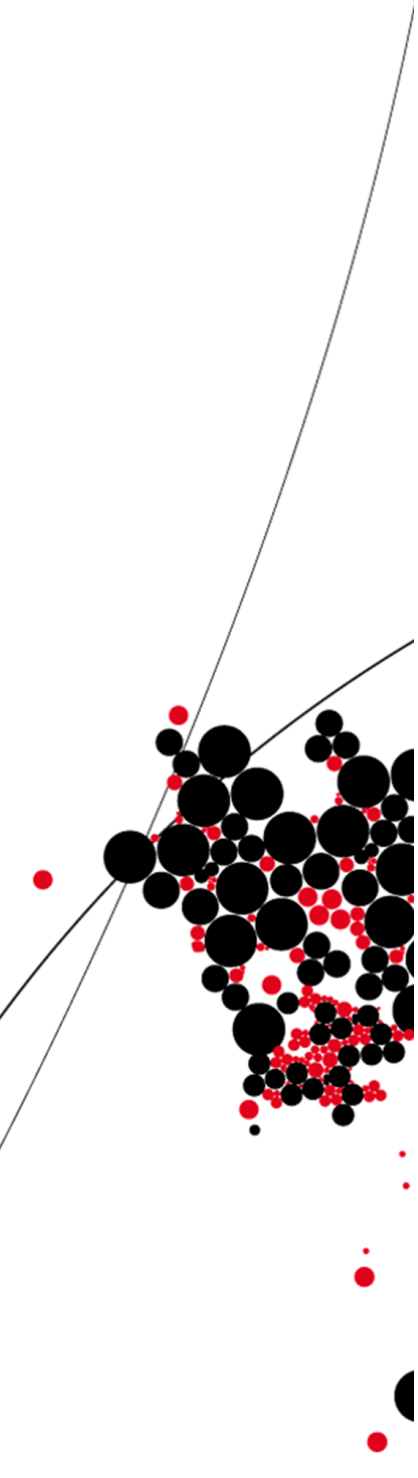
UNIVERSITY OF TWENTE.

Faculty of Electrical Engineering, Mathematics & Computer Science

30-Days Post-Operative Mortality Prediction of Elderly Hip Fracture Patients

Berk Yenidogan

Master's in Computer Science
Specialization: Data Science and Technology
M.Sc. Thesis
July 2020



External Supervisors:
dr. Han Hegeman
ing. Jeroen Geerdink

Ziekenhuis Groep Twente (ZGT)
Geerdinksweg 141
7555 DL Hengelo
The Netherlands

Supervisors:
dr. *ir.* Maurice van Keulen
dr. Jasper Reenalda
Shreyasi Pathak MSc

Data Management and Biometrics
Faculty of Electrical Engineering,
Mathematics and Computer Science
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

ABSTRACT

Hip fractures on the elderly are a major health care problem in society. In the clinic, it is important to identify high-risk patients to guide the decision making with respect to the treatment of the patient. This study presents a prediction model for 30-days mortality of elderly hip fracture patients by following a multimodal machine learning approach. This approach fuses the image modality with the structured modality for the prediction task. At the same time, it also addresses the problems related to the class imbalanced dataset and the high number of missing values. The early fusion model, developed in this study, first extracts features from the chest and hip x-ray images by the use of convolutional neural networks. Subsequently, it combines extracted features with structured modality and feeds into a Random Forest Classifier to finalize the prediction. The proposed model outperforms a replicated version of Almelo Hip Fracture Score (AHFS-a) with an AUC score of 0.742 vs 0.706. Finally, by the analysis of feature importances, this study also demonstrates that chest x-ray images contain important signs related to 30-days mortality of elderly hip fracture patients.

CONTENTS

1 INTRODUCTION

With changing socioeconomic conditions, the life expectancy of humans increases. Apart from other consequences, this also results in a higher number of elderly showing up in the emergency room with an acute hip fracture. According to [1], the estimated number of hip fractures in 1990 throughout the world was 1.26 million, this number is expected to reach 2.6 million and 4.5 million in 2025 and 2050 respectively. These elderly hip fracture patients usually have comorbidities. And due to their frailty and comorbidities, some of them are considered to be at the high-risk group for mortality. It was reported that 30-days mortality rate of hip fracture patients can be up to 13.3% [2]. With respect to the scope of this study, only the 30-days mortality rate of elderly patients is of interest which is 8% in the dataset of this study. This rate for early mortality is in fact requires high attention on how it is handled.

In the clinic, it is crucial to identify the high-risk patients to consider different treatment pathways and take surgical decisions. ZGT (Ziekenhuis Groep Twente), the hospital collaborating on this study, aims to improve the quality of care, reduce the costs involved, and inform patients and their relatives more thoroughly by identifying high-risk patients. For this reason, a prediction model is necessary which can determine the high-risk patients. However, the prediction models developed so far was limited to conventional techniques and did not make use of recent technologies.

There were various studies which proposed risk/prediction models regarding the early mortality of hip fracture patients after surgery [3–8]. All of these studies used a conventional technique, namely logistic regression, to develop the 30-days mortality prediction/risk model. The general set of variables that turned out to have a significant impact on 30-days mortality are age, gender, fracture type, pre-fracture residence, pre-fracture mobility, ASA [1] score, signs of malnutrition, comorbidities, cognitive problems. The performance of the mentioned models are evaluated with the AUC (Area Under the ROC curve) score and ranged from AUC of 0.70 to 0.82.

The technical motivation of this study is to employ different approaches for predicting 30-days mortality of hip fracture patients. These approaches include several categories: class imbalance handling, use of different machine learning algorithms, make use of records that contained missing values for some variables, feature extraction from x-ray images, and multimodal learning.

As the first category, a method for class imbalance handling is required. As mentioned earlier, early mortality(positive sample) rate is 8% which makes the distribution of the classes imbalanced. Although this mortality rate is representative of the real life situation, it still challenges a classifier to classify a sample as positive. Especially, when the dataset is relatively small. This study includes only 193 positive samples which is indeed a small sample size.

Furthermore, it is aimed to make use of a wider set of variables than previous studies. How-

[1] American Society of Anaesthesiologists

ever, these variables started being collected at different times in the study period. Therefore, a significant amount of patients have lots of missing variables. In order to include these patients in the study, their missing values should be imputed.

Due to the fact that collecting structured data is costly and requires a lot efforts during the treatment process, it is aimed to extract features from chest and hip x-ray imaging regarding the early mortality of the patient. The motivation behind this is that most of the patients have chest and hip x-rays and extracted features from these imaging data could be used as comorbidity findings alongside the structurally collected data. In this way, a multimodal methodology is aimed to be achieved by fusing image modality and structured modality to predict the early mortality of hip fracture patients.

Moreover, instead of using only logistic regression, other machine learning techniques and deep learning techniques wanted to be used in the classification phase to find if they can perform better in this task.

1.1 Problem Statement

The main goal of this project is to predict if an elderly patient will survive or decease in 30 days after a hip fracture surgery by processing pre-operative variables. It can be acknowledged as a binary classification problem.

1.2 Research Questions

There are 2 main research questions/problems regarding this study with multiple sub-questions.

1. To what extent, one can predict 30-days mortality of the elderly hip fracture patients after surgery using machine learning with pre-operative variables?
   (a) With respect to the class imbalanced dataset, to what extent, class imbalance handling techniques are useful to preprocess the data for classification?
   (b) Due to the high amount of missing values, to what extent, the missing value imputation techniques are suitable in predicting 30-days mortality of the elderly hip fracture patients?
   (c) Which machine learning algorithm performs best in the classification task?
2. As the literature suggests that multimodal machine learning showed good results in the medical domain, it is important to question whether different modalities would contribute to predicting 30-days mortality of elderly hip fracture patients.
   (a) To what extent, one can predict 30-days mortality by using chest and hip X-ray images?
   (b) Different variable groups have difficulties in the collection and extraction phases and might be costly as well. However, if it was possible to extract these variables from x-ray images, it would be less costly and easier. To what extent, extracted features from x-ray images can be used to replace structurally collected variables?
   (c) What is the most suitable way to fuse image modality and structured modality when predicting 30-days mortality of the elderly hip fracture patients?

(d) To what extent, multimodal fusion improves the prediction on 30-days mortality when compared to prediction with only structured modality?

1.3 Research Method

This research was conducted in collaboration with ZGT (Hospital Group Twente). The dataset used in this study included 2404 patients with an early mortality rate of 8%. The study period is from 04-2008 to 01-2020. The author used several class imbalance techniques, missing value imputation techniques by means of regression, machine learning algorithms including traditional machine learning and deep learning, furthermore, different ways of multimodal(image modality and structured modality) fusion to answer the research questions. Evaluation of the prediction models was done with AUC scores as it is the best metric to use on imbalanced datasets.

1.4 Outline

Regarding the structure of this thesis, Chapter 2 introduces the background information about the thesis. Chapter 3 presents the related work about predicting 30-days mortality for elderly hip fracture patients and multimodal practices in the medical domain. Chapter 4 introduces the terminology used by the author. Chapter 5 explains the author's methodology. Chapter 6 describes the dataset preparation and variables used in the study. Chapter 7 focuses on the multimodal perspective of the methodology. Chapter 8 presents experimental settings and results. Discussion about the experiments takes place in Chapter 9 including the limitations of the study and future work. The author concludes in Chapter 10.

2 BACKGROUND

In this section, the author introduces the background which is useful to have a better understanding of this study.

2.1 Hip Fractures

Hip fractures typically happen when older people stumble or fall. It was observed that women tend to fracture their hip more often than men and it is strongly related to the age of the person [9]. It was reported that in 15% of the cases, the fracture is undisplaced and radiographic changes may be minimal, in 1% fractures may not be visible on radiographs and further investigation is needed, apart from those they can be confirmed by the plain radiographs of the hip [9]. Hip fractures can be classified as intracapsular and extracapsular, but these might be further subdivided based on the level of fracture or existence of displacement and comminution [9]. Figure 2.1 gives a clear overview of the classification of fractures. The type of fracture is one of the determinant factors for which surgery type should be performed and thereby it has also considerable effect on the patient's postoperative recovery process [10].

