.

Weighing the limited sample size of the dataset (comprising 65 patients for imaging models and 64 for tabular models), several distinct train-test splits were considered. This was done to investigate the potential advantages different training set sizes could offer in enhancing algorithm performance. In the end, a division of 75/25 % was done, where 75 % of the dataset was employed for training and the rest for testing.

4.3.1.3 Evaluation of Model Performance

This study's complete evaluation of model performance contains various classification metrics chosen to provide a multifaceted understanding of the models' efficacy. Three metrics were considered for all the models: the ROC-AUC, accuracy and confusion matrix metrics. The selection of the ROC-AUC and accuracy was done because they are used in most of the studies presented in Section 3.2. The confusion matrix was used to verify if any biases were present for any class.

This study's complete evaluation of model performance contains various classification metrics chosen to provide a multifaceted understanding of the models' efficacy. Three selected classification metrics converge to offer a full understanding of these algorithms' predictive power. Among the trio, the ROC-AUC, accuracy, and confusion matrix metrics were chosen to provide insights into distinct aspects of model performance. The selection of ROC-AUC and accuracy metrics finds its rationale in their presence across a range of studies expounded in Section 3.2. The strategic integration of the confusion matrix serves to examine potential biases within class prediction distribution.

In the pursuit of multiclass ROC-AUC estimation, this study navigates an array of methodologies to uncover the most fitting approach for our predictive models:

Multiclass Metric Estimation Methodologies

In the context of multiclass classification tasks, the assessment of performance metrics such as the F1-score and the ROC-AUC metric can be conducted for individual classes within the target label, following a one-vs-rest approach. To facilitate more straightforward comparisons and to derive

overarching performance measures, the predictions can be subjected to the following averaging methodologies:

- **Micro -** Computes positives and negatives globally, i.e., all positives and negatives across every class are considered for the final metric calculation. Notably, this approach does not involve measuring one-vs-rest values.
- **Macro -** Here, the average of the chosen metric scores for each distinct class is estimated. This method treats each class equally, irrespective of its prevalence, making it particularly suitable for balanced datasets or for giving equal significance to all classes
- **Weighted -** This approach shares similarities to macro averaging, except each class's contribution is weighted by its size in the dataset.

For the current project, the macro averaging methodology was chosen for the ROC-AUC calculations. This decision is grounded in the equal importance of predicting all the possible outcomes of the patient to allocate resources for their treatment. Although there are minor distribution imbalances, they weren't considered significant enough to require other averaging methods.

4.3.2 Tabular-Data Models

During the tabular-data model development, our approach was anchored to the framework sketched in Figure 2.1. This framework dictated a systematic progression open to retrogressive iterations within the procedural schema. This iterative feature facilitated the implementation of trial and error methodologies aimed at the identification of efficient models for prognosticating the functional statuses of patients across distinct divisions of the modified Rankin Scale (mRS) variable (Section 4.3.1.1).

Implementing the models followed an iterative methodology that first trained a foundational model using a base data preparation pipeline. This initial model served as a reference point against which various alternative data pre-processing pipelines were contrasted using more intricate methodologies. Subsequently, the determined techniques that proved to be beneficial were amalgamated to form an ultimate pre-processing pipeline. This pipeline was utilised in training more effective models, which are subsequently extracted for integration within the multimodal solution.

4.3.2.1 Data pre-processing

To explore and leverage diverse data pre-processing strategies, a range of normalization and imputation techniques were systematically employed. The intention was to establish the most suitable methods to improve model performance. These methodologies were discerned by applying various pre-processing techniques available within the Pycaret Python library, which provides a suite of versatile data preparation capabilities.

Specifically, the exploration surrounded the following domains of data preparation, each contributing to the refinement of the model's predictive potential:

Data Imputation

Within the array of pre-processing pipelines, the application of data imputation techniques is always present, intending to address missing values within the dataset.

This process aids in the integrity and completeness of the data, ensuring that subsequent analyses are founded on reliable information.

The techniques of this domain were used both in numerical and categorical features. The employed methods made available by the Pycaret library were the following.

- **Numerical data:**
  - **Mean Imputation** - Missing values replaced by column average.
  - **Iterative Imputation** - Uses regression to impute numerical values. In this study, the default classifier (LightGBM[1]) provided by Pycaret, is utilized.
- **Categorical data:**
  - **Arbitrary Imputation** - Missing categorical values are replaced arbitrarily. In this study, these were substituted by -1.

Data Normalization

Data normalization is a critical facet of data pre-processing used to homogenize the scales and distributions of the various features within a dataset. This study applies four normalization techniques (to both numerical and categorically encoded data) inherent in the Pycaret library to mitigate the biases stemming from varying scales. This facilitates the model's capacity to extract meaningful patterns from the data. The methodologies used were:

- **Min-Max Normalization** - Transforms the values into a specific range between 0 and 1. This technique is appropriate when preserving the relationships between data values is important. It can be particularly effective for algorithms that rely on distance comparisons. The formula of this method is given by:

$$x_{normalized} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{4.1}$$

- **Max-Abs Normalization** - This method scales the data such that the absolute maximum value is 1. This technique is useful when preserving characteristics like the sparsity of the data or when negative values hold significance. It is similar to Min-Max normalization, the only difference being that the sign of the values is maintained. To do the normalization, the following calculation needs to be done:

$$x_{normalized} = \frac{x}{|x_{max}|} \tag{4.2}$$

[1] https://lightgbm.readthedocs.io/en/latest/pythonapi/lightgbm.LGBMClassifier.html

- **Robust Normalization** - Designed to mitigate outliers' influence, robust normalization uses the interquartile range, i.e., the difference between the 75th and 25th percentiles of the dataset. To calculate the normalized value, the following equation is used:

$$x_{normalized} = \frac{x - med}{IQR} \tag{4.3}$$

where

IQR = Interquartile range
med = Column median

- **Z-score Normalization** - Also known as standardization, this technique converts the data to have a mean of zero and a standard deviation of one. This technique changes the data distribution to have a central mean point and ensures that the data values are expressed in terms of their standard deviations from the mean. The formula for z-score normalization is given by:

$$x_{normalized} = \frac{x - \mu}{\sigma} \tag{4.4}$$

where:

x = Data point value
μ = Mean of the column that is being normalized
σ = Standard deviation of the column that is being normalized

Categorical Encoding

Categorical variables were strategically transformed into numerical representations to enhance categorical variables' compatibility with the models' numerical supports. This elevates the capacity of the model to learn from the data.

The strategy employed for this effect was the Leave-One-Out (LOO) Encoding. This technique is particularly useful when dealing with high-cardinality categorical features with numerous unique categories. This method helps mitigate the risk of introducing many new dimensions to the data, which could lead to issues like the curse of dimensionality. For instance, using one-hot encoding, which generates one binary variable per category, might result in many dimensions or sparse representations. LOO encoding provides a more compact representation while capturing the relationship between the category and the target.

The LOO process involves encoding a categorical value by calculating the mean of the target variable for all instances belonging to that category, excluding the instance being encoded, i.e., for each category, the technique calculates the mean target value across all samples except the current one. This mean is then assigned as the numerical representation of the categorical value in question.

This approach aimed to retain valuable information from categorical features while mitigating the challenges associated with introducing numerous dimensions in the modelling process.

Oversampling

Recognizing the significance of a balanced class representation, the SMOTE oversampling technique was embraced, as some of the models discussed in the literature review chapter also do. This technique addresses class imbalances by artificially increasing the representation of minority classes.

This methodology might prove useful in ensuring that dominant class influences do not sway the model.

Pipelines

As already stated, different pre-processing pipelines were implemented. The following table depicts the employed methods:

Pipeline	Imputation	Normalization	Oversampling
P0	Mean	None	None
P1	Iterative	None	None
P2	Mean	Z-score	None
P3	Mean	Min-Max	None
P4	Mean	Max-Abs	None
P5	Mean	Robust	None
P6	Iterative	Z-score	None
P7	Mean	None	SMOTE
P8	Mean	Z-score	SMOTE
P9	Iterative	Z-score	SMOTE

Table 4.3: Pre-processing pipelines

The LOO encoding is done on the non-binary categorical variables within each of these pipelines.

4.3.2.2 Algorithms

In the process of constructing a robust prognostic model based on tabular data, the selection of appropriate algorithms is a crucial step. Drawing inspiration from the predictive modelling solutions presented in Section 3.2, this study embraces a strategy by electing algorithms that have demonstrated commendable performance using only tabular data. The foundation for these selections lies in the literature review that examined various algorithms' efficacy and the prevalence of the algorithms in the different studies. As such, the selected algorithms were the RF, GBC, SVM (with RBF kernel) and MLP. The best RF model was also employed to view the feature importance of the variables used to train them.

4.3.2.3 Models selection

In the pursuit of identifying the optimal predictive algorithms, a model selection process was executed, underpinned by a 10-fold stratified validation technique applied to the training dataset. The delineated methodology adheres to a sequence of steps, ensuring a thorough evaluation of the models' predictive capacities [6]:

1. Separates the training dataset into ten different subsets (folds) with the same class proportions (stratified)
2. Select the last fold as the test dataset and the rest as the training dataset
3. Train the model with the training dataset and validate on the test fold
4. Save validation metrics
5. Repeat steps 2-4, swapping the fold used for validation to the previous one. When all the folds have been used for validation, pass to the next step
6. Average the validation metrics obtained for the different test folds

This cross-validation strategy increases the models' potential to generalize across heterogeneous instances. The technique's stratification aspect supports the fidelity of the models' evaluation by conserving the distribution of class labels. Given the dataset's propensity for imbalanced classes, this attribute is of particular significance, as it can prevent any bias that might otherwise emerge from skewed class representation.

Supplementing the model selection approach was an evaluation for overfitting, a phenomenon that can compromise. For that effect, once trained using the entire dataset during cross-validation, the predictive model underwent a trial on a separate test split. This test split was obtained through the train-test dataset split. This division aimed to examine the model's performance on unseen data.

A scenario where the model exhibited a noticeable drop in performance on the test split compared to the cross-validated metrics could indicate a potential overfitting problem. These problems transpire when a model becomes exceedingly attuned to the train data's intricacies, damaging its potential to generalize proficiently to fresh data. This vigilant evaluation served as a crucial checkpoint to strengthen the robustness and reliability of the models selected for the prognostication endeavour.

4.3.2.4 Hyperparameter Tuning

Apart from the model selection strategies, hyperparameter tuning was also undertaken to select the best parameters for the chosen algorithms. To accomplish this, the hyperparameters of each trained model underwent refinement through the utilization of the tune_model function provided by the Pycaret library.