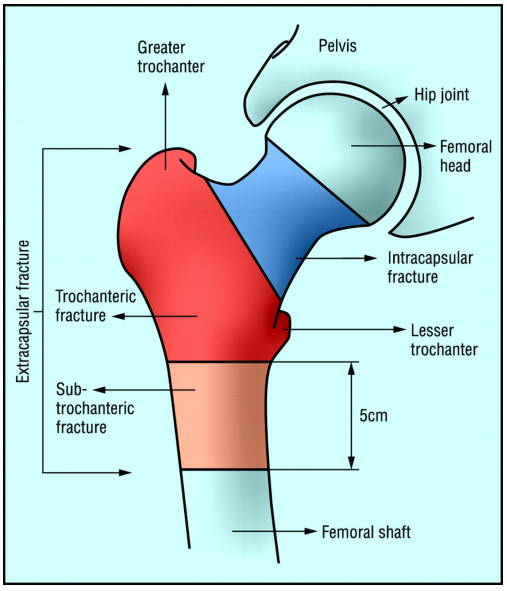


Figure 2.1: Classification of Hip Fractures. Fractures in the blue area are intracapsular and those in the red and orange areas are extracapsular. (taken from [9])

Most hip fractures are treated with surgery due to the long hospital stay and bad results of conservative approaches [9]. In general, the treatment of a hip fracture might involve multiple disciplines such as ambulance service, the emergency room, radiology, anesthetics, orthopedic

surgery, trauma surgery, medicine, and rehabilitation [9]. Depending on the health care system, an orthopedic surgeon or the trauma surgeon makes the treatment plan. In the Netherlands, if a patient is presented at the emergency department with a hip fracture and has former complaints of the hip joint (arthrosis), the trauma surgeon asks the orthopedic surgeon if the patient needs a total hip arthroplasty. As this study is conducted in collaboration with Ziekenhuisgroep Twente (ZGT, Hospital Group Twente), the author will present the general treatment procedure of this Hospital for hip fracture patients. It is called an integrated orthogeriatric treatment model and executed by the Geriatric Traumatology Center (CvGT) at ZGT. However, not all the hip fracture patients are treated in CvGT, due to their status, healthier patients are treated by orthopedics surgeons. This study does not include these patients, therefore CvGT is the main focus. Figure 2.2, demonstrates the flowchart of the model in CvGT. Folbert et al. [11] describe an integrated orthogeriatric treatment model as follows:

> *The aim of the introduction of the integrated orthogeriatric treatment model was to prevent complications and loss of function by implementing a proactive approach by means of early geriatric co-management from admission to the emergency department (ED) by following clinical pathways and implementing a multidisciplinary approach. A nurse practitioner or physician's assistant specialized in trauma surgery made daily visits to the ward under the supervision of a trauma surgeon and geriatrician. For purposes of fall prevention, chronic medication was evaluated, osteoporosis status was investigated, and treatment was started if necessary. A multidisciplinary meeting was held twice a week to discuss the treatment goals, patient progress, and discharge plan. The aim was to have the patients ready for discharge within 5–7 days. Surgery follow-up appointments involved patients attending a multidisciplinary outpatient clinic where they visited a trauma surgeon, physiotherapist, and nurse specialized in osteoporosis ("osteo-physio-trauma outpatient clinic").*

The journey of a patient from the perspective of data collection passes through multiple stages. Firstly, when a patient arrives in the emergency room, standardized imaging and lab tests are ordered. Standardized imaging includes pelvis and chest x-rays on the AP (Anterior-Posterior) view. However, in some cases, doctors might request additional images of other body parts or other views of the same parts. Subsequently, physical examination, recording of vital signals, and electrocardiography take place. If possible, the patient is questioned about their complaint and medical history. At the same time, the medical history of the patient is also collected by means of letters registered in the hospital. After these, based on the collected data, a conclusion and suggestion take place. From that point, the patient's medical admission finishes, and they get transferred to the clinic. All elderly patients (70 or older) which creates the study group of this thesis, are visited by a geriatrician. If there is cardiac risk involved, a cardiologist is consulted. In such cases, an ultrasound study of the heart takes place. Structured survey data such as nutrition, mobility, activities of daily living, cognitive problems, comorbidities, living situations are all collected in the clinic.

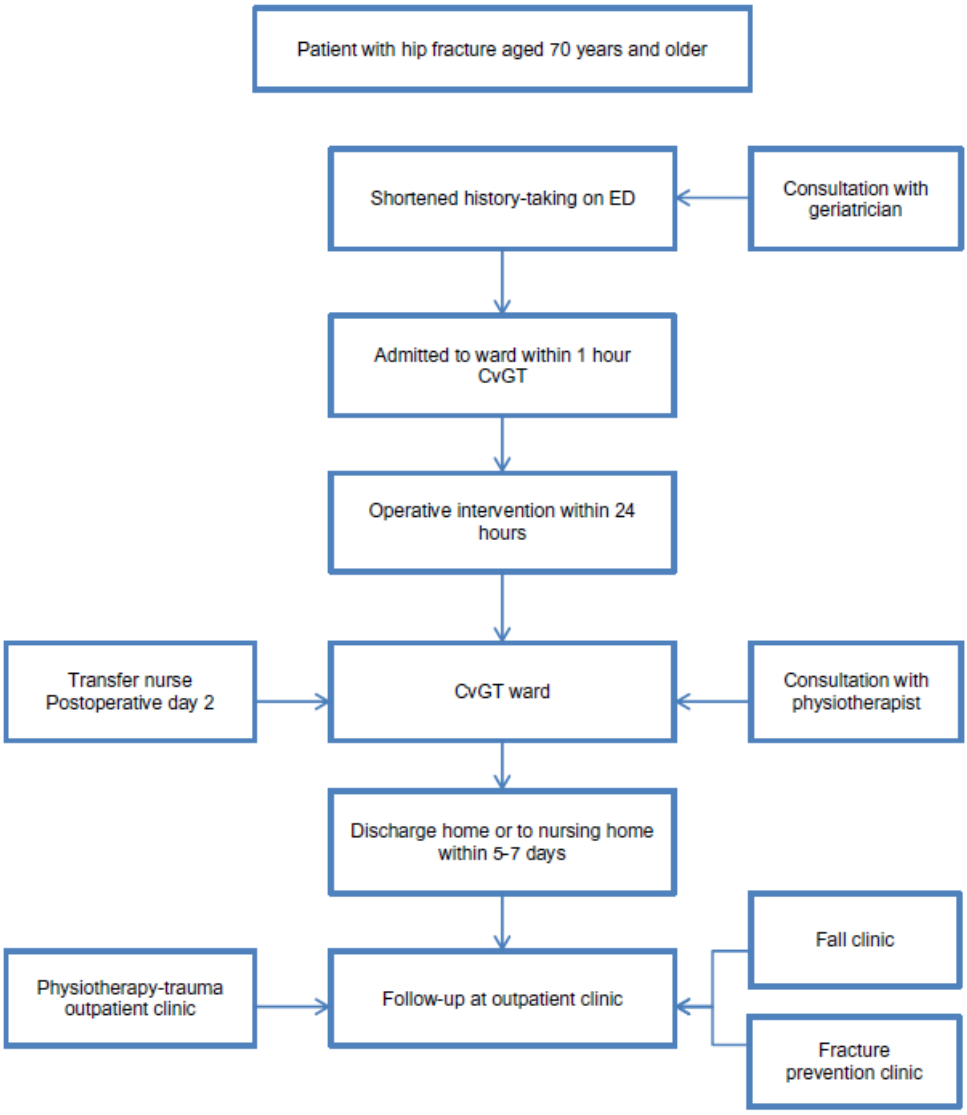


Figure 2.2: Integrated orthogeriatric treatment model (taken from [11])

2.2 Machine Learning

Machine learning is programming computers to identify patterns in data by using algorithms and statistical models, in order to accomplish various tasks. It is a subset of Artificial Intelligence. Two main branches of Machine Learning are Supervised Learning and Unsupervised Learning.

2.2.1 Supervised Learning

In Supervised Learning, the goal is to predict an outcome or a phenomenon by the use of relevant features related to that particular case. An example would be to predict if a customer will buy a specific product using the customer's demographics information. In order to perform such a prediction, a model has to be developed first. Simply, the basic idea is to map feature set X to output label Y by using a method f, $Y = f(X)$. Once the model is fed with input tokens

and output labels, it will optimize its mapping function and this process is called supervised learning. In this example, customer demographics information are the feature set X and the buying decision is the outcome label Y.

After the learning(training) is done, a model has to be validated that it gives accurate predictions. This validation is done simply by predicting the outcome of cases and comparing these predictions with the known ground truth. At this point, one would try multiple models to get the best performance during validation. This process is called hyperparameter optimization. After finding the best model, testing has to be performed. This process is the same as validation but the only difference is the predicted dataset. More precisely, it has to be tested with unseen data by the model to find its actual accuracy. The author uses the word actual here because the performance metrics obtained in the validation phase might be biased. The reasoning for this comes from the fact that the model which scores best on a particular validation set is chosen but this model might perform differently on another dataset. After achieving desired results on testing, the model is ready for deployment. It can now predict the buying decisions of customers based on their demographics information.

2.2.1.1 Decision Tree

A decision tree aims to separate a dataset in small subsets which are created based on decision- and leaf nodes. Decision nodes representing a test on an attribute and are followed by 2 or more branches where each stands for a certain value of a feature. Whereas a leaf node is representing the decision on the unknown value (label). The result is a tree structure that can be followed according to the attributes/features and end up in a certain label that corresponds to the target prediction. The selection of attributes for decision rules are based on impurity measures such as Gini and Entropy. Most popular algorithms for decision trees are ID3, C4.5, C5.0 and CART [12–14]

2.2.1.2 Random Forest

Random Forest is an ensemble algorithm. Simply it trains multiple decision trees with different parts of the data. Then it takes the votes of those decision trees for classification and it decides for the class of the test object based on those votes. [15]

2.2.1.3 AdaBoost Classifier

AdaBoost is an ensemble learning method. First, it fits the data on some estimators(e.g. decision trees) then, it tries to fit the data on more estimators but by giving higher importance to the misclassified subjects. Therefore, the next generation estimators focus on more difficult cases in terms of classification [16].

2.2.1.4 XGBClassifier (eXtreme Gradient Boosting Classifier

Similar to AdaBoost, XGBClassifier is an ensemble learning method that implements boosted trees. Main motivation of this library is to allow scalability and maximize system optimization. [17]

2.2.1.5 Support Vector Machines

During the training, Support Vector Machines(SVMs) learn a decision boundary that maximizes the margin between different classes. This learning is done through the use of Lagrange multipliers. Learning the parameters for the model in fact corresponds to a convex optimization problem where any local solution is the global optimum. SVMs do not provide posterior probabilities. However, they use different kernel functions (e.g. linear, polynomial, radial basis function) to transform the data to higher-dimensional spaces to make it separable [18].

2.2.1.6 Logistic Regression

Logistic Regression calculates the conditional probability of an object belonging to a particular class by using a sigmoid cost function. The formulation for the model is as follows where Y is the target variable, X denotes the vector for features and w denotes the vector for weights:

$$Pr(Y|X = x) = \sigma(wx) = \frac{1}{1 + e^{-wx}} \tag{2.1}$$

There are multiple algorithms for fitting the data to a logistic regression model such as SAG, SAGA, Liblinear [19–21].

2.2.2 Unsupervised Learning

In Unsupervised Learning, the goal is to identify unknown patterns without the use of any labeled data. Principal component analysis, auto-encoders are examples of this learning technique in dimensionality reduction of the feature set. Cluster analysis is also a very common usage of unsupervised learning to categorize units with minimum human instructions such as the number of categories. In this study, unsupervised learning is applied by the use of auto-encoders, therefore more background information on this will be in section 2.3.

2.3 Deep Learning

Deep learning is actually a subfield of machine learning where artificial neural networks are employed for various tasks such as natural language processing, computer vision, speech recognition, drug discovery and genomics [22]. Training such networks, need high computational powers and might take a long time. But during the last decade, thanks to advancements in computers, especially in graphics cards, computational power has increased and Deep Neural Networks has become more usable and popular. These networks have multiple layers, that is why this type of learning is called deep. It is inspired by the form of the human brain. Deep learning methods are rather black-box compared to traditional machine learning techniques mentioned 2.2.1, thus their biggest disadvantage is the lack of interpretability. Deep Learning could be used both for Supervised and Unsupervised learning.

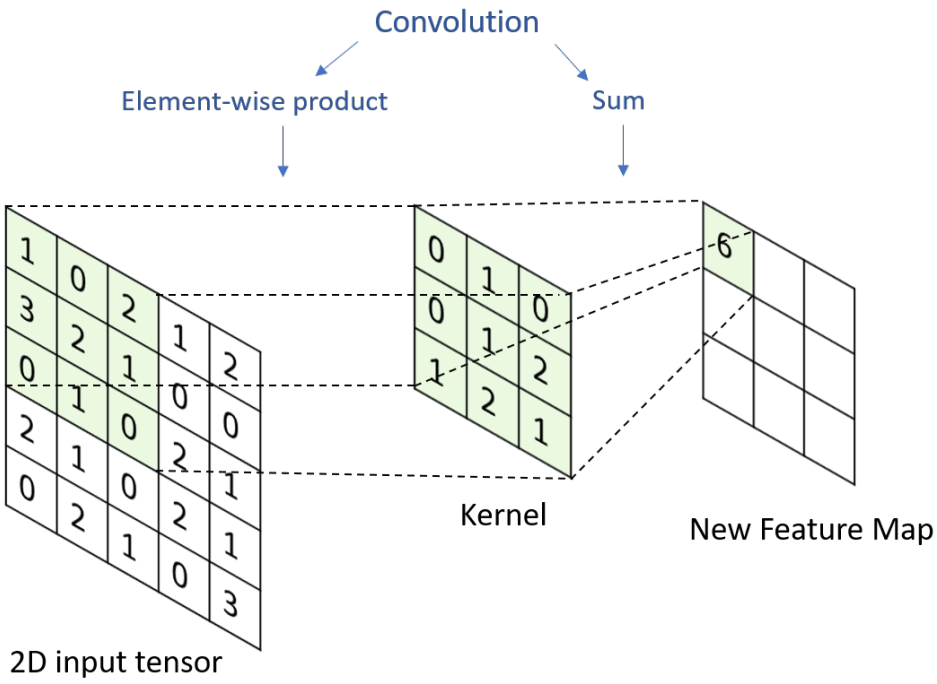


Figure 2.3: Illustration of a kernel applied on a 2D input in a Convolutional layer

2.3.1 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) is one kind of Deep Learning method. It is used on data with a grid pattern, such as images. They are designed to learn spatial-hierarchies of features automatically, in other words, they are planned to extract features from images automatically [23]. Characteristic building blocks of CNNs are convolutional layers, pooling layers, and fully connected layers. Convolutional layers and pooling layers constitute convolutional blocks, which are used in feature extraction. On the other hand, fully connected parts of these networks are used for mapping extracted features to the target output. Convolutional layers are the most important layer of CNNs as they do convolution operation which is a linear mathematical operation. There are a bunch of kernels in a convolutional layer, and all kernels are applied to every position of the image to find the relevant features. An illustration of a kernel applied on a 2D grid input tensor can be seen in figure 2.3. Right after this, a pooling layer is used to summarize the process and get the most significant outcome. To have a deeper understanding of Convolutional Neural Networks, the reader is advised to read pages 1-9 of [23].

2.3.2 Transfer Learning

Traditional Machine Learning methods works under the assumption of data used for training and testing are drawn from the same space [24]. However, this is not always possible, especially when doing Deep Learning due to an insufficient amount of data as training a Deep Neural Network requires large datasets such as ImageNet which is consisted of 14 million images from 20000 subcategories. Transfer Learning or Knowledge Transfer emerged due to the lack of data in a domain of interest. The basic idea of Transfer Learning is to use the knowledge gained in a particular domain on a different one. The inspiration comes from the fact that generic features do not change on datasets from different domains, therefore neural networks that are pre-trained on large datasets come as a convenient and effective solution for extracting features

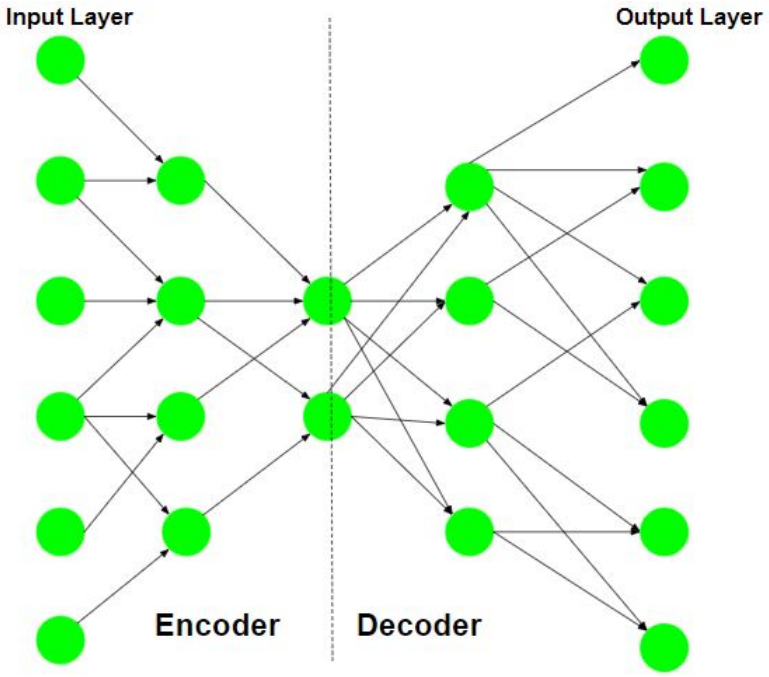


Figure 2.4: Example Auto-encoder Architecture (taken from [30])

on rather small datasets. Although these pre-trained networks perform well on famous datasets, they still have to be adjusted in order to be usable on the dataset at hand. There are two ways of adjusting this pre-trained network. The first option is a fixed feature extraction method, where the convolutional part of the network is kept the same and the fully connected layer (classifier) part is removed. If there is no replacement on the classifier part, it is possible to use this option for unsupervised learning only to extract features, but this approach is not advised when working on very specific domains such as medical imaging. The second option is a fine-tuning method, where alongside fully connected layers, weights in the convolutional layers are also retrained. This retraining on the convolutional part could be partly where only the deeper layers are retrained as they carry more domain-specific information, or it could be completed, in a way that, all the weights in the convolutional part are retrained [25]. Some examples of the most popular pre-trained networks are DenseNet, VGG, Inception [26–28].

2.3.3 Auto-encoders

Auto-encoders are one kind of unsupervised learning technique that makes use of artificial neural networks. The idea is to decrease the dimensionality of the feature space, in other words, compress the data. The goal of the network is to represent the original input in fewer dimensions. During the training part, original features are used both for input features and output labels. And during the prediction, the output is the reconstruction of the original input from the lowered dimensions. Auto-encoders consists of an encoding part and decoding part. In figure 2.4, an example architecture of an auto-encoder can be seen. Encoding and decoding parts are symmetrical with respect to the center where the encoded representation is located. Using Auto-encoders for image data is also possible however, convolutional auto-encoders are employed for this task. In Convolutional Auto-encoders, fully connected layers are replaced by convolutional blocks. As convolutional layers do not ignore the 2D form of images which makes them more suitable and practical [29].

2.4 Multimodal Machine Learning

Humans experience things in different ways with different senses, such as seeing, hearing, tasting, touching, and smelling. A modality implies how something happened or experienced, and a research problem is identified as multimodal when it consists of more than one modality in its nature [31].The multimodal machine learning concept has the goal to develop models that can process and associate information from different modalities.

Recently in 2019, Baltrušaitis et al. introduced a new taxonomy to identify challenges of multimodal machine learning [31]. These challenges are representation, translation, alignment, fusion, and co-learning. According to their definition, representation is about "how to represent and summarize multimodal data in a way that exploits the complementarity and redundancy", translation "addresses how to translate (map) data from one modality to another", alignment concerns "to identify the direct relations between (sub)elements from two or more different modalities", fusion focuses to "join information from two or more modalities to perform a prediction" and co-learning aims "to transfer knowledge between modalities, their representation, and their predictive models" [31].

Regarding 30-days mortality prediction, the relevant challenges are representation and fusion. Therefore, the author will focus more on these.

2.4.1 Representation with Deep Neural Networks

In [31], it was stated that representing data in a meaningful way is of importance in multimodal problems. There were two proposed categories for representation, namely joint and coordinated representations [31]. Former combines each modalities' signals in the same representation space, whereas latter processes each modality separately and enforce similarity constraints to put them on a "coordinated space" [31].

Joint representations could be done in three ways, by using, neural networks, probabilistic graphical models, or sequential representations.

In deep neural networks, using the last layer or predecessor of the last layer is popular for data representation as they carry more precise and relevant information [32]. In [31], it was observed that in order to build a joint multimodal representation in neural networks, the first modalities have to have their individual neural layers, then a hidden layer should bring them to joint space. Due to the fact that neural networks require a high amount of labeled data, the pre-training for representations could be unsupervised as autoencoders or supervised from a related domain [31].

2.4.2 Fusion with Model Agnostic Approaches

Multimodal fusion refers to combining information from multiple modalities with the objective of predicting an event, i.e. classification of numerical prediction [31]. It is suggested that the fusion of multiple modalities leads to more robust predictions, captures complementary information, and can operate when a modality is missing [31]. According to Baltrušaitis et al., fusion is classified into two approaches, namely, model-agnostic approaches and model-based approaches [31]. The former does not depend on a particular machine learning algorithm [31]. By

contrast, the latter is in the construction of algorithm [31].

Model-agnostic approaches have three levels of fusion, namely, early fusion, late fusion, and their combination hybrid fusion [33]. Early fusion refers to the combination of modalities right after the feature extraction, late fusion refers to the association of decisions coming from modalities [33]. Hybrid fusion is done by combining the outputs of early and late fusion.