The criterion for model selection rests upon the optimal configuration of hyperparameters, which is guided by the metric chosen for optimization as discussed in Section 4.3.1.3. The ROC-AUC metric is presented with significance within this study, consequently emerging as the target for optimization efforts. The tune_model function encompasses a systematic grid search methodology to discern the optimal combination of parameter values.

The grid search methodology entails constructing an expansive grid containing diverse parameter configurations, subjecting each configuration to a cross-validation procedure. The main aim of this methodological approach is to unveil the set of parameter values that result in the highest achievable AUC, the chosen metric for hyperparameter tuning in this project.

The parameter grids employed in this context adhere to the default specifications inherent to the Pycaret library. This practice ensures adherence to established conventions while catering to the intricacies of the models and data under scrutiny.

4.3.3 Imaging-Data Models

For the pre-processing of the patient images, we chose a 3D approach. Despite being more expensive computationally, this decision was taken because all the information contained in the scans was considered important. If a bi-dimensional approach was taken, the correlations found between layers of the CNN could be lost. For example, the volumetric layout of the hemorrhage on the brain can provide valuable information for outcome prediction.

Based on the methodology proposed by Zeng et al. [52], the initial step involved the conversion of DICOM images into the NIfTI format. This conversion was facilitated by using the dicom2nifti library[2]. Using this library, various DICOM images associated with individual patients were amalgamated into a consolidated volume stored in the NIfTI format.

Following the conversion, the resultant volumes were subsequently processed using the nibabel library. Within these volumes, the raw voxel intensities were expressed in HU, ranging from -1024 to values exceeding 2000. Notably, the HU values related to bone structures exceeded 1000, signifying bone density, while values below -1000 represented air pockets.

To enhance the cohesiveness and comparability of the data, a rescaling procedure was applied. Values outside the range from 15-100 were set to 0, a conventional procedure for brain CT scan-oriented models as depicted by Pérez del Barrio et al. [37]. After this rescaling step, the values underwent a min-max normalization procedure, transforming them into a range between 0 and 1. This process was key in achieving a consistent and standardized set of voxel intensities across the dataset. By adjusting the intensities, the subsequent analysis could yield more meaningful insights, potentially restricted by variations in the original intensity ranges.

After normalizing, the volumes were downsampled to a size of 128x128x32 using spline interpolation. This decision was taken because the size is used by both of the described literature review solutions that employ a CNN and develop effective results [37, 52].

[2] https://pypi.org/project/dicom2nifti/

4.3.3.1 CNN architectures

Building upon the favourable outcomes achieved by the architecture formulated by Zeng et al. [52], the architectures designed in this study were inspired by this precedent. Hence, the constructed architectures constituted a 3D-CNN formed by eight convolutional layers, five maximum pool layers, and three dense (fully connected) layers. However, despite the good results made with these architectures, the development of these 3D-CNN models highlighted the emergence of overfitting issues, particularly within the context of the multilabel classification model. This was due to the sample size being way smaller when compared to the one used for the original architecture study (851 patients).

To counteract this concern, measures were taken to address overfitting by incorporating dropout and batch normalization layers. Specifically, the multilabel model was fortified with batch normalization and dropout layers, while the binary models were reinforced with dropout layers. The detailed architectural designs are in Appendix B.

These architectures (3D-CNNs) differentiate from conventional 2D-CNNs due to their capacity to extract feature maps from volumetric information by exploiting these volumes both globally and locally. These architectures were developed using the Keras Python library. Training was conducted for 100 epochs, employing a learning rate of 10^{-3} across all models. In each epoch, the AUC, accuracy and loss (estimated through loss function) metrics were gathered for the training and test set.

To update the network weights and minimize the loss, the optimization process relied on an adaptive moment estimation optimizer, commonly known as the Adam optimizer, which is an extension of the stochastic gradient descent. Each training batch consisted of 8 CT scans.

The chosen loss functions were tailored to the specific model types. For the binary models, the binary cross-entropy loss function was used. On the other hand, the multilabel models utilized the categorical cross-entropy loss function. In terms of activation functions, the rectified linear unit function was the one selected.

In model selection, a callback function was used to monitor and retain the best-performing model throughout 100 epochs. This callback function ensured the preservation of the model trained in the epoch that exhibited the lowest loss function value on the test dataset. The checkpoint function was important in this process, facilitating the storage and retrieval of the identified optimal model.

4.3.4 Multimodal-Data Models

As expounded in Section 4.2, the approach adopted employs a late fusion technique designed to synergize the prediction score derived from the best-performing models of both tabular and imaging solutions. The fusion methodology consists of a combination of prediction probabilities assigned to patients of specific classes.

The subsequent step in this process entails training a meta-classifier on the combined prediction scores. The meta-classifier is a high-level learning entity equipped to discern patterns and

associations from the outputs of other models. To ensure the efficacy of this meta-classifier, a training process similar to the one described for the tabular data model is employed, with a linear SVM selected as the meta-classifier. This choice is guided by the insights drawn from the study by Sadasivuni et al. [52], where the SVM with a linear kernel exhibited commendable performance.

As this section outlines, the late fusion approach displays the study's attempt to harness the latent potential present within diverse data modalities. This fusion strategy aims to amplify the predictive effectiveness of the final prognostic model, offering an integrated understanding of patients' outcomes that transcends the confines of individual data streams. The selection of this fusion approach was founded on the small dataset size, which has been proven to be preferable under a late fusion approach since the methods within that realm can be developed using classical ML techniques [33].

Chapter 5

Results and Discussion

This chapter is structured into three sections, each dedicated to presenting and discussing the results derived from the developed models. The initial section presents the outcomes and discussions related to the tabular data models. Subsequently, the second section delves into the imaging data models' results. Lastly, the third section is devoted to the models that emerged from amalgamating the solutions presented in the first and second sections.

5.1 Tabular Data-Models

As detailed in Subsections 4.3.2.2 and 4.3.2.1, an array of machine learning algorithms were harnessed, comprising four distinct models (RF, GBC, RBF-SVM and MLP). These algorithms were rigorously trained across ten distinct data pre-processing pipelines, resulting in an array of models tasked with the mission of prognosticating the outcomes of patients suffering from ICH. For this task, we adopted a classification approach, bifurcating the target variable, mRS, at cutoffs 3, 4, and 5 and was divided into three categories.

5.1.1 Results

As already approached in Section 4.3.1.3, to gauge the performance of the deployed algorithms and their corresponding pipelines, a trio of comprehensive metrics was employed. These metrics encompass the ROC-AUC, accuracy, and confusion matrices, the latter exclusively utilized during the testing phase (where the models were applied to test split) to obtain the class-wise accuracy to get a comprehensive view of the prediction distribution.

5.1.1.1 Algorithms performance

The extensive application of the implemented pre-processing pipelines in training the classification models yielded results that are depicted in tables C.1, C.2, C.3 and C.4, located in Appendix C. These provide a complete overview of the models' overall performance across various pipelines and solutions. Additionally, they highlight the best-performing models for each algorithm.

Given the relatively small size of the overall dataset, we observed some unusual variance in performance metrics when comparing cross-validation results with the test results. Due to this effect, we considered the cross-validation ROC-AUC value as the primary metric for estimating the overall effectiveness of the models and, thus, for selecting the best models. Consequently, our selection criterion for determining the best-performing algorithms revolved around choosing the highest cross-validation AUC score. In cases where multiple pipelines yielded the same maximum AUC value, we prioritize models based on their cross-validation accuracy. If accuracy results were also identical, we turned to the test set's AUC, followed by test accuracy, to make our selection. In the unlikely event that all these metrics remained equivalent, we opted for the pipeline with lower computational costs.

However, it's worth noting that class-wise accuracy on the test set played a crucial role in understanding the behaviour of the models, particularly the impact of class imbalances within the dataset. As the overall results reveal, models with lower accuracies but higher ROC-AUC values proved to be better choices due to class imbalances. These imbalances can provoke certain tendencies to predict one class over the others. This was observed in the binary models, in which there was, in most cases, a class with higher accuracy in the test set than the other, even with oversampling used. This can also be a product of the small sample size, which can make the encountering of patterns in data more difficult. Within the overall models, the following tendencies were found:

- **Binary model for the mRS cutoff of 3** - The classifications were biased towards bad outcomes
- **Binary model for the mRS cutoff of 4** - The classifications tended to predict good outcomes
- **Binary model to predict whether or not the patient dies (mRS cutoff of 5)** - The accuracies in detecting deaths were, overall, lower
- **Multilabel model** - Accuracy for class 2 (death outcome) tends to be lower

In the context of our tabular data experiment pipelines, we observed that the normalization process had a generally positive impact on the overall training process. This effect was most pronounced in the case of SVM and MLP models, where normalization played a significant role in improving performance. However, for RF and GBC models, the impact of normalization was less pronounced, and these models performed reasonably well even without it.

Furthermore, when normalization was combined with iterative imputation, we observed considerable improvements in the performance of the RF and GBC solutions. This suggests that the synergy between imputation and normalization can be particularly beneficial for these algorithms.

Additionally, our experiments revealed that the SMOTE oversampling strategy proved advantageous for RF and GBC algorithms, mainly for the binary model for the mRS cutoff of 3 (the most imbalanced). This technique helped these models by addressing class imbalances in the dataset.

Overall, the combination of preprocessing techniques such as normalization, imputation, and oversampling had a substantial impact on the performance of the machine learning algorithms, and the specific benefits varied depending on the algorithm's characteristics and the dataset's characteristics.