Model-based approaches can be split into three, Multiple Kernel Learning (MKL) methods, graphical models, and neural networks [33].

2.5 Imbalanced Learning

Imbalanced learning is the concept where learning(machine learning) is done with imbalanced data in terms of the classes. Generally, most of the supervised learning algorithms work under the assumption that classes in a dataset are evenly distributed. However, this is not always the case. This uneven distribution of classes is called a class imbalance. Class imbalance can be in many different forms, such as 1:10, 1:100, 1:1000. On the other hand, class imbalance may also exist in datasets with multi-classes. As this study is concerned with a binary classification problem, the author will focus only on two-class imbalanced learning problems.

When working with imbalanced datasets, one crucial thing to realize is that using a single evaluation metric such as accuracy or error rate is not sufficient. Therefore, during the evaluation, one should employ other performance metrics such as precision, specificity, ROC curve [34]. Performance evaluation metrics used in this study will be described in section 2.6.

There are various ways to deal with the class imbalance problem. The author will now describe some of the most popular methods.

2.5.1 Random Oversampling and Undersampling

Random Oversampling is a technique where randomly selected examples from the minority class are appended to the dataset until the desired ratio between classes is achieved. With the same logic, Random Undersampling is selecting random examples from the majority class and removing them from the dataset until the desired ratio is achieved.

2.5.2 Informed Undersampling with K-Nearest Neighbor Classifier

This way of dealing with class imbalance problems makes use of the K-Nearest Neighbor (KNN) classifier when selecting samples to remove from the majority class. Zhang and Mani proposed and evaluated a few methods using this approach [35]. NearMiss-2 was the best performing one. The key idea of this method is to select majority class examples that are closer to all minority class samples. It is done by selecting and removing the majority class examples with the smallest average distance to K farthest minority class examples.

2.5.3 The Synthetic Minority Oversampling Technique (SMOTE)

SMOTE is an approach of oversampling. However, SMOTE does not duplicate minority class examples. Instead, it generates synthetic data to achieve a balanced class distribution. The generation process is as follows. Firstly, a minority class example is selected, let x_i denote this example's features, then K nearest minority class examples to x_i are collected, among these neighbors, one of them is randomly selected, let x_j denote the randomly selected neighbor example's features. Finally, the new synthetic data is generated by,

$$x_{new} = x_i + (x_i - x_j) \times \alpha \tag{2.2}$$

where alpha is a random number between [0,1]. This process is repeated for each minority class example. For better understanding of this algorithm, the reader is advised to refer to section 4.2 of [36].

2.5.4 Borderline-SMOTE

Borderline-SMOTE algorithm is another oversampling technique. The motivation of this algorithm is the fact that minority class examples that are closer to the borderline (i.e. majority class examples), are harder to classify correctly, therefore giving higher importance to these examples improve the performance of the oversampling. It works in the same way as SMOTE but it generates synthetic examples based on the minority class examples which are on the borderline [37]. There are two variations of this approach, namely, Borderline-SMOTE-1 and Borderline-SMOTE-2. The main difference of the second is that it generates synthetic data also from the majority class neighbors. For a detailed explanation of the algorithm, the reader is advised to refer to section 3 of [37].

2.5.5 ADASYN: Adaptive synthetic sampling

ADASYN is an oversampling approach where synthetic examples are generated based on minority class examples which are harder to learn. It is similar to Borderline-SMOTE as they are both considered as adaptive and care more about difficult examples. In the ADASYN algorithm, a density distribution, regarding their level of difficulty in learning, is used to decide the number of synthetic examples to be generated based on each minority class example. For a complete explanation of this algorithm, the reader is advised to refer to ADASYN Algorithm section of [38].

2.5.6 Adjusting Class Weights

One approach to deal with class imbalance is to adjust the class weights during the learning. Idea is to give more importance to minority class during the calculation of loss function. Machine learning techniques used in this paper, Decision Tree, Logistic Regression, Neural Networks, Support Vector Machines, support this technique.

2.6 Evaluation Metrics

In this section, evaluation metrics that are relevant to this study are introduced. ***This is a two-class study. Positive class are the patients who have deceased in 30-days after a hip fracture surgery where Negative class are the patients who have survived 30-days.*** The author firstly describes the important terms for the evaluation metrics.

- True Positives (TP): Correctly predicted samples which are originally positive
- True Negatives (TN): Correctly predicted samples which are originally negative
- False Positives (FP): Wrongly predicted samples which are originally negative
- False Negatives (FN): Wrongly predicted samples which are originally positive

In figure 2.5, an example of a Confusion Matrix can be seen which is a table used for representing these 4 values.

		Actual Class	
		1	0
Predicted Class	1	TP	FP
	0	FN	TN

Figure 2.5: Example Confusion Matrix

Now, evaluation metrics which measure different kinds of performances, will be presented.

- Specificity or True Negative Rate (TNR), evaluates the proportion of actual negatives that are correctly predicted

$$Specificity = \frac{TN}{FP + TN} \tag{2.3}$$

- Recall or Sensitivity or True Positive Rate (TPR), evaluates the proportion of actual positives that are correctly predicted

$$Recall = \frac{TP}{TP + FN} \tag{2.4}$$

- False Positive Rate (FPR), evaluates the proportion of actual negatives that are incorrectly predicted

$$FPR = \frac{FP}{TN + FP} = 1 - Specificity \tag{2.5}$$

- Precision or Positive Predictive Value(PPV), evaluates the proportion of actual positives that are predicted as positives

$$Precision = \frac{TP}{TP + FP} \tag{2.6}$$

- Negative Predictive Value(NPV), evaluates the proportion of actual negatives that are predicted as negatives

$$NPV = \frac{TN}{TN + FN} \tag{2.7}$$

- F1-score evaluates the balance between precision and recall

$$F1\text{-}score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{2.8}$$

- Accuracy, evaluates the proportion of all correctly predicted samples regardless of their class

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.9}$$

- Receiver Operating Characteristic (ROC) curve is a graph where TPR is plotted against FPR. Area Under the ROC Curve (AUC), shows the capability of the model in distinguishing classes in binary classification problems. Analyzing this curve gives a better understanding of the trade-off between specificity and sensitivity According to [39], AUC between 0.7 and 0.8 is considered as acceptable discrimination, AUC between 0.8 and 0.9 is considered as excellent, AUC above 0.9 is considered as outstanding. An illustration of ROC-AUC can be seen in figure 2.6

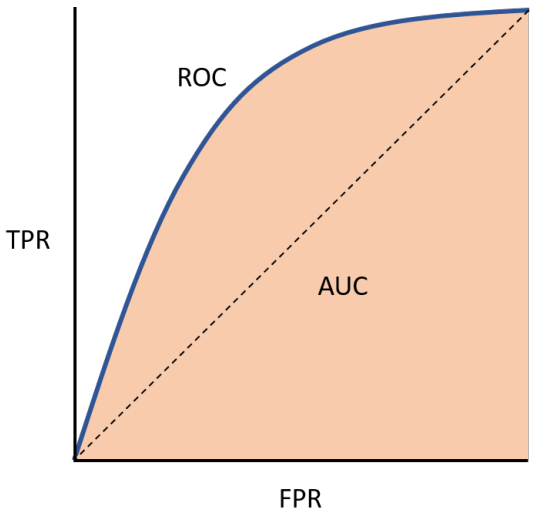


Figure 2.6: Example ROC Curve

- Hosmer–Lemeshow (HL) is a goodness-of-fit test for logistic regression. A significant result (eg. <0.05) indicates that there is a lack of fit in the model [40]. HL test statistic is given by:

$$\begin{aligned} &= \sum_{g=1}^{G} \left(\frac{(O_{1g} - E_{1g})^2}{E_{1g}} + \frac{(O_{0g} - E_{0g})^2}{E_{0g}} \right) \\ &= \sum_{g=1}^{G} \left(\frac{(O_{1g} - E_{1g})^2}{N_g \pi_g} + \frac{(N_g - O_{1g} - (N_g - E_{1g}))^2}{N_g (1 - \pi_g)} \right) \\ &= \sum_{g=1}^{G} \frac{(O_{1g} - E_{1g})^2}{N_g \pi_g (1 - \pi_g)} \end{aligned} \tag{2.10}$$

Where O_{1g}, E_{1g}, O_{0g}, E_{0g}, N_g, and π_g denote respectively the observed class = 1 events, expected class = 1 events, observed class = 0 events, expected class = 0 events, total observations, predicted risk for the gth risk decile group, and G is the number of groups [41]. For a deeper understanding of this test statistic, reader is advised to read [40].

There is of course a trade-off between some of the mentioned metrics such as precision and recall. Evaluation metrics and how they are used in this study will be elaborated in the methodology. AUC and HL test statistic is important to understand the mentioned results in chapter 3, Related Work.

2.7 Missing Value Imputation

Almost all datasets contain missing values. Handling these missing values is an important part of preprocessing in data science projects. In this section, the author will introduce different types of missing data, and various techniques used to handle them in this study.

One common way to distinguish missing data is regarding its randomness. There are 3 classes in this differentiation [42].

- Missing Completely at Random (MCAR), when the reason for the missingness is not related to anything within the dataset.
- Missing not at Random (MNAR), when the reason for the missingness is related to the variable itself.
- Missing at Random (MAR), when the reason for the missingness is not related to the variable itself but on other observed data.

Moreover, the amount of missing values of a variable or amount of missing values a unit has is also an important aspect during this process.

Techniques used to handle missing values in this study are listed as follows:

- Listwise deletion(complete case analysis), deleting a unit from the dataset if one or more variables of that unit are missing.
- Mean, mode, median imputation, filling missing values with mean, mode(most frequent value), and median of the variables respectively. It is very easy to implement, however, it leads to bias as all missing values of a variable are filled with a fixed value.
- Iterative Imputation: Multivariate imputation by Chained Equations (MICE), a technique for imputing missing values by mapping each variable with missing values as a function of other variables iteratively [43], i.e. training a regression model to predict the missing variable by using other variables.

3 RELATED WORK

3.1 30-days Mortality Prediction on Elderly Hip Fracture Patients

There have been many studies to find predictors for mortality after hip fracture. As comorbidities have a big influence in healthcare, they also do in hip fractures, therefore, finding relevant comorbidities to early mortality is important for accurate predictions. Furthermore, there have been various studies in order to predict the risk of early mortality of hip fracture patients before the operations, in those studies, usually, the end goal is to come up with a risk model to score the patients. In almost all of the studies, in order to make the model understandable and easier to use, coefficients of the regression models are transformed into risk scores. In this section, the findings and methodologies of these studies will be reviewed.

In [44], research is conducted to determine the risk factors of 30-day mortality of orthogeriatric trauma patients. Hip fracture patients are included in this group. It was found that increased age, male sex, decreased hemoglobin levels, living in an institutional care facility and a decreased Body Mass Index (BMI) are independent risk factors for 30-day mortality. In [4], it was concluded that advanced age, low BMI, and high Charlson Comorbidity Index (CCI) are independently related to postoperative 30-day mortality after a hip fracture surgery but admission glucose concentration has no association.

In [3], the association of case-mix (age, gender, fracture type, pre-fracture residence, pre-fracture mobility, ASA scores) and management (time from fracture to surgery, time from admission to surgery, the grade of surgical and anaesthetic staff undertaking the procedure and anaesthetic technique) variables to 30 and 120-day mortality after hip fracture surgery were studied. From the multivariate logistic regression analysis, it was concluded that all the case-mix variables have a strong association with post-operative early mortality [3]. By contrast, it was found that management variables have no significant contribution to the post-operative early mortality except the grade of anaesthetist [3].

Maxwell et al. [5] aimed to predict 30-day mortality in hip fracture patients having surgery, it was found that advanced age, male sex, having more than 2 comorbidities, having mini-mental test score less than or equal to 6 (out of 10), admission haemoglobin concentration (≤10 g dl−1), living in an institution, and presence of malignant disease are independent predictors of early mortality. These variables then used in logistic regression to constitute the Nottingham Hip Fracture Score (NHFS) which measures the risk of mortality. It was shown that NHFS(Nottingham Hip Fracture Score) had AUC of 0.719 [5]. The NHFS was validated externally [45] but in [46], it was taken to one step further and got validated with national data set and recalibrated. This recalibration improved the fit of predicted and observed rates of mortality to a p-value of 0.23 which was previously smaller than 0.0001(Hosmer–Lemeshow statistic).

Almelo Hip Fracture Score (AHFS) was developed, validated, and compared with an adjusted

version of NHFS (NHFS-a) [6]. Results showed that AHFS had an AUC 0.82 whereas NHFS-a had 0.72. Both models showed no lack of fit between observed and predicted values ($p > 0.05$, Hosmer-Lemeshow test). AHFS was also calculated with a multivariate logistic regression model [6]. Besides the variables used in NHFS-a; ASA score, Parker Mobility Score (PMS), the Dutch Hospital Safety Management Frailty score (VMS) Physical limitations, and VMS Malnutrition were also included in the AHFS model [6].

The Hip fracture Estimator of Mortality Amsterdam (HEMA) was developed as a risk prediction model for 30-day mortality after hip fracture surgery [7]. Logistic regression analysis was used to detect relevant variables to compute the risk [7]. The analysis ended up by finding 9 relevant factors affecting early mortality, namely, age ≥85 years, in-hospital fracture, signs of malnutrition, myocardial infarction, congestive heart failure, current pneumonia, renal failure, malignancy, and serum urea >9 mmol/L [7]. Corresponding 9 variables were used to constitute the final model [7]. The AUC was 0.81 and 0.79 in the development cohort and validation cohort respectively, the Hosmer–Lemeshow test showed no lack of fit in both cohorts ($p>0.05$) [7].

In [8], Brabant Hip Fracture Score (BHFS-30) was developed to predict the risk of early mortality after hip fracture surgery and internally validated. By the use of manual backward multivariable logistic regression, it was found that age, gender, living in an institution, admission serum haemoglobin, respiratory disease, diabetes, and malignancy are independent predictors of risk in early mortality [8]. However, in BHFS-365, which is the risk score for mortality in 1 year, risk factors included cognitive frailty and renal insufficiency [21]. BHFS-30 showed acceptable discrimination with an area under the ROC curve of 0.71 after the internal validation and the Hosmer–Lemeshow test indicated no lack of fit ($p>0.05$) [8].

In [47], it was aimed to define the factors affecting in-hospital and 1-year mortality after hip fracture, moreover, using these factors a model was built to identify the in-hospital and 1-year mortality risk of patients. By entering all the variables using a multivariable backward selection procedure for logistic regression ($p<0.05$ for retention in the model), independent determinants were found; older age, male sex, long-term care residence, chronic obstructive pulmonary disease (COPD), pneumonia, ischemic heart disease, previous myocardial infarction, any cardiac arrhythmia, congestive heart failure, malignancy, malnutrition, any electrolyte disorder, renal failure [47]. The interaction between variables was tested however none achieved statistical significance [47]. In-hospital and 1-year mortality predictions used the same variables with the same adjusted odds, meaning that factors affecting each of them are essentially the same [47]. Predictions for in-hospital mortality achieved AUC of 0.83 on training and 0.82 on the validation set, without showing any lack of fit according to Hosmer-Lemeshow statistic. In addition to that, predictions for 1-year mortality achieved AUC of 0.75 on training and 0.74 on the validation set [47].

Apart from these risk scores particularly dedicated to predicting the risk of hip fracture patients, there have been studies to come up with more generic risk scores which are also applicable to hip fracture patients as well as other relevant patients. These are The Charlson Comorbidity Index (CCI), a model to predict the risk based on classification of comorbid conditions [48], Orthopaedic Physiologic and Operative Severity Score for the enUmeration of Mortality and Morbidity (O-POSSUM), a model to predict the risk in orthopaedic surgery [49], Estimation of Physiologic Ability and Surgical Stress (E-PASS), a model that predicts the post-operative mortality risk, comprised of a preoperative risk score, a surgical stress score, and a comprehensive risk score [50], the Surgical Outcome Risk Tool (SORT), a model to predict the risk of 30-day mortality after non-cardiac surgeries [51].

In [45], evaluation of the 30-day mortality prediction models (CCI, O-POSSUM, E-PASS, a risk

model by Jiang et al. [47], NHFS, a model by Holt et al. [3]) was done. All models except the O-POSSUM reached acceptable discrimination (AUC > 0.70). It is reported that the best AUC (0.78) belongs to the risk model by Jiang et al. [47]. Models who are specifically designed for hip fracture cases (model by Jiang et al. [2], NHFS, model by Holt et al. [3]) showed a significant lack of fit according to Hosmer–Lemeshow statistic. Marufu et al. evaluated SORT in [46], it was found that SORT had AUC of 0.70 with a significant lack of fit despite the recalibration.

Overview of the results of studies on 30-days mortality of elderly hip fracture patients including the results of this thesis could be found in table 3.1

Study	Technique Used	Number of Features Used	AUC Score
NHFS [5]	Logistic Regression	7	0.719
AHFS [6]	Logistic Regression	>10	0.82
HEMA [7]	Logistic Regression	9	0.79
BHFS-30 [8]	Logistic Regression	7	0.71
This thesis	CNN[1], Random Forest Classifier, Early Fusion, Random Over Sampling, Iterative Imputation of missing values with KNeighbors Regressor	>100(including image modality)	0.742

Table 3.1: Overview of the results of studies on 30-days mortality of elderly hip fracture patients

3.2 Applications of Multimodality on Medical Domain

Suk et al. tried to diagnose Alzheimer's disease with a multimodal approach using deep learning [52]. The fusion of multimodal information from Magnetic Resonance Imaging (MRI) and Positron Emission Tomography (PET) data was done by a novel method they proposed using multimodal DBM (Deep Boltzmann Machines) [52]. This method outperformed competing methods in terms of accuracy [52]. Moreover, they further investigated the trained model visually and found that their method can hierarchically expose the latent patterns in MRI and PET [52].

In 2005, Kenneth et al. fused ECG, blood pressure, saturated oxygen content and respiratory data in order to improve clinical diagnosis of patients in cardiac care units, they concluded that better results can be achieved with the novel fusion system they proposed [53]. Moreover, Bramon et al. proposed and evaluated an information-theoretic approach for multimodal data fusion, it was found that their approach showed promising results on medical data sets [54]. Nunes et al. [55] developed a multimodal approach that integrates a dual path convolutional neural network (CNN) processing images with a bidirectional RNN processing text. Experiments showed promising results for the multimodal processing of radiology data [55]. Pre-training with large datasets improved the AUC 10% on average [55].

[1] Convolutional Neural Networks

4 TERMINOLOGY

As this research is concerned with the application of machine learning to the health care domain, there might be some overlapping terminologies amongst these disciplines or other terminologies that are not clear for readers coming from backgrounds of either of the domains. For this reason, in this section, the author would like to describe some of the terminology used in the rest of this thesis to clarify what is exactly meant by particular terms.

4.1 Modality

In section 2.4, Multimodal Machine Learning has been described. It was mentioned that a modality implies how something happened or experienced. In this research, the term ”modality” is used to differentiate the data in a way that humans process it to understand. To be more precise, regarding hip fractures, present modalities are identified as follows:

- Structured modality, which includes only data that are structured such as survey questions, lab tests, emergency room measurements, medication usage. In this modality, for a particular variable, the question is always the same, and answers might be numerical with some range or nominal with a fixed amount of categories.
- Image modality, which includes thorax (chest) and pelvis (hip) x-ray images of the patients. In exceptional cases, there are also other imaging involved such as computational tomography (CT).
- Text modality (natural language), which includes letters from nurses, surgeons, radiologists where they tell about their observations and findings for patients in an unstructured or semi-structured form.
- Signal modality, which includes only electrocardiograms (ECG).

Although the author has listed 4 modalities involved with hip fractures patients, the scope of this thesis does not cover the study of all of them. Further information on this will be given in chapter 6.

4.2 Source of Data

As its name suggests, the source of data is concerned with where the data is coming from. This could be an alternative to ”modality” for grouping variables, maybe a more detailed version. For example in the modality section, it was mentioned that Structured Data consists of

multiple sets of variables, such as survey questions, lab tests, emergency room measurements. One can obviously see that source of variables concerning lab tests and emergency room measurements are different as one of them is recorded by the lab and the other by the emergency room. However, this is not always straightforward. Especially, when a particular question can be answered in different ways. A very important example is the comorbidities of the patient. There are multiple sources to find comorbidities of a patient. Going through old letters written by doctors or nurses is the first option. Looking at financial hospital records of the patient is the second option. When the doctors apply a treatment, they have to enter relevant financial codes in the patient's file for declaration purposes to the insurance. The final option to learn the comorbidities of the patient is simply by surveying the patient or their family and keeping structured records of it. It can be seen that, as the source changes, modality might also change which makes the extraction of a variable more challenging.

4.3 Subject of Data

The subject of data describes what concept, discipline, or body part/organ a variable is related to. It is another dimension of grouping variables. To make it more clear, the author would like to give the example of lab tests. There are about 18 lab tests employed in the evaluation of a patient. However, different lab tests are concerned with answering questions about different subjects such as liver, kidney, infection. More examples could be given on survey questions which could have subjects such as nutrition, mobility, cognitive problems.