To represent the better-performing algorithms, Table 5.1 depicts these algorithm cross-validation performances, respective hyperparameters and pre-processing pipelines (encountered in Table 4.3):

mRS	Model	Train split		Pipelines	Hyperparameters
		Acc	AUC		
> 3	**GBC**	74.5%	64.2%	P8	Listing C.1
	SVM	75.5%	70%	P5	
	RF*	73%	83.3%	P8	
	MLP	75%	66.7%	P3	
> 4	**GBC**	67%	68.3%	P8	Listing C.2
	SVM	69%	72.5%	P4	
	RF*	64.5%	78.3%	P1	
	MLP	65%	66.7%	P9	
> 5	**GBC**	61.5%	75%	P9	Listing C.3
	SVM	67%	61.7%	P2	
	RF*	70%	78.8%	P6	
	MLP	65%	73.3%	P7	
multi	**GBC**	53.5%	76.6%	P0	Listing C.4
	SVM	52%	64.7%	P8	
	RF*	59%	79.1%	P0	
	MLP	44%	71.3%	P8	

Table 5.1: Best models cross-validation performance metrics, hyperparameters and pre-processing pipelines

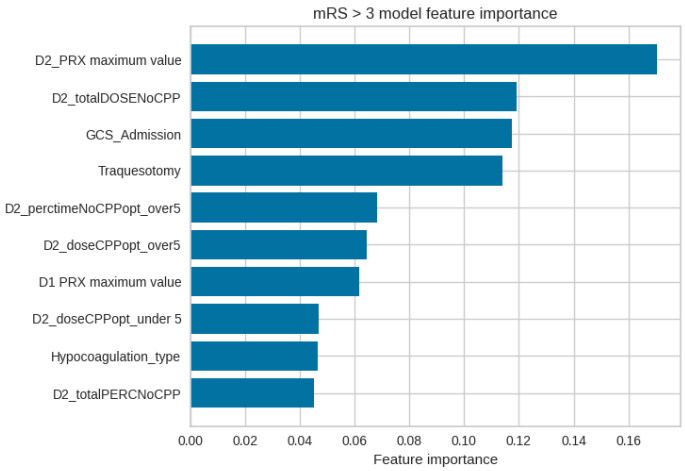
*Models selected for the multimodal solution

As can be observed, the better-performing model among all solutions is the Random Forest. As such, the prediction scores extracted for each patient by these models will be used as learning parameters for the multimodal solution.

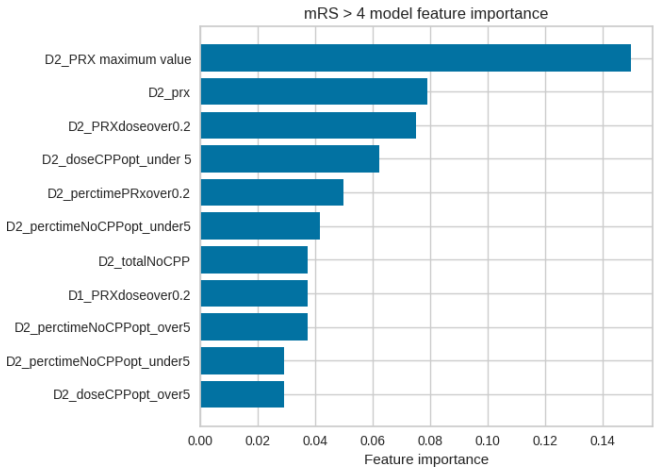
5.1.1.2 Feature Importance

A complementary feature importance study was also undertaken to shed light on the variables contributing significantly to the predictive power of the algorithms. The top 10 features have been extracted from the best-performing random forest algorithms. This feature importance analysis is a valuable resource for understanding the key determinants influencing the functional statuses of the different patients at different levels.

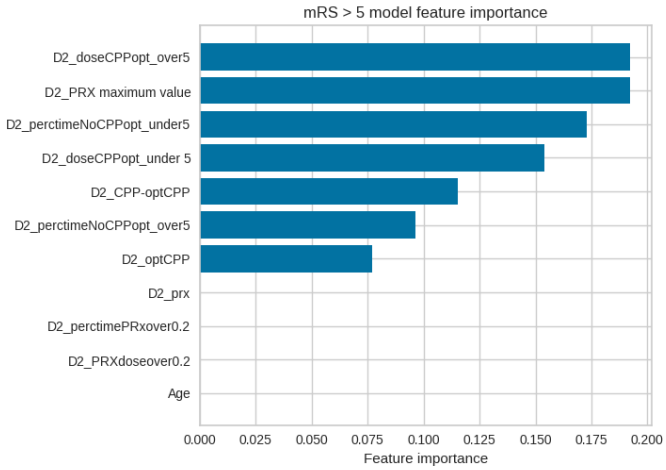
In Figure 5.1, the top 10 features are ordered by their weight or feature importance for the predictions carried by the models.



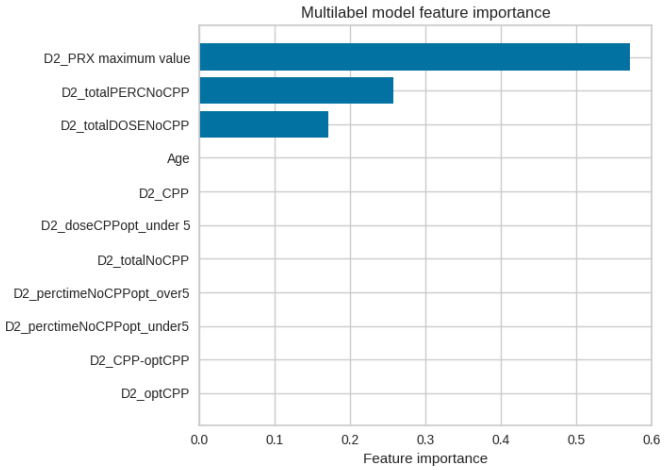
(a) Binary Model for mRS cutoff of 3



(b) Binary Model for mRS cutoff of 4



(c) Binary Model for mRS cutoff of 5



(d) Multilabel Model

Figure 5.1: Top 10 features for each model (feature importance in descending order)

In the scope of these four studies, several expected variables commonly employed in autoregulation practices for neuromonitoring, such as PRx and CPPopt, consistently appeared with high rankings in the top 10 feature importance lists. Notably, these variables exhibited their significance, particularly when measured on the second day following admission to the ICU. This observation aligns with expectations, as prior research has demonstrated their reliability in predicting outcomes for neurocritical patients [16, 5] and further reinforced by the multilabel model, which

only considers three variables, the maximum PRx and the time dose and percentage without CP-Popt on the second day after the ICH.

Furthermore, it is crucial to underscore the importance of other variables, such as the GCS score upon admission, which played a prominent role in the binary model at the cutoff of 3.

While it was anticipated that the variable indicating whether a patient had undergone tracheostomy would feature prominently in predicting the mortality of the patients, its performance in predicting outcomes at $mRS \geq 3$ is noteworthy. This observation aligns with previous studies showing a strong association between tracheostomy and subsequent patient mortality or dependence six months after admission[41].

In the model designed for mortality and multilabel outcome prediction, a curious characteristic was the consideration of only a small set of attributes for the final predictions. This selection underscores the critical relevance of these specific attributes in forecasting patient outcomes for the particular purposes of those models.

5.1.2 Discussion

Keeping the primary goal of our study in perspective, we developed a set of prognostic tools designed to forecast patient outcomes in cases of ICH by harnessing demographic, clinical, and MCM data. This multifaceted task entailed the employment of diverse pre-processing pipelines, each applied to four distinct supervised learning algorithms, all primed for the classification mission. As elucidated in the preceding section, the utilization of diverse pre-processing pipelines yielded substantial benefits, sometimes resulting in notable variations in the AUC metric, ranging, for example, from 50% to 70% across pipelines for a single algorithm.

The comprehensive results presented in the previous section highlight random forest as the better-performing algorithm. This alignment with the findings in Section 3.2.1 is noteworthy, where RF consistently demonstrated its effectiveness in various prediction tasks. Beyond its prowess in prediction, RF played a dual role in our study, as it was also employed in conducting a feature importance analysis. This analysis further underscored the significance of variables that have already been validated as pivotal indicators for certain patient outcomes in the realm of neurocritical pathologies. Moreover, it sheds light on the indispensable role played by MCM data in predicting these outcomes.

When we juxtapose the performance of our project's solutions against those documented in the literature review for tabular data [29, 49, 30], we observe a nuanced picture. It becomes evident that our implemented solutions, while promising, do not attain the same level of efficacy as their literature counterparts. This variance can be largely attributed to the size of our dataset, which is notably smaller when compared to the sample sizes prevalent in the literature.

Despite these disparities with the literature review solutions, our incursion into this domain has yielded promising results, with all AUC values consistently approaching 80%, and one of them even surpassing it (binary model at cutoff 3). However, it is noteworthy that the corresponding accuracy scores appeared relatively lower than the AUC metrics. Such disparities often hint at classifier bias towards a particular class, especially in the presence of data imbalances. Mitigating

this issue could involve fine-tuning the decision threshold, potentially leading to an enhancement in the model's accuracy by ensuring a more balanced prediction of both classes.

Overfitting, on the other hand, presented a unique challenge. The small size of our test splits made the detection of overfitting difficult. To obtain more robust and reliable assessments of model generalization and overfitting, it would be advisable to validate the obtained models on a larger dataset. Additionally, for training models that perform better overall and generalize well to new data, acquiring a larger training dataset would also be recommended. Larger datasets provide a more representative sample of the underlying data distribution and can help assess and mitigate overfitting more effectively.

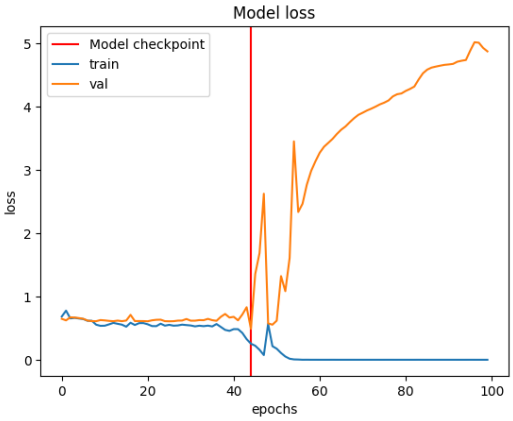
5.2 Imaging models

Similar to the previous section, we present, accompanied by a discussion, the results of the four idealized models, but, in this case, trained on CT scans. These imaging models are specifically tailored to image data. They will depict the training process of the CNNs and the performance of the selected models on both the train and test splits of the dataset.

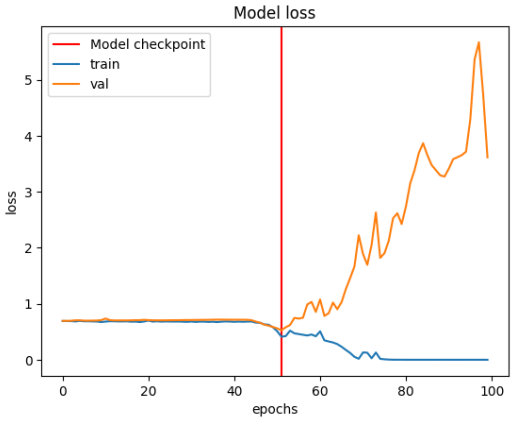
5.2.1 Results

To develop the four models, CNNs were trained over 100 epochs, continuously monitoring the loss value for each epoch on both the train and test splits. The loss evolution for the different models is visible in Figure 5.2.

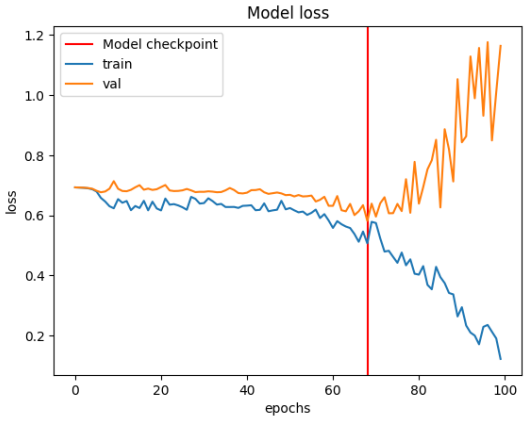
Within the training split of our dataset, we observe the expected behaviour – a progressive reduction in the loss value as the training process unfolds. However, a different pattern emerges when we focus on the test split, where a swift rise in the loss value becomes evident. This contrast in behaviours is a clear indicator of overfitting, where the neural network has excessively tailored itself to the nuances of the training dataset. As elaborated in Section 4.3.3.1, to address this overfitting challenge, we employed a checkpoint strategy. Specifically, we saved a checkpoint each time the loss value reached a new minimum. These checkpoints are denoted in the figures by vertical red lines and, curiously, mark the points at which overfitting became notably pronounced.



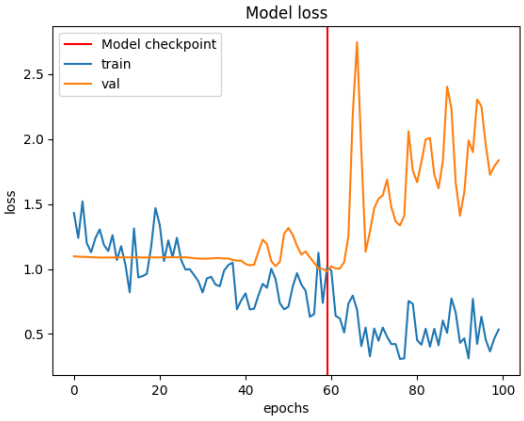
(a) Binary Model for mRS cutoff of 3



(b) Binary Model for mRS cutoff of 4



(c) Binary Model for mRS cutoff of 5



(d) Multilabel Model

Figure 5.2: CNN models loss per epoch of training

With the model weights selected through the training process, we extracted accuracy and AUC metrics for both the training and test splits. This evaluation served as a critical measure of the model's performance and its ability to generalize beyond the training data. Subsequently, we compiled the findings into Table 5.2, which is presented below:

Model	**Train split**		**Test split**	
	Acc	**AUC**	**Acc**	**AUC**
mRS > 3	81.3%	93.1%	76.5%	80%
mRS > 4	81.3%	83.4%	70.6%	70.1%
mRS > 5	70.8%	84.3%	71.3%	76.4%
multilabel	60%	78.1%	55%	69.8%

Table 5.2: Imaging data models results

5.2.2 Discussion

In alignment with our overarching objectives, we engineered a predictive tool akin to the one devised for tabular data. However, a notable distinction lies in using images rather than tabular data. Consequently, this section is dedicated to elucidating how the outcomes of our image-based tool compare to those of the tabular counterpart and how they stack up against the set of existing literature.

The output metrics derived from the imaging-based models exhibit an apparent divergence between the training and test splits, strongly indicative of potential overfitting to the training data, caused most likely by the small sample size of the dataset. Consequently, our evaluation predominantly centers on the test subset, unlike the tabular data models, where we primarily scrutinized the training performance. When we delve specifically into the results obtained from the imaging models, it becomes discernible that certain models outperform others, as evidenced by their superior accuracy and AUC metrics.

This performance hierarchy among the imaging models becomes apparent, with the multilabel model exhibiting the lowest performance, followed by the binary model at cutoff 4, followed by the model at cutoff 5, and the most proficient being the model at cutoff 3.

In the comparative analysis between the optimal tabular data models and their imaging counterparts, a notable distinction emerges in performance evaluation. Specifically, when evaluating accuracy, the imaging models showcase superior results across the board, except for the multilabel model. However, upon closer examination of the AUC metric, the tabular models consistently exhibit higher values.

In light of this evaluation, it is essential to emphasize the significance of data imbalances within the dataset, as previously elucidated in preceding sections. In such scenarios, prioritising the AUC metric is important, as it offers a more robust assessment of model performance, particularly when confronted with class imbalances.

Considering this, the tabular models emerge as the preferable choice (if they were to be used independently), boasting superior overall results when measured against the critical AUC metric. This preference for tabular models over imaging may be subject to revision in the presence of a larger dataset. It is a well-established principle that neural networks exhibit enhanced training performance when provided with larger and more comprehensive datasets. Such an expansion in data size could potentially lead to reevaluating the relative performance of tabular and imaging models.

Comparing now to the literature review presented works, it is relevant to refer that the imaging side of the multimodal solution pursued by Pérez del Barrio et al. [37] had an AUC close to 76.4% obtained by the model for mortality prediction. Contrarily, the rest of the works presented in Section 3.2.2 and more image-oriented had significantly higher results. Again, this was probably a result of the fact that their sample size was higher.

In comparing our results to the works presented in the literature review (imaging-only studies), particularly in the imaging side of the multimodal solution pursued by Pérez del Barrio et al. [37],

we observe an AUC close to 76.4% obtained by their model for mortality prediction (76.3%). In contrast, the other works, more focused on image-oriented approaches and presented in Section 3.2.2, consistently yielded significantly higher results. Once again, this variance can likely be attributed to the larger sample sizes employed in these studies, underscoring the pivotal role of data size in influencing model performance.

5.3 Multimodal Models

After the creation of both imaging and tabular models, the integration of these models culminated in forming an ensemble of models orchestrated through a metaclassifier. This section presents the results derived from the combination of prediction scores generated by the imaging and tabular models. Furthermore, a brief evaluation of the performance metrics is expounded upon.

5.3.1 Results

Upon the realisation of the imaging and tabular models, we implemented the late fusion approach, and the outcomes have been consolidated into Table 5.3:

Table 5.3: Multimodal data performance metrics

Model	**Train split**		**Test split**	
	Acc	**AUC**	**Acc**	**AUC**
mRS > 3	92%	100%	100%	100%
mRS > 4	83%	96.7%	82%	96%
mRS > 5	80%	93.3%	82.5%	95%
multilabel	64.5%	87.5%	76.4%	86.7%

5.3.2 Discussion

As elucidated earlier, the amalgamation of two distinct sets of models, each sharing a common object but train on divergent data modalities, has been successfully developed through a late fusion strategy, culminating by combining model prediction scores. This approach has yielded an overall favourable outcome, manifesting enhancements in both accuracy and AUC metrics when compared with the performance of the individual imaging and tabular models. Significantly, throughout the evaluation of the acquired multimodal results, there has been no apparent indication of overfitting.

Aligned with the multimodal literature review solutions [37, 40, 34], the attained performance across various models has demonstrated satisfactory outcomes, with some even surpassing the metrics reported in the reviewed studies, particularly in terms of AUC. However, these results with a grain of salt.

An important facet to underscore is that the overfitting observed in the imaging models introduces a susceptibility to an overly positive multimodal solution. This effect might be hidden in the

multimodal models, primarily due to the possibility of data points in the test split (used to validate the combined models) having been exposed to training within the overfitted imaging models. This interconnectedness between the imaging and multimodal models necessitates a cautious interpretation of our findings and suggests the need for further exploration and refinement to mitigate potential overfitting challenges, for instance, by evaluating the model using another dataset with a set of additional patients.

The results obtained from our study have shed light on the progressive difficulty of predicting higher severity outcomes in patients afflicted with ICH using machine learning techniques. This trend is particularly pronounced in the binary models, which exhibit diminishing performance as higher mRS scores are employed as cutoffs for defining good or bad outcomes. Furthermore, our findings highlight the increased complexity introduced by the multilabel approach in predicting patient outcomes, as this approach yields the least favourable results among all the models evaluated.

Chapter 6

Conclusions

The integration of machine learning techniques within the realm of neurology, specifically for predicting outcomes in neurocritical patients with traumatic and non-traumatic brain injuries, has seen significant growth. These predictive models hold immense potential in aiding medical professionals in prognosticating patient outcomes and allocating resources effectively for patient rehabilitation. However, regarding the specific pathology of ICH, such machine-learning approaches remain relatively scarce, especially those applied to multimodal datasets.

To address this research gap, this dissertation embarked on developing models that utilize multimodal datasets, encompassing both tabular and imaging data, as inputs for predicting patient outcomes.

The journey began with a thorough analysis of existing state-of-the-art works akin to the one envisioned in this research. This exploration, detailed in Section 3.2, answered four critical research questions. It involved examining selected studies from a plethora of projects, extracting valuable insights, and identifying techniques and concepts that would form the foundation for the proposed solution and methodology.

The reviewed studies not only provided answers to pertinent questions but also offered solutions with satisfactory results. They enriched this thesis's work and facilitated the formulation of a comprehensive resolution to the problem at hand.

Subsequently, following the review of the existing studies, a set of prognostication models was developed, obtained from the developed imaging and tabular data prognostication tools. These models collectively aimed to predict the 6-month outcomes of ICH patients at various levels, using the mRS as the reference point for assessing patients' functional statuses. The models included three binary outcome prediction models at different mRS binarization cutoffs and one multilabel outcome prediction model.

Upon examining the results, it becomes evident that the multimodal prognostication tools yielded a set of commendable outcomes. Cross-validation AUC values for these models exceeded those of the individual tabular and imaging models, as well as some of the studies reviewed in the literature:

- Binary outcome prediction at cutoff 3: 100%

- Binary outcome prediction at cutoff 4: 96.7%
- Binary outcome prediction at cutoff 5: 93.3%
- Multilabel outcome prediction: 87.5%

However, it is important to approach these results with a degree of caution, as discussed in Section 5.3.2. There exists the possibility of overfitting, primarily stemming from the overfitting observed in the imaging data models and possibly from the tabular data models (undetected), which could potentially transfer to the multimodal data without immediate detection.

Another noteworthy insight is the pivotal role of specific features, such as PRx and CPPopt, in enhancing the predictive capabilities of tabular models. The feature importance study underscored the effectiveness of features obtained through MCM in enhancing algorithmic predictive performance.

In conclusion, the objective of this thesis study is achieved, since, while there is a degree of scepticism due to the potential presence of overfitting in the final model, the overall outcomes of this study seem to underscore the suitability of machine learning methodologies for predicting outcomes in ICH patients using multimodal datasets, encompassing MCM data variables and, potentially if the overfitting is abolished, neuroimaging.