4.4 Parameters and Hyperparameters

In machine learning tasks there are two types of parameters involved. The first type of parameter is the one that is learned and adapted by the model during the training phase. As an example, weights of a neural network or decision nodes of a decision tree are falling under this type, they are learned from data and in the rest of the thesis, they will be called by the term ”parameter”. The second type will be called ”hyperparameter”. Hyperparameters are determined and fixed before the training of the machine learning model has started. For example, how many layers and neurons will a neural network have, or what is the maximum depth a decision tree can grow, how many estimators a random forest model can have, these types of questions are answered and hyperparameters are set before the training begins.

4.5 Train, Validation and Test sets

This section describes the three sets used in machine learning when building a model. Train, validation, and test terms could be used in different meanings in different domains therefore, the author finds it necessary to mention how they are used in this study.

The train set is used to initially train a machine learning model. The validation set is used to validate and select the machine learning model, imputation technique, variables used, generally all the hyperparameters involved in the methodology. The test set is used to estimate and evaluate how the model will perform in real-life. This set must include only the samples which are not seen by the model. The main reason for having a validation set beside the test set is, not to overfit on test data. Basically, during the training, parameters are learned, and during the

validation, hyperparameters, which perform better are identified. Once the hyperparameters are validated, one can use both train and validation set to train their machine learning models and evaluate them on the test set. One should note that the performance on the test set depends significantly on the split especially with relatively smaller datasets. Furthermore, it is very important to have a test set that is representative of the real-life application of the model.

5 METHODOLOGY

In this chapter, the author will describe the methodology used in this research. The diagram in figure 5.1 can be reviewed to have general understanding of stages in the methodology.

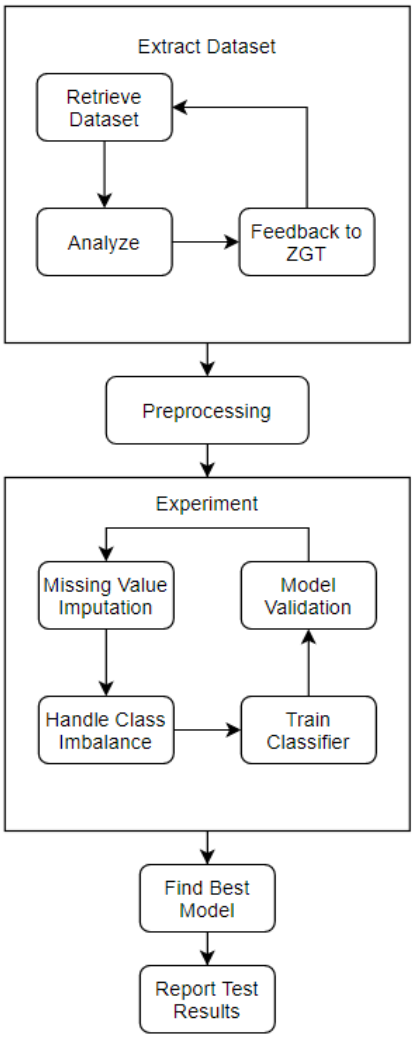


Figure 5.1: Diagram illustrating the stages in the methodology.

5.1 Extract Dataset

The first main stage of the methodology is to extract the dataset from the hospital database systems. This stage was the one which took the longest time. In this part, a dataset was received

from the hospital, it was analyzed and problems related to it was reported to the hospital, this process repeated iteratively to achieve the most complete dataset one can obtain in the given time. Regarding image modality dataset, two deep learning models, one for chest x-rays and one for hip x-rays, had to be developed for image selection as there are more than one images attached to each patients file which are not always correctly labeled. Further details on model development for image selection could be found in appendix A. After doing pre-selection with the built deep learning models, patients who do not have desired imaging were reviewed manually and corrected if it was possible. In the end of this analysis, patients who had missing imaging either on chest part or hip part were excluded from the study.

5.2 Preprocessing

Details of the integration with the dataset of [56] and other preprocessing steps are described in section 6.2.

5.3 Experimenting

The general execution of the experiments was done in a sequential way so that, insights from one set of experiments were used to shape the next generation of experiments. At the end of the experimenting stage, the best model based on the validation set was identified and results on the test set were reported. The multimodal approach adopted during experimenting which constitutes the technical contribution is explained in Chapter 7.

5.4 Evaluation

The main performance evaluation metric was selected as Area Under the Curve (AUC). Higher this metric becomes, a model can perform better discrimination between two classes. Descriptions of related performance metrics can be found in section 2.6. Although metrics such as precision, recall, specificity are of importance, they are not used to evaluate the results of experiments. Because one can change the decision threshold of a model and have completely different results for these metrics with the exact same model. Adjustment of the decision threshold is left out for the future users of the model.

6 DATASET PREPARATION

The study period used in this dataset is between 01-04-2008 and 31-01-2020. All the patients who are 70 years or older and have admitted to the Emergency Room (ER) with an acute hip fracture were included. The selection is based on diagnostic treatment code (DBC) which is '218 Femur, proximal (+collum)'. Patients with a femur fracture, periprosthetic fracture, or pathological fracture were excluded. Patients who had total hip arthroplasty or deceased before the surgery were excluded. Furthermore, patients without thorax or hip/pelvis x-rays were also excluded. With all inclusion and exclusion filters, the complete dataset ended up with 2404 patients. The distribution of the patients according to years including class categories can be seen in fig 6.1. In total, the dataset includes 2211 patients who have survived 30 days after hip fracture surgery and 193 patients who have deceased within 30 days after the operation. Meaning that the 30-days postoperative mortality rate (positive sample rate) of this dataset is 8%.

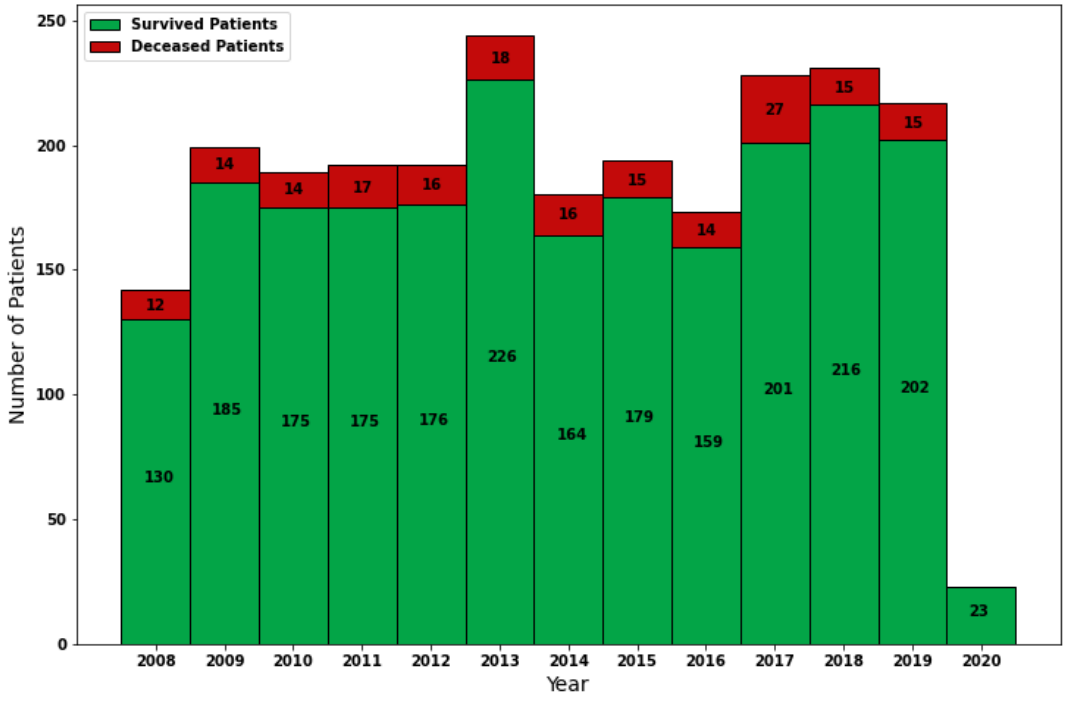


Figure 6.1: Patient Distribution in Years with Class Categories

Variables used in the dataset are extracted from different data sources of the hospital. The different data sources can be listed as follows: Laboratory including lab tests, Emergency Room(ER) including physical exam results and vital signals, clinic including most of the structured data, diagnostic treatment codes (DBC) including comorbidities, radiology including x-ray images and dataset of [56]. As mentioned in section 4.1, there are 4 modalities involved in the data of hip fracture patients, however, the scope of this study covers structured modality and image modality. Structured modality and image modality were retrieved by the database systems of Ziekenhuis Groep Twente (ZGT, Hospital Group Twente). Text modality which contains a lot of useful information had not been processed. However, a study that had overlapping patients with this study, was conducted in 2018 by Nijmeijer et al. at ZGT [56]. During the generation of the dataset of that study, text modality was processed by researchers and converted to structured data manually without any automation tool. Therefore, with the goal of improving the quality of the current dataset, some of the variables of [56] were mapped to the variables of the current dataset for overlapping patients as they point to the same information. However, this information could be missing in some cases in the dataset of this study, thus [56] used to complement these missing parts to some extent with the appropriate mapping of values. This improvement was required due to the fact that different variables were started being collected in the structured form in different periods, resulting in a dataset with a significant amount of missing values. The best way to handle missing values is not to have missing values as much as possible. By the integration of the dataset of [56], the missing value rate was decreased on average by 30%. Details of this integration and other preprocessing steps are explained in section 6.2. In figure 6.2 and 6.3, missing value identification heatmaps can be seen before and after the integration with the dataset of [56] respectively. A yellow line indicates that the corresponding variable in the vertical axis is missing for the patient in the horizontal axis. The pattern of different collection start dates also becomes obvious on these figures as the patients are sorted according to admission dates.

With respect to image data, there are two series of images, namely pelvis x-rays and chest x-rays. However, in some cases instead of the complete pelvis, an x-ray of only one hip exists due to the specific order of the doctors but these are also treated as pelvis x-rays in this study. Moreover, usually, there are more than one view for each series besides the AP (Anterior-Posterior) view, such as lateral, axial, lauenstein views. However, the quality of views except for AP drops significantly due to the effort required by imaging position, especially when the patient is in strong pain. For this reason, all imaging used in the study is AP views.

Variables used in the dataset and their subjects are presented in tables 6.1,6.2. These two tables do not indicate any grouping, they were just split into two, due to the high number of variables.

6.1 Small Dataset

Due to the fact that a very high amount of missing values depend on the date, the dataset was shrunk by deleting patients who had admitted before a particular date. To be precise, 01-04-2012 was selected as the date to exclude patients admitted before which resulted in 1654 samples. This date was selected based on the data collection start dates. This dataset will be called as small dataset in the rest of the thesis. Although the main dataset used during experiments was the complete dataset, the important stages of the experiments were repeated on the small dataset. This can be considered as a missing value imputation technique and it is very similar to listwise deletion (complete case analysis) as explained in section 2.7. The similarity to listwise deletion comes from the fact that all the patients before a particular date

miss most of the variables. The reader is encouraged to review the missing value illustration in figure 6.3 to have a better understanding of the conceptual similarity of this application with listwise deletion. For example, if one deletes all patients who had admitted before 2012, the proportion of missing values in the dataset would drop drastically.