6.1 Limitations and Future Work

The primary limitation of this study derives from the dataset's relatively small sample size. While it offered a satisfactory variable size, its limited scale raises questions and presents challenges related to overfitting and data distribution (e.g., imbalances). To strengthen the credibility of the conclusions drawn in this work, future endeavours should involve larger, more diverse datasets, enabling the evaluation and development of models with more conclusive results and better equipped to detect realistic patterns and tackle overfitting and biases.

Moreover, further investigations are warranted in the domain of imaging-based models and fusion techniques for the multimodal solution. Future research could explore the training of models with varying hyperparameters for 3D-CNNs and pre-processing steps to identify the optimal configurations for training ICH CT scan outcome prediction models and could explore other fusion strategies like early fusion.

In summary, following the implementation of the aforementioned steps and provided that the results consistently align with the conclusions drawn, the solutions developed in this study hold the potential for practical application in real-life scenarios. The prognostication tools, if effective, will give the medical professionals sufficient insights to establish therapeutic ceilings, thus preventing the use of excessive resources for rehabilitating these patients.

References

[1] DICOM homepage. Available at https://www.dicomstandard.org, July 2023.

[2] NIfTI: — Neuroimaging Informatics Technology Initiative. Available at https://nifti.nimh.nih.gov/, July 2023.

[3] Moez Ali. *PyCaret: An open source, low-code machine learning library in Python*, April 2020. PyCaret version 1.0.0.

[4] Franck Amyot, David B. Arciniegas, Michael P. Brazaitis, Kenneth C. Curley, Ramon Diaz-Arrastia, Amir Gandjbakhche, Peter Herscovitch, Sidney R. Hinds, Geoffrey T. Manley, Anthony Pacifico, Alexander Razumovsky, Jason Riley, Wanda Salzer, Robert Shih, James G. Smirniotopoulos, and Derek Stocker. A Review of the Effectiveness of Neuroimaging Modalities for the Detection of Traumatic Brain Injury. *Journal of Neurotrauma*, 32(22):1693–1721, November 2015.

[5] Marcel J. H. Aries, Marek Czosnyka, Karol P. Budohoski, Luzius A. Steiner, Andrea Lavinio, Angelos G. Kolias, Peter J. Hutchinson, Ken M. Brady, David K. Menon, John D. Pickard, and Peter Smielewski. Continuous determination of optimal cerebral perfusion pressure in traumatic brain injury. *Critical Care Medicine*, 40(8):2456–2463, August 2012.

[6] Nisha Arya. Why Use k-fold Cross Validation? Available at https://www.kdnuggets.com/why-use-k-fold-cross-validation.html, August 2023. Section: KDnuggets Evergreen.

[7] Mohammad Samy Baladram, Atsushi Koike, and Kazunori D. Yamada. Introduction to Supervised Machine Learning for Data Science. *Interdisciplinary Information Sciences*, 26(1):87–121, 2020.

[8] Renesh Bedre. Support Vector Machine (SVM) basics and implementation in Python. Available at https://www.reneshbedre.com/blog/support-vector-machine.html, July 2020.

[9] Leo Breiman. Random Forests. *Machine Learning*, 45(1):5–32, October 2001.

[10] Matthew Brett, Christopher J. Markiewicz, Michael Hanke, Marc-Alexandre Côté, Ben Cipollini, Paul McCarthy, Dorota Jarecka, Christopher P. Cheng, Yaroslav O. Halchenko, Michiel Cottaar, Eric Larson, Satrajit Ghosh, Demian Wassermann, Stephan Gerhard, Gregory R. Lee, Zvi Baratz, Hao-Ting Wang, Erik Kastman, Jakub Kaczmarzyk, Roberto Guidotti, Jonathan Daniel, Or Duek, Ariel Rokem, Cindee Madison, Dimitri Papadopoulos Orfanos, Anibal Sólon, Brendan Moloney, Félix C. Morency, Mathias Goncalves, Ross Markello, Cameron Riddell, Christopher Burns, Jarrod Millman, Alexandre Gramfort, Jaakko Leppäkangas, Jasper J.F. van den Bosch, Robert D. Vincent, Henry Braun,

Krish Subramaniam, Andrew Van, Krzysztof J. Gorgolewski, Pradeep Reddy Raamana, Julian Klug, B. Nolan Nichols, Eric M. Baker, Soichi Hayashi, Basile Pinsard, Christian Haselgrove, Mark Hymers, Oscar Esteban, Serge Koudoro, Fernando Pérez-García, Jérôme Dockès, Nikolaas N. Oosterhof, Bago Amirbekian, Horea Christian, Ian Nimmo-Smith, Ly Nguyen, Samir Reddigari, Samuel St-Jean, Egor Panfilov, Eleftherios Garyfallidis, Gael Varoquaux, Jon Haitz Legarreta, Kevin S. Hahn, Lea Waller, Oliver P. Hinds, Bennet Fauber, Fabian Perez, Jacob Roberts, Jean-Baptiste Poline, Jon Stutters, Kesshi Jordan, Matthew Cieslak, Miguel Estevan Moreno, Tomáš Hrnčiar, Valentin Haenel, Yannick Schwartz, Benjamin C Darwin, Bertrand Thirion, Carl Gauthier, Igor Solovey, Ivan Gonzalez, Jath Palasubramaniam, Justin Lecher, Katrin Leinweber, Konstantinos Raktivan, Markéta Calábková, Peter Fischer, Philippe Gervais, Syam Gadde, Thomas Ballinger, Thomas Roos, Venkateswara Reddy Reddam, and freec84. nipy/nibabel: 5.1.0. Available at https://zenodo.org/record/7795644, April 2023.

[11] Jason Brownlee. A Gentle Introduction to Ensemble Learning Algorithms. Available at https://machinelearningmastery.com/tour-of-ensemble-learning-algorithms/, April 2021.

[12] Junhua Chen, Leonard Wee, Andre Dekker, and Inigo Bermejo. Using 3D deep features from CT scans for cancer prognosis based on a video classification model: A multi-dataset feasibility study. *Medical Physics*, 50(7):4220–4233, 2023. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/mp.16430.

[13] Sasank Chilamkurthy, Rohit Ghosh, Swetha Tanamala, Mustafa Biviji, Norbert G Campeau, Vasantha Kumar Venugopal, Vidur Mahajan, Pooja Rao, and Prashant Warier. Deep learning algorithms for detection of critical findings in head CT scans: a retrospective study. *The Lancet*, 392(10162):2388–2396, December 2018.

[14] François Chollet et al. Keras. Available at https://keras.io, July 2023.

[15] Krzysztof J. Cios, Roman W. Swiniarski, Witold Pedrycz, and Lukasz A. Kurgan. The Knowledge Discovery Process. In Krzysztof J. Cios, Roman W. Swiniarski, Witold Pedrycz, and Lukasz A. Kurgan, editors, *Data Mining: A Knowledge Discovery Approach*, pages 9–24. Springer US, Boston, MA, 2007.

[16] Marek Czosnyka, Zofia Czosnyka, and Peter Smielewski. Pressure reactivity index: journey through the past 20 years. *Acta Neurochirurgica*, 159(11):2063–2065, November 2017.

[17] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From Data Mining to Knowledge Discovery in Databases. *AI Magazine*, 17(3):37–37, March 1996. Number: 3.

[18] Jerome H. Friedman. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4):367–378, February 2002.

[19] Henry Han and Xiaoqian Jiang. Overcome Support Vector Machine Diagnosis Overfitting. *Cancer Informatics*, 13(Suppl 1):145–158, December 2014.

[20] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant,

Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

[21] Simon S. Haykin and Simon S. Haykin. *Neural networks and learning machines*. Prentice Hall, New York, 3rd ed edition, 2009. OCLC: ocn237325326.

[22] J. Claude Hemphill, David C. Bonovich, Lavrentios Besmertis, Geoffrey T. Manley, and S. Claiborne Johnston. The ICH score: A simple, reliable grading scale for intracerebral hemorrhage. 32(4):891–897.

[23] David Y. Hwang, Cameron A. Dell, Mary J. Sparks, Tiffany D. Watson, Carl D. Langefeld, Mary E. Comeau, Jonathan Rosand, Thomas W.K. Battey, Sebastian Koch, Mario L. Perez, Michael L. James, Jessica McFarlin, Jennifer L. Osborne, Daniel Woo, Steven J. Kittner, and Kevin N. Sheth. Clinician judgment vs formal scales for predicting intracerebral hemorrhage outcomes. *Neurology*, 86(2):126–133, January 2016.

[24] Jones J, Sharma R, Qureshi P, et al. Intracerebral hemorrhage. Available at Radiopaedia.org, February 2023.

[25] Rohit Kundu. F1 Score in Machine Learning: Intro & Calculation. Available at https://www.v7labs.com/blog/f1-score-guide, August 2023.

[26] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, May 2015. Number: 7553 Publisher: Nature Publishing Group.

[27] M. H. Lev and R. G. Gonzalez. 17 - CT Angiography and CT Perfusion Imaging. In Arthur W. Toga and John C. Mazziotta, editors, *Brain Mapping: The Methods (Second Edition)*, pages 427–484. Academic Press, San Diego, second edition edition, 2002.

[28] Andy Liaw and Matthew Wiener. Classification and Regression by randomForest. 2, 2002.

[29] Ching-Heng Lin, Kai-Cheng Hsu, Kory R. Johnson, Yang C. Fann, Chon-Haw Tsai, Yu Sun, Li-Ming Lien, Wei-Lun Chang, Po-Lin Chen, Cheng-Li Lin, and Chung Y. Hsu. Evaluation of machine learning methods to stroke outcome prediction using a nationwide disease registry. *Computer Methods and Programs in Biomedicine*, 190:105381, July 2020.

[30] Kazuya Matsuo, Hideo Aihara, Tomoaki Nakai, Akitsugu Morishita, Yoshiki Tohma, and Eiji Kohmura. Machine Learning to Predict In-Hospital Morbidity and Mortality after Traumatic Brain Injury. *Journal of Neurotrauma*, 37(1):202–210, July 2019. Publisher: Mary Ann Liebert, Inc., publishers.

[31] Wes McKinney et al. Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, volume 445, pages 51–56. Austin, TX, 2010.

[32] Francisco Melo. Area under the ROC Curve. In Werner Dubitzky, Olaf Wolkenhauer, Kwang-Hyun Cho, and Hiroki Yokota, editors, *Encyclopedia of Systems Biology*, pages 38–39. Springer, New York, NY, 2013.

[33] Farida Mohsen, Hazrat Ali, Nady El Hajj, and Zubair Shah. Artificial intelligence-based methods for fusion of electronic health records and imaging data. *Scientific Reports*, 12:17981, October 2022.

[34] Jawed Nawabi, Helge Kniep, Sarah Elsayed, Constanze Friedrich, Peter Sporns, Thilo Rusche, Maik Böhmer, Andrea Morotti, Frieder Schlunk, Lasse Dührsen, Gabriel Broocks, Gerhard Schön, Fanny Quandt, Götz Thomalla, Jens Fiehler, and Uta Hanning. Imaging-Based Outcome Prediction of Acute Intracerebral Hemorrhage. *Translational Stroke Research*, 12(6):958–967, December 2021.

[35] Seungwon Oh, Sae-Ryung Kang, In-Jae Oh, and Min-Soo Kim. Deep learning model integrating positron emission tomography and clinical data for prognosis prediction in non-small cell lung cancer patients. *BMC Bioinformatics*, 24(1):39, February 2023.

[36] Neel T. Patel and Scott D. Simon. Intracerebral hemorrhage. Available at https://www.aans.org/en/Patients/Neurosurgical-Conditions-and-Treatments/Intracerebral-Hemorrhage, January 2023.

[37] Amaia Pérez del Barrio, Anna Salut Esteve Domínguez, Pablo Menéndez Fernández-Miranda, Pablo Sanz Bellón, David Rodríguez González, Lara Lloret Iglesias, Enrique Marqués Fraguela, Andrés A. González Mandly, and José A. Vega. A deep learning model for prognosis prediction after intracranial hemorrhage. *Journal of Neuroimaging*, 33(2):218–226, 2023. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/jon.13078.

[38] An Ramesh, C Kambhampati, Jrt Monson, and Pj Drew. Artificial intelligence in medicine. *Annals of The Royal College of Surgeons of England*, 86(5):334–338, September 2004.

[39] Chethan P. Venkatasubba Rao, Jose I. Suarez, Renee H. Martin, Colleen Bauza, Alexandros Georgiadis, J. Claude Hemphill III Eusebia Calvillo, Gene Sung, Mauro Oddo, Fabio Silvio Taccone, Peter D. LeRoux, et al. Global Survey of Outcomes of Neurocritical Care Patients: Analysis of the PRINCE Study Part 2. *Neurocritical Care*, 32:88–103, February 2020.

[40] Sudarsan Sadasivuni, Monjoy Saha, Neal Bhatia, Imon Banerjee, and Arindam Sanyal. Fusion of fully integrated analog machine learning classifier with electronic medical records for real-time prediction of sepsis onset. *Scientific Reports*, 12:5711, April 2022.

[41] David B. Seder. Tracheostomy Practices in Neurocritical Care. *Neurocritical Care*, 30(3):555–556, June 2019.

[42] Sertan Serte and Hasan Demirel. Deep learning for diagnosis of COVID-19 using 3D CT scans. *Computers in Biology and Medicine*, 132:104306, May 2021.

[43] Abhishek Sharma. Random Forest vs Decision Tree | Which Is Right for You? Available at https://www.analyticsvidhya.com/blog/2020/05/decision-tree-vs-random-forest-algorithm/, May 2020.

[44] P. Smielewski, A. Lavinio, I. Timofeev, D. Radolovich, I. Perkes, J. D. Pickard, and M. Czosnyka. ICM+, a Flexible Platform for Investigations of Cerebrospinal Dynamics in Clinical Practice. *Acta Neurochirurgica Supplements*, pages 145–51, 2008.

[45] Richard Szeliski. *Computer Vision - Algorithms and Applications, Second Edition*. Texts in Computer Science. Springer, 2022.

[46] Kai Ming Ting. Confusion Matrix. In Claude Sammut and Geoffrey I. Webb, editors, *Encyclopedia of Machine Learning and Data Mining*, pages 260–260. Springer US, Boston, MA, 2017.

[47] Guido Van Rossum and Fred L. Drake. *Python 3 Reference Manual*. CreateSpace, Scotts Valley, CA, 2009.

[48] John van Swieten. Modified rankin scale for neurologic disability. Available at https://www.mdcalc.com/calc/1890/modified-rankin-scale-neurologic-disability, February 2023.

[49] Hsueh-Lin Wang, Wei-Yen Hsu, Ming-Hsueh Lee, Hsu-Huei Weng, Sheng-Wei Chang, Jen-Tsung Yang, and Yuan-Hsiung Tsai. Automatic Machine-Learning-Based Outcome Prediction in Patients With Primary Intracerebral Hemorrhage. *Frontiers in Neurology*, 10:910, 2019.

[50] Wikipedia. Deep learning — Wikipedia, the free encyclopedia. Available at http://en.wikipedia.org/w/index.php?title=Deep%20learning&oldid=1175441072, 2023.

[51] Xulei Yang, Q. Song, and Aize Cao. Weighted support vector machine for data classification. volume 21, pages 859 – 864 vol. 2, 08 2007.

[52] Ying Zeng, Chen Long, Wei Zhao, and Jun Liu. Predicting the Severity of Neurological Impairment Caused by Ischemic Stroke Using Deep Learning Based on Diffusion-Weighted Images. *Journal of Clinical Medicine*, 11(14):4008, January 2022. Number: 14 Publisher: Multidisciplinary Digital Publishing Institute.

Appendix A

Variables obtained through ICM+ MCM

Variable	Description
ABP	Arterial Blood Pressure mean value
HR	Heart Rate mean value
ICP	ICP mean value
ICP_over21	Percentage of time ICP was over 21 mmHg
ICPdose	Area under the curve (time dose) of ICP over 21 mmHg
ICP max Value	ICP maximum value
PRX	PRx mean value
perctimePRxover0.2	Percentage of time PRx was over 0.2
PRXdoseover0.2	Time dose of PRx over 0.2
PRX maximum value	PRX maximum value
optCPP	Optimal CPP value
CPP-optCPP	Difference between mean CPP and optimal CPP
perctimeCPPopt_under5	Percentage of time CPP was under CPP optimal minus 5mmhg
perctimeCPPopt_over5	Percentage of time CPP was under CPP optimal plus 5mmhg
doseCPPopt_under 5	Time dose of CPP under CPP optimal minus 5mmHg
doseCPPopt_over 5	Time dose of CPP under CPP optimal plus 5mmHg
totalNoCPP	Time dose the patient was without CPPopt
CO2	End-tidal carbon dioxide mean value
RAC	RAC (correlation between the amplitude of the fundamental harmonic of ICP pulse wave and CPP) mean
RAP	RAP (brain compliance index) mean

Table A.1: MCM Variables extracted from ICM+

Appendix B

3D CNN Architectures

Listing B.1: Binary models 3D CNN architecture

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input (InputLayer)           [(None, 128, 128, 32, 1)  0
                            ]

conv1 (Conv3D)               (None, 128, 128, 32, 16)  448

maxPool1 (MaxPooling3D)      (None, 64, 64, 16, 16)    0

conv2 (Conv3D)               (None, 32, 32, 8, 32)     13856

maxPool2 (MaxPooling3D)      (None, 16, 16, 4, 32)     0

conv3 (Conv3D)               (None, 16, 16, 4, 64)     55360

conv4 (Conv3D)               (None, 8, 8, 2, 64)       110656

maxPool3 (MaxPooling3D)      (None, 4, 4, 1, 64)       0

conv5 (Conv3D)               (None, 4, 4, 1, 96)       165984

conv6 (Conv3D)               (None, 2, 2, 1, 96)       248928

maxPool4 (MaxPooling3D)      (None, 1, 1, 1, 96)       0

conv7 (Conv3D)               (None, 1, 1, 1, 128)      331904

conv8 (Conv3D)               (None, 1, 1, 1, 128)      442496

maxPool5 (MaxPooling3D)      (None, 1, 1, 1, 128)      0

dense1 (Dense)               (None, 1, 1, 1, 128)      16512
```

```
dropout1 (Dropout)          (None, 1, 1, 1, 128)      0

dense2 (Dense)              (None, 1, 1, 1, 64)       8256

dropout2 (Dropout)          (None, 1, 1, 1, 64)       0

flatten (Flatten)           (None, 64)                0

Sigmoid (Dense)             (None, 1)                 65

=================================================================
Total params: 1,394,465
Trainable params: 1,394,465
Non-trainable params: 0
_________________________________________________________________
```

Listing B.2: Multilabel model 3D CNN architecture

```
_________________________________________________________________
Layer (type)                Output Shape              Param #
=================================================================
input (InputLayer)          [(None, 128, 128, 32, 1)  0
                            ]

conv1 (Conv3D)              (None, 128, 128, 32, 16)  448

maxPool1 (MaxPooling3D)     (None, 64, 64, 16, 16)    0

batch_normalization (BatchN  (None, 64, 64, 16, 16)   64
ormalization)

conv2 (Conv3D)              (None, 32, 32, 8, 32)     13856

maxPool2 (MaxPooling3D)     (None, 16, 16, 4, 32)     0

batch_normalization_1 (Batc  (None, 16, 16, 4, 32)    128
hNormalization)

conv3 (Conv3D)              (None, 16, 16, 4, 64)     55360

conv4 (Conv3D)              (None, 8, 8, 2, 64)       110656

maxPool3 (MaxPooling3D)     (None, 4, 4, 1, 64)       0

batch_normalization_2 (Batc  (None, 4, 4, 1, 64)      256
hNormalization)

conv5 (Conv3D)              (None, 4, 4, 1, 96)       165984
```

```
conv6 (Conv3D)               (None, 2, 2, 1, 96)       248928

maxPool4 (MaxPooling3D)      (None, 1, 1, 1, 96)       0

batch_normalization_3 (Batc  (None, 1, 1, 1, 96)       384
hNormalization)

conv7 (Conv3D)               (None, 1, 1, 1, 128)      331904

conv8 (Conv3D)               (None, 1, 1, 1, 128)      442496

maxPool5 (MaxPooling3D)      (None, 1, 1, 1, 128)      0

batch_normalization_4 (Batc  (None, 1, 1, 1, 128)      512
hNormalization)