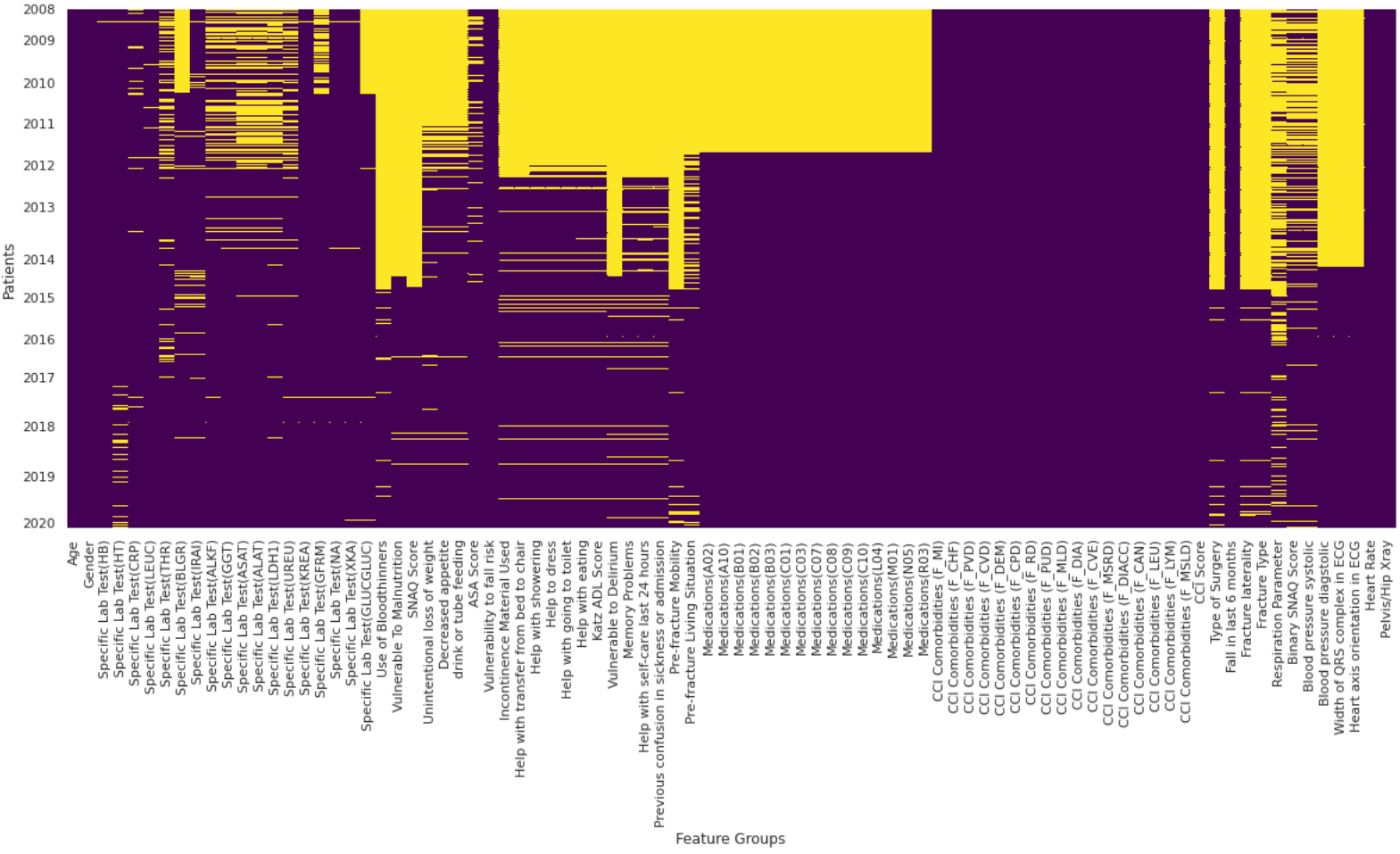


Figure 6.2: Missing value identification heatmap before integrating the dataset of [56], yellow lines indicate missing values

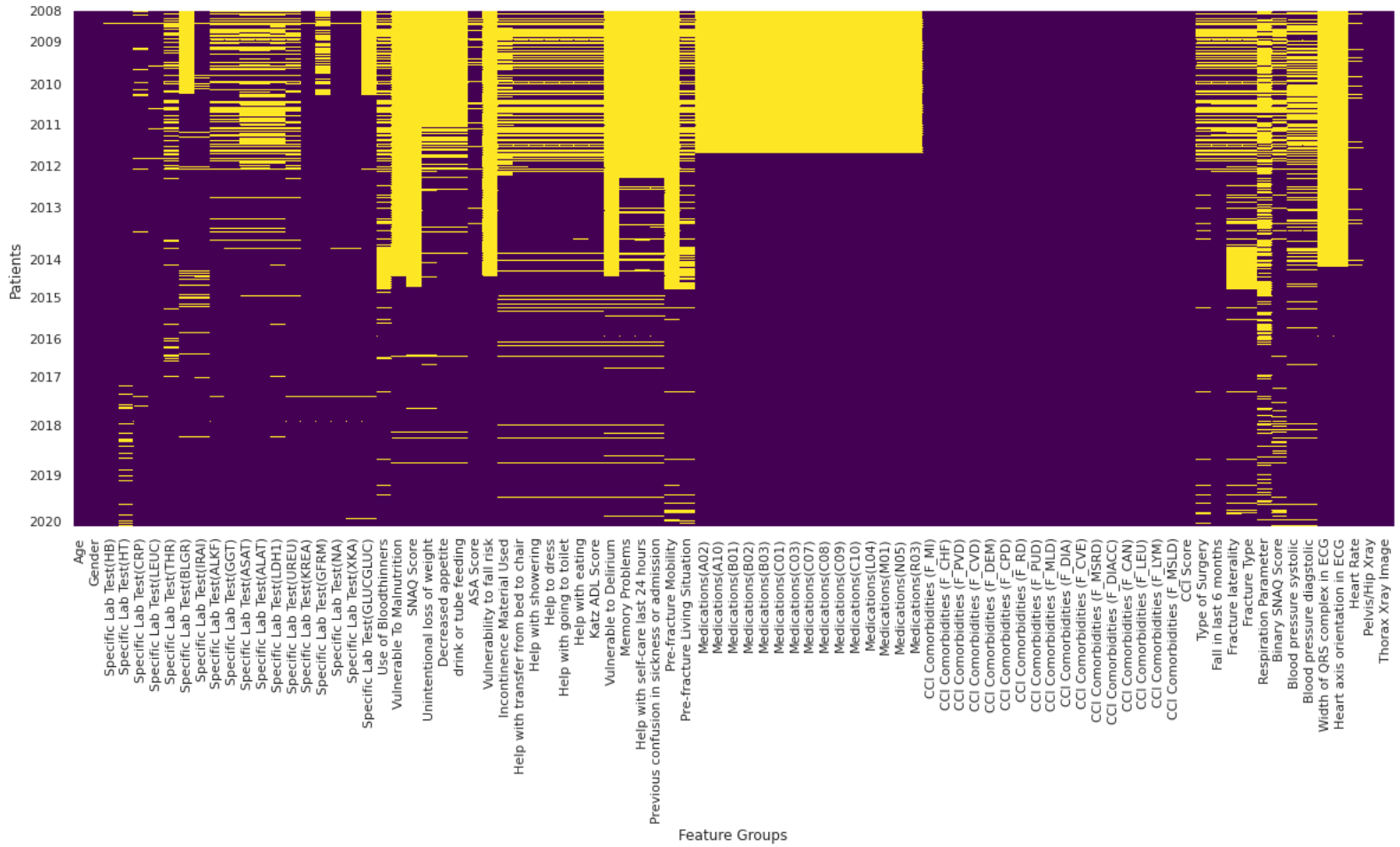


Figure 6.3: Missing value identification heatmap after integrating the dataset of [56], yellow lines indicate missing values

Variable	Subject	Source
Age	Demographics	Clinic
Gender	Demographics	Clinic
HB	Blood	Lab
HT	Blood	Lab
BLGR	Blood	Lab
IRAI	Blood	Lab
CRP	Infection	Lab
LEUC	Infection	Lab
THR	Coagulation	Lab
ALKF	Liver	Lab
GGT	Liver	Lab
ASAT	Liver	Lab
ALAT	Liver	Lab
LDH1	Liver	Lab
UREU	Kidney	Lab
KREA	Kidney	Lab
GFRM	Kidney	Lab
NA	Electrolytes	Lab
XKA	Electrolytes	Lab
GLUCGLUC	Glucose	Lab
Bloodthinners	Medication	Clinic & [56]
Prone to under-nutrition	Nutrition	Clinic
Unintentional loss of weight	Nutrition	Clinic
Decreased appetite	Nutrition	Clinic
Drink or tube feeding	Nutrition	Clinic
SNAQ[1]Score	Nutrition	Clinic
Binary SNAQ Score	Nutrition	ER[2]& [56]
ASA[3]Score	Assessment	Clinic& [56]
Fall risk	Falling	Clinic
Fall happened in last 6 months	Falling	Clinic& [56]
Help with self-care last 24hrs	Activities of Daily Living	Clinic
Incontinence material used	Activities of Daily Living	Clinic& [56]
Help with transfer from bed to chair	Activities of Daily Living	Clinic& [56]
Help with shower	Activities of Daily Living	Clinic& [56]
Help to dress up	Activities of Daily Living	Clinic& [56]
Help with going to the toilet	Activities of Daily Living	Clinic& [56]
Help with eating food	Activities of Daily Living	Clinic& [56]
KATZ ADL[4]Score	Activities of Daily Living	Clinic& [56]
Prone to delirium	Cognitive Problems	Clinic
Memory Problems	Cognitive Problems	Clinic
Previous confusional state	Cognitive Problems	Clinic

Table 6.1: Variables used in this study (part 1)

[1]Short Nutritional Assesment Questionnaire
[2]Emergency Room
[3]American Society of Anesthesiologists Physical Status Classification
[4]Katz Index of Independence in Activities of Daily Living (ADL)