dropout1 (Dropout)           (None, 1, 1, 1, 128)      0

dense1 (Dense)               (None, 1, 1, 1, 128)      16512

dropout2 (Dropout)           (None, 1, 1, 1, 128)      0

dense2 (Dense)               (None, 1, 1, 1, 64)       8256

dropout3 (Dropout)           (None, 1, 1, 1, 64)       0

flatten (Flatten)            (None, 64)                0

Softmax (Dense)              (None, 3)                 195

=================================================================
Total params: 1,395,939
Trainable params: 1,395,267
Non-trainable params: 672
_________________________________________________________________
```

Appendix C

Additional results

C.1 Tabular Data Model Results per Pipeline

Table C.1: Binary classification model results, at mRS cutoff of 3

Pipeline	Model	Train split		Test split			
		Acc	AUC	Acc0	Acc1	Acc	AUC
P0	**GBC**	77%	62.5%	50%	100%	87.5%	100%
	SVM	75%	50%	0%	100%	75%	50%
	RF	74.5%	81.7%	75%	92%	87.8%	90%
	MLP	66.5%	65.8%	25%	92%	75.3%	40%
P1	**GBC**	75%	64.2%%	0%	100%	75%	85%
	SVM	75%	50%	0%	100%	75%	50%
	RF	72.5%	79.2%	75%	92%	87.8%	88%
	MLP	68%	62.5%	0%	92%	69%	40%
P2	**GBC**	75%	62.5%	0%	100%	75%	90%
	SVM	75%	51.7%	0%	100%	75%	98%
	RF	74.5%	81.7%	75%	92%	87.8%	90%
	MLP	75%	61.7%	0%	100%	75%	98%
P3	**GBC**	77%	62.5%	50%	100%	87.5%	100%
	SVM	75%	47.5%	0%	100%	75%	94%
	RF	74.5%	81.7%	75%	92%	87.8%	90%
	MLP*	75%	66.7%	0%	100%	75%	96%
P4	**GBC**	77%	62.5%	50%	100%	87.5%	100%
	SVM	75%	59.2%	0%	100%	75%	94%
	RF	74.5%	81.7%	75%	92%	87.8%	90%
	MLP	74%	64.2%	25%	100%	87.5%	71%
P5	**GBC**	77%	62.5%	50%	100%	87.5%	100%
	SVM*	75.5%	70%	25%	100%	81.3%	79%

	RF	74.5%	81.7%	75%	92%	87.8%	90%
	MLP	77.5%	46.7%	50%	100%	87.5%	100%
P6	**GBC**	75%	64.2%	0%	100%	75%	83%
	SVM	75%	51.7%	0%	100%	75%	98%
	RF	72.5%	79.2%	75%	92%	87.8%	88%
	MLP	75%	61.7%	0%	100%	75%	98%
P7	**GBC**	58%	52.1%	75	92%	87.8%	75%
	SVM	75%	50%	0%	100%	75%	50%
	RF	75%	79.2%	75%	100%	93.8%	92%
	MLP	60%	52.5%	75%	83%	81%	69%
P8	**GBC***	74.5%	64.2%	50%	100%	87.5%	98%
	SVM	75%	45.8%	25%	100%	81.3%	100%
	RF*	73%	83.3%	75%	92%	87.8%	90%
	MLP	69.5%	50%	50%	100%	87.5%	100%
P9	**GBC**	77%	62.5%	75%	92%	87.8%	100%
	SVM	77%	45%	25%	100%	81.3%	79%
	RF	75%	79.2%	75%	100%	93.8%	92%
	MLP	79%	52.5%	50%	92%	81.5%	94%

Acc0 - Accuracy at predicting mRS lower or equal to 3; Acc1 - Accuracy at predicting mRS higher than 3; Acc - Overall Accuracy; AUC - ROC-AUC; *Best performing specific algorithm (one for each of the SVMs, RFs, GBCs and MLPs)

Table C.2: Binary classification model performances, at mRS cutoff of 4

Pipeline	**Model**	**Train split**		**Test split**			
		Acc	**AUC**	**Acc0**	**Acc1**	**Acc**	**AUC**
P0	**GBC**	56%	71.7%	100%	0%	56.2%	60%
	SVM	56%	50%	100%	0%	56.2%	50%
	RF	62%	78.3%	89%	29%	62.8%	73%
	MLP	50.5%	53.3%	100%	29%	68.9%	70%
P1	**GBC**	56%	73.3%	100%	0%	56.2%	63%
	SVM	56%	50%	100%	0%	56.2%	50%
	RF*	64.5%	78.3%	89%	29%	62.8%	73%
	MLP	50.5%	55.8%	100%	14%	62.4%	73%
P2	**GBC**	56%	71.7%	100%	0%	56.2%	60%
	SVM	60.5%	56.7%	100%	14%	62.4%	71%
	RF	62%	78.3%	89%	29%	62.8%	73%
	MLP	65%	65%	100%	29%	68.9%	57%
P3	**GBC**	56%	71.7%	100%	0%	56.2%	60%
	SVM	63%	55.8%	100%	0%	56.2%	59%

	RF	62%	78.3%	89%	29%	62.8%	73%
	MLP	56%	57.5%	100%	0%	56.2%	41%
P4	**GBC**	56%	71.7%	100%	0%	56.2%	60%
	SVM*	52%	60%	100%	43%	75.1%	75%
	RF	62%	78.3%	89%	29%	62.8%	73%
	MLP	56.5%	57.5%	100%	57%	81.2%	86%
P5	**GBC**	56%	71.7%	100%	0%	56.2%	60%
	SVM	60%	54.2%	89%	29%	62.8%	78%
	RF	62%	78.3%	89%	29%	62.8%	73%
	MLP	56%	52.5%	100%	0%	56.2%	49%
P6	**GBC**	56%	73.3%	100%	0%	56.2%	63%
	SVM	60.5%	58.3%	100%	14%	62.4%	70%
	RF	64.5%	78.3%	89%	29%	62.8%	73%
	MLP	60.5%	66.7%	100%	29%	68.9%	60%
P7	**GBC**	71%	68.3%	78%	29%	56.6%	60%
	SVM	56%	50%	100%	0%	56.2%	50%
	RF	58%	72.1%	89%	43%	68.9%	60%
	MLP	50.5%	53.3%	89%	43%	68.9%	73%
P8	**GBC***	69%	72.5%	89%	43%	68.9%	67%
	SVM	60.5%	55%	100%	29%	68.9%	75%
	RF	60%	65%	78%	43%	62.7%	73%
	MLP	63%	63.3%	100%	29%	68.9%	56%
P9	**GBC**	68.5%	70%	78%	29%	56.6%	60%
	SVM	58%	59.2%	100%	29%	68.9%	73%
	RF	56%	73.3%	78%	43%	62.7%	71%
	MLP*	65%	66.7%	100%	29%	68.9%	59%

Acc0 - Accuracy at predicting mRS lower or equal to 4; Acc1 - Accuracy at predicting mRS higher than 4; Acc - Overall Accuracy; AUC - ROC-AUC; *Best performing specific algorithm (one for each of the SVMs, RFs, GBCs and MLPs)

Table C.3: Binary classification model results, at mRS cutoff of 5 (detects whether or not patients die within six months after the ICH)

Pipeline	**Model**	**Train split**		**Test split**			
		Acc	**AUC**	**Acc0**	**Acc1**	**Acc**	**AUC**
P0	**GBC**	67%	68.3%	100%	0%	62.5%	78%
	SVM	67%	50%	100%	0%	62.5%	50%
	RF	61%	68.3%	100%	33%	74.9%	78%
	MLP	56.5%	73.3%	80%	17%	56.4%	60%
P1	**GBC**	67%	68.3%	100%	0%	62.5%	82%

	SVM	67%	50%	100%	0%	62.5%	50%
	RF	61.5%	70%	70%	67%	68.9%	72%
	MLP	54.5%	71.7%	80%	33%	62.4%	59%
P2	**GBC**	67%	70%	100%	0%	62.5%	80%
	SVM*	67%	61.7%	100%	0%	62.5%	80%
	RF	61%	68.3%	100%	33%	74.9%	78%
	MLP	64.5%	66.7%	80%	33%	62.4%	63%
P3	**GBC**	67%	68.3%	100%	0%	62.5%	78%
	SVM	67%	61.7%	100%	0%	62.5%	68%
	RF	61%	68.3%	100%	33%	74.9%	78%
	MLP	63%	60.8%	70%	17%	50.1%	48%
P4	**GBC**	67%	68.3%	100%	0%	62.5%	80%
	SVM	67%	58.3%	100%	0%	62.5%	70%
	RF	61%	68.3%	100%	33%	74.9%	78%
	MLP	58.5%	59.2%	70%	50%	62.5%	48%
P5	**GBC**	67%	70%	100%	0%	62.5%	82%
	SVM	67%	60%	90%	17%	62.6%	57%
	RF	62.5%	68.3%	100%	33%	74.9%	78%
	MLP	62.5%	66.7%	70%	33%	56.1%	43%
P6	**GBC**	67%	73.3%	100%	0%	62.5%	77%
	SVM	67%	45.8%	100%	0%	62.5%	78%
	RF*	70%	78.8%	60%	67%	62.6%	60%
	MLP	63%	59.2%	70%	33%	56.1%	65%
P7	**GBC**	56.5%	56.7%	90%	33%	68.6%	73%
	SVM	67%	50%	100%	0%	62.5%	50%
	RF	56%	65.8%	60%	67%	62.6%	80%
	MLP*	65%	73.3%	80%	50%	68.8%	63%
P8	**GBC**	64%	71.7%	60%	50%	56.3%	72%
	SVM	63%	56.7%	90%	33%	68.6%	72%
	RF	63%	75.8%	60%	67%	62.6%	68%
	MLP	63.5%	59.2%	80%	67%	75.1%	75%
P9	**GBC***	61.5%	75%	60%	50%	56.3%	73%
	SVM	63%	59.2%	90%	33%	68.6%	70%
	RF	61.5%	75%	60%	50%	56.3%	65%
	MLP	59%	64.2%	80%	67%	75.1%	73%

Acc0 - Accuracy at predicting mRS lower or equal to 5; Acc1 - Accuracy at predicting mRS higher than 5; Acc - Overall Accuracy; AUC - ROC-AUC; *Best performing specific algorithm (one for each of the SVMs, RFs, GBCs and MLPs)

Table C.4: Multilabel classification model results

Pipeline	Model	Train split		Test split				
		Acc	AUC	Acc0	Acc1	Acc2	Acc	AUC
P0	**GBC***	53.5%	76.6%	75%	50%	50%	56.2%	63.7%
	SVM	40.5%	50%	0%	100%	0%	37.5%	50%
	RF*	59%	79.1%	100%	33%	33%	49.8%	65.7%
	MLP	47%	66.7%	50%	67%	67%	62.8%	62%
P1	**GBC**	40.5%	75.8%	0%	100%	0%	37.5%	70.7%
	SVM	40.5%	50%	0%	100%	0%	37.5%	50%
	RF	59%	79.1%	100%	33%	33%	49.8%	65.7%
	MLP	49.5%	69.2%	0%	67%	50%	43.9%	66%
P2	**GBC**	53.5%	76.6%	75%	50%	50%	56.2%	63.7%
	SVM	34.5%	41.2%	25%	50%	33%	37.4%	70.7%
	RF	59%	79.1%	100%	33%	33%	49.8%	65.7%
	MLP	41%	66.2%	25%	50%	33%	37.4%	63%
P3	**GBC**	53.5%	76.6%	75%	50%	50%	56.2%	63.7%
	SVM	37%	45.9%	25%	67%	33%	43.8%	62.7%
	RF	59%	79.1%	100%	33%	33%	49.8%	65.7%
	MLP	41.5%	58%	75%	17%	33%	37.5%	60.7%
P4	**GBC**	53.5%	76.6%	75%	50%	50%	56.2%	63.7%
	SVM	37.5%	56.4%	25%	50%	33%	37.4%	49.3%
	RF	59%	79.1%	100%	33%	33%	49.8%	65.7%
	MLP	44%	64%	75%	67%	33%	56.2%	64%
P5	**GBC**	53.5%	76.6%	75%	50%	50%	56.2%	63.7%
	SVM	46%	57.2%	25%	33%	17%	25.0%	46%
	RF	59%	79.1%	100%	33%	33%	49.8%	65.7%
	MLP	43%	64.2%	25%	33%	17%	25.0%	53.3%
P6	**GBC**	40.5%	75.6%	0%	100%	0%	37.5%	70%
	SVM	36.5%	43%	25%	50%	33%	37.4%	70.3%
	RF	59%	79.1%	100%	33%	33%	49.8%	65.7%
	MLP	41%	64.3%	25%	50%	33%	37.4%	63.3%
P7	**GBC**	51%	74.8%	75%	50%	50%	37.5%	61.7%
	SVM	36%	50%	0%	100%	0%	37.4%	50%
	RF	51%	75.6%	100%	17%	50%	49.8%	65.3%
	MLP	45.5%	62.3%	50%	67%	33%	37.4%	59%
P8	**GBC**	53.5%	72.8%	25%	50%	50%	43.8%	66.7%

	SVM*	52%	64.7%	25%	89%	67%	62.5%	64%
	RF	57.5%	77.3%	25%	67%	50%	50.1%	67%
	MLP*	44%	71.3%	50%	67%	50%	56.4%	72%
P9	**GBC**	51.5%	71.4%	50%	67%	33%	49%	66.7%
	SVM	49%	60.2%	25%	83%	33%	46.7%	73%
	RF	57%	75.3%	100%	17%	50%	49.5%	65.3%
	MLP	48.5%	63.6%	75%	33%	33%	43.3%	63%

Acc0 - Accuracy of the predictions of label 0 (mRS = 0-3); Acc1 - Accuracy of the predictions of label 1 (mRS = 4-5); Acc2 - Accuracy of the predictions of label 2 (mRS = 6); Acc - Accuracy; AUC - macro average of ROC-AUC; *Best performing specific algorithm (one for each of the SVMs, RFs, GBCs and MLPs)

C.2 Hyperparameters

Listing C.1: Hyperparameters of the binary models for mRS > 3

```
GradientBoostingClassifier(ccp_alpha=0.0, criterion='friedman_mse', init=None,
                           learning_rate=1e-06, loss='log_loss', max_depth=5,
                           max_features='log2', max_leaf_nodes=None,
                           min_impurity_decrease=0.0001, min_samples_leaf=1,
                           min_samples_split=7, min_weight_fraction_leaf=0.0,
                           n_estimators=180, n_iter_no_change=None,
                           random_state=53, subsample=0.3, tol=0.0001,
                           validation_fraction=0.1, verbose=0,
                           warm_start=False)

SVC(C=32.82, break_ties=False, cache_size=200, class_weight={}, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='auto', kernel='rbf',
    max_iter=-1, probability=True, random_state=53, shrinking=True, tol=0.001,
    verbose=False)

RandomForestClassifier(bootstrap=False, ccp_alpha=0.0, class_weight='balanced',
                       criterion='gini', max_depth=6, max_features='sqrt',
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.1, min_samples_leaf=2,
                       min_samples_split=2, min_weight_fraction_leaf=0.0,
                       n_estimators=140, n_jobs=-1, oob_score=False,
                       random_state=53, verbose=0, warm_start=False)

MLPClassifier(activation='logistic', alpha=0.5, batch_size='auto', beta_1=0.9,
              beta_2=0.999, early_stopping=False, epsilon=1e-08,
              hidden_layer_sizes=[100, 50, 50], learning_rate='constant',
              learning_rate_init=0.001, max_fun=15000, max_iter=500,
              momentum=0.9, n_iter_no_change=10, nesterovs_momentum=True,
              power_t=0.5, random_state=53, shuffle=True, solver='adam',
              tol=0.0001, validation_fraction=0.1, verbose=False,
```

```
              warm_start=False)
```

Listing C.2: Hyperparameters of the binary models for mRS > 4

```
GradientBoostingClassifier(ccp_alpha=0.0, criterion='friedman_mse', init=None,
                           learning_rate=0.001, loss='log_loss', max_depth=2,
                           max_features=1.0, max_leaf_nodes=None,
                           min_impurity_decrease=0.05, min_samples_leaf=2,
                           min_samples_split=2, min_weight_fraction_leaf=0.0,
                           n_estimators=40, n_iter_no_change=None,
                           random_state=53, subsample=0.8, tol=0.0001,
                           validation_fraction=0.1, verbose=0,
                           warm_start=False)

SVC(C=22.98, break_ties=False, cache_size=200, class_weight='balanced',
    coef0=0.0, decision_function_shape='ovr', degree=3, gamma='auto',
    kernel='rbf', max_iter=-1, probability=True, random_state=53,
    shrinking=True, tol=0.001, verbose=False)

RandomForestClassifier(bootstrap=False, ccp_alpha=0.0, class_weight={},
                       criterion='gini', max_depth=1, max_features='log2',
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.01, min_samples_leaf=6,
                       min_samples_split=10, min_weight_fraction_leaf=0.0,
                       n_estimators=240, n_jobs=-1, oob_score=False,
                       random_state=53, verbose=0, warm_start=False)

MLPClassifier(activation='logistic', alpha=0.001, batch_size='auto', beta_1=0.9,
              beta_2=0.999, early_stopping=False, epsilon=1e-08,
              hidden_layer_sizes=[100], learning_rate='adaptive',
              learning_rate_init=0.001, max_fun=15000, max_iter=500,
              momentum=0.9, n_iter_no_change=10, nesterovs_momentum=True,
              power_t=0.5, random_state=53, shuffle=True, solver='adam',
              tol=0.0001, validation_fraction=0.1, verbose=False,
              warm_start=False)
```

Listing C.3: Hyperparameters for the binary models for mRS > 5

```
GradientBoostingClassifier(ccp_alpha=0.0, criterion='friedman_mse', init=None,
                           learning_rate=0.001, loss='log_loss', max_depth=8,
                           max_features=1.0, max_leaf_nodes=None,
                           min_impurity_decrease=0.4, min_samples_leaf=5,
                           min_samples_split=5, min_weight_fraction_leaf=0.0,
                           n_estimators=270, n_iter_no_change=None,
                           random_state=58, subsample=0.55, tol=0.0001,
                           validation_fraction=0.1, verbose=0,
                           warm_start=False)

SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='auto', kernel='rbf',
```

```
    max_iter=-1, probability=True, random_state=53, shrinking=True, tol=0.001,
    verbose=False)

RandomForestClassifier(bootstrap=False, ccp_alpha=0.0,
                       class_weight='balanced_subsample', criterion='gini',
                       max_depth=6, max_features='log2', max_leaf_nodes=None,
                       max_samples=None, min_impurity_decrease=0.1,
                       min_samples_leaf=5, min_samples_split=9,
                       min_weight_fraction_leaf=0.0, n_estimators=160,
                       n_jobs=-1, oob_score=False, random_state=58, verbose=0,
                       warm_start=False)

MLPClassifier(activation='relu', alpha=0.005, batch_size='auto', beta_1=0.9,
              beta_2=0.999, early_stopping=False, epsilon=1e-08,
              hidden_layer_sizes=[50, 50, 100], learning_rate='adaptive',
              learning_rate_init=0.001, max_fun=15000, max_iter=500,
              momentum=0.9, n_iter_no_change=10, nesterovs_momentum=True,
              power_t=0.5, random_state=53, shuffle=True, solver='adam',
              tol=0.0001, validation_fraction=0.1, verbose=False,
              warm_start=False)
```

Listing C.4: Hyperparameters for the multilabel model

```
GradientBoostingClassifier(ccp_alpha=0.0, criterion='friedman_mse', init=None,
                           learning_rate=0.4, loss='log_loss', max_depth=9,
                           max_features='sqrt', max_leaf_nodes=None,
                           min_impurity_decrease=0.001, min_samples_leaf=3,
                           min_samples_split=7, min_weight_fraction_leaf=0.0,
                           n_estimators=270, n_iter_no_change=None,
                           random_state=53, subsample=0.85, tol=0.0001,
                           validation_fraction=0.1, verbose=0,
                           warm_start=False)

SVC(C=1.0, break_ties=False, cache_size=200, class_weight=None, coef0=0.0,
    decision_function_shape='ovr', degree=3, gamma='auto', kernel='rbf',
    max_iter=-1, probability=True, random_state=54, shrinking=True, tol=0.001,
    verbose=False)

RandomForestClassifier(bootstrap=False, ccp_alpha=0.0, class_weight='balanced',
                       criterion='entropy', max_depth=1, max_features='log2',
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.3, min_samples_leaf=2,
                       min_samples_split=5, min_weight_fraction_leaf=0.0,
                       n_estimators=150, n_jobs=-1, oob_score=False,
                       random_state=53, verbose=0, warm_start=False)

MLPClassifier(activation='logistic', alpha=0.005, batch_size='auto', beta_1=0.9,
              beta_2=0.999, early_stopping=False, epsilon=1e-08,
              hidden_layer_sizes=[50, 50, 50], learning_rate='invscaling',
              learning_rate_init=0.001, max_fun=15000, max_iter=500,
```

```
momentum=0.9, n_iter_no_change=10, nesterovs_momentum=True,
power_t=0.5, random_state=54, shuffle=True, solver='adam',
tol=0.0001, validation_fraction=0.1, verbose=False,
warm_start=False)
```