Variable	Subject	Source
Fracture laterality	Fracture	Clinic& [56]
Fracture type	Fracture	Clinic& [56]
Type of therapy	Operation Material	Clinic& [56]
Pre-fracture living situation	Residence	Clinic& [56]
A02(drugs for acid related disorders)	Medication	Clinic
A10(drugs uses in diabetes)	Medication	Clinic
B01(antithrombotic agents)	Medication	Clinic
B02(antihemorrhagics)	Medication	Clinic
B03(antianemic preparations)	Medication	Clinic
C01(cardiac therapy)	Medication	Clinic
C03(diuretics)	Medication	Clinic
C07(beta blocking agents)	Medication	Clinic
C08(calcium channel blockers)	Medication	Clinic
C09(agents acting on the renin–angiotensin system)	Medication	Clinic
C10(lipid modifying agents)	Medication	Clinic
L04(immunosuppressants)	Medication	Clinic
M01(anti-inflammatory and antirheumatic products)	Medication	Clinic
N05(psycholeptics)	Medication	Clinic
R03(drugs for obstructive airway diseases)	Medication	Clinic
Myocardial infarction	Comorbidities	DBC[5]
Congestive heart failure	Comorbidities	DBC
Peripheral vascular disease	Comorbidities	DBC
Cerebrovascular disease	Comorbidities	DBC
Dementia	Comorbidities	DBC
Chronic pulmonary disease	Comorbidities	DBC
Rheumatologic disease	Comorbidities	DBC
Peptic ulcer disease	Comorbidities	DBC
Mild liver disease	Comorbidities	DBC
Diabetes	Comorbidities	DBC
Diabetes with chronic complications	Comorbidities	DBC
Cerebrovascular event(Hemiplegia)	Comorbidities	DBC
Moderate to severe renal disease	Comorbidities	DBC
Cancer[6]	Comorbidities	DBC
Leukemia	Comorbidities	DBC
Lymphoma	Comorbidities	DBC
Moderate or severe liver disease	Comorbidities	DBC
Respiration Parameter	Vital signals	ER
Blood pressure systolic	Cardiology	ECG System
Width of QRS complex in ECG	Cardiology	ECG System
Heart axis orientation in ECG	Cardiology	ECG System
Heart rate	Cardiology	ECG system&ER
Thorax(chest x-ray) (image modality)	Chest	Radiology
Pelvis/Hip x-ray (image modality)	Hip	Radiology

Table 6.2: Variables used in this study (part 2)

[5]Diagnostic treatment code
[6]Discrimination of cancer with or without metastasis is not possible with this variable

6.2 Preprocessing

In this section, the author will explain the details of the integration with the dataset of [56] and other preprocessing steps. The motivation behind the integration was that there were two datasets pointing to same information in different ways and sometimes in one dataset that information is missing. Therefore dataset of [56] used to complement missing parts.

6.2.1 Categorical Variables

All the categorical variables re-coded by one-hot encoding, meaning that, a new binary variable was created for each category of a variable. In case of missing values, all of the binary variables generated for this variable are encoded as 0.

6.2.2 Null Values in Lab Tests

Null values in lab tests come in different forms. There are ‘null’, ‘-volgt-‘, ‘<memo>’, ’Zie op.’, ’STHEM’ and ‘===’ values. The ‘null’ values are simply not available. However, for ‘-volgt-‘ and ‘<memo>’, there is a different story behind. The former means that, the corresponding test has been done but the result was not available and to be followed by that time, meaning that, doctors who call for the decisions do not always wait for all test results to become available for urgent cases. The latter means that, the corresponding test has been done, but there is something wrong with the results, and a person can read about what is wrong if they go into details of that test. A second iteration of extraction from the database was done by ZGT to reduce the number of special null cases such as ’-volgt-’. This reduced number of such cases however, remaining ‘-volgt-‘, ‘<memo>’, ‘===’, ’Zie op.’, ’STHEM’ values are treated as regular ’null’ values.

6.2.3 Non-numeric Lab results

Most of the lab tests are continuous variables. They have specific ranges which indicate that patient is normal if the result lies in that range. For example, the lab test called ‘CRP’, which is related to infection. This test is considered to be abnormal if the result is higher than 10. In this feature, there are values of ‘<1’ which indicates that the test results in a value smaller than 1. But in order to process this as a numeric variable, a transformation had to be made. To this end, following transformations was applied:

- In variable CRP, values of ‘<1’ are replaced by a uniform distribution between 0.5 and 1
- In variable ALAT, values of ‘<5’ are replaced by a uniform distribution between 1 and 5
- In variable GRFM, values of ‘>90’ are replaced by a uniform distribution between 90 and 100

6.2.4 Lab Tests - BLGR & IRAI (Blood Group)

The lab test called as BLGR which is the blood group test was recoded as a binary variable, indicating only if the blood group is ”O” or ”not O”, this puts blood groups of A, B and AB in the category ”not”. On the other hand, lab test ’IRAI’ was recoded in a way that it indicates if the blood group is positive or not positive, i.e. all other values than positive, including missings, are treated as not positive. These recodings took place with the supervision of clinical supervisors from ZGT.

6.2.5 Pre-fracture Living Situation

Pre-fracture Living Situation is one of the variables where 2 sources were used to end-up with a variable with less missing values. These sources are collected data in clinical pathway and the dataset used in [56]. However, categories used in these two sources differ from each other and a mapping had to be done in order to merge two sources. To this end, categories of the dataset of [56], used as default categories, and categories of dataset from clinic was mapped to these with the clinical supervision of ZGT. By the merge of two sources, missing rate of Pre-fracture Living Situation dropped from 39% to 24%.

Categories(values) of pre-fracture living situation variable in clinic (dataset of this study):

A. zelfstandig
B. zelfstandig met (dagelijkse/ADL) hulp
C. alleenwonend
D. verpleeghuis
E. meerpersoonshuishouden
F. verzorgingshuis
G. bejaardentehuis
H. verpleeghuis revalidatie
I. woonzorgcentrum
J. anders, nl.
K. kleinschalig wonen twentsche zorg-centra
L. appartement of flatwoning
M. PG afdeling in verpleeghuis

Categories(values) of pre-fracture living situation variable in the dataset of [56]:

1. zelfstandig
2. zelfstandig met hulp
3. verzorgingshuis
4. verpleeghuis
5. anders bepaald

Categories in the dataset of [56] used as the main categories. Values for variable pre-fracture living situation in the dataset of this study are replaced according to mapping below. Categories which are not possible to map: C, E, L. These were re-coded as 'null' and handled during missing value imputation phase.

A. → 1
B. → 2
C. → null
D. → 4
E. → null
F. → 3
G. → 3
H. → 4
I. → 3
J. → 5
K. → 4
L. → null
M. → 4

6.2.6 ASA and SNAQ Scores

ASA (American Society of Anaesthesiologists) score is one of the variables where 2 sources were used to end-up with a variable with less missing values. No mapping was required during the merge of two sources. Distribution of two sources were checked before merging and no difference was observed. By merging, missing rate of ASA score dropped from 9% to %5. On the other hand, the same process was not applied to SNAQ (Short Nutritional Assessment Questionnaire) score due to the fact that distribution of two datasets differ from each other. Instead of that, another version of SNAQ score, namely Binary SNAQ, was merged from two sources as an alternative of SNAQ score. Both SNAQ Score and Binary SNAQ score were used during modelling. SNAQ Scores which are greater than 5 are considered as invalid values and consequently treated as null.

6.2.7 Variables with subject of Activities of Daily Living

In order to decrease the amount of missing values in variables concerning activities of daily living, a merge with the dataset of [56] took place. Thus, Barthel Index of Daily Activies are used to fill missings of KATZ Index of Independence in Activities of Daily Living. Mapping used during the merge is listed below. Finally KATZ ADL score was recalculated after the merge which resulted a drop in missing rate from 35% to 16%.

- barthel_toileting_preop → hulp_bij_toiletgang
- barthel_transfer_bed_chair_preop → hulp_bij_transfer_bed_stoel
- barthel_bathing_preop → hulp_bij_douchen
- barthel_dress_preop → hulp_bij_aankleden
- barthel_faeces_preop → gebr_incontinentie_mat
- barthel_urinary_preop → gebr_incontinentie_mat

6.2.8 Bloodthinners

Usage of Bloodthinners data was being collected in 3 categories:

A. ja, heb ze door gebruikt (yes, I continued using them)

B. ja, gestopt vlgs afspraak (yes, I stopped according to the appointment)

C. nee, gebruik geen bloedverdunners (no, I do not use blood thinners)

Firstly, these categories were recoded into a binary variable where 1 = A,B and 0 = C. Afterwards, variable 'bl_trost_bloodthinners' from the dataset of [56] was used to fill the missings. Although distribution of two sources were slightly different, the merge was still applied because of the fact missing values occur in earlier times of the study period and the interpretation was that usage of bloodthinners increased slightly in years.

6.2.9 Fracture Type and Surgery Type

Initially, fracture types were recoded with the supervision of ZGT. Then missing values for the fracture type were filled by the variable 'type_heupfractuur' from the dataset of [56] as they point to the same information. This resulted a drop in missing rate from 55% to 25%. Recoding of fracture types are as follows:

- trochantere femur fractuur AO-A2 $\rightarrow$ pertrochantaire fractures
- trochantere femur fractuur AO-A3 $\rightarrow$ pertrochantaire fractures
- trochantere femur fractuur AO-A1 $\rightarrow$ pertrochantaire fractures
- mediale collum fractuur gedisloceerd $\rightarrow$ mediale collumfracturen
- mediale collum fractuur niet gedisloceerd $\rightarrow$ mediale collumfracturen
- subtrochantere femurfractuur $\rightarrow$ subtrochantaire femur
- unspecified $\rightarrow$ null

Similarly to fracture type, type of surgery was recoded first and then merged with the variable 'type_of_surgery' from the dataset of [56]. Missing rate on surgery type dropped from 55% to 16%. Recoding was done as follows:

- glijdende heupschroef (DHS) $\rightarrow$ Internal fixation for femoral neck fracture
- hemiarthroplastiek heup (Kop-Hals Prothese) $\rightarrow$ Endoprosthesis for femoral neck fracture
- intra medullaire pen heup (PFNA) $\rightarrow$ Internal fixation intertrochanteric and subtrochanteric femur fracture
- gecanuleerde schroef heup $\rightarrow$ Internal fixation for femoral neck fracture
- overige $\rightarrow$ Other

6.2.10 Emergency Room Measurements

Emergency room measurements that were used in the study are 'NIBP'(Non-Invasive Blood Pressure), 'RESP'(Respiration Parameter),'KOSNAQ'(Binary SNAQ), 'PR'(Pulse Rate). Although each of the measurements are done usually more than once, only the first measurement was extracted from the Emergency Room Report. PR was used to fill missings of Heart Frequency which is measured by ECG. Binary SNAQ was merged with dataset of [56] as mentioned in section 6.2.6.

7 MULTIMODALITY

In this chapter, the author emphasizes the technical contribution of this study. This is done by explaining and illustrating the designs regarding multimodality. As this study includes both image modality and structured modality, some of the experiments were executed with only image modality, some of them were performed with only structured modality, and on later stages, some of the experiments fused these modalities in various ways to predict 30-days mortality. The general outline of the experiments is illustrated in diagram 7.1. In figure 7.1, small numbers below the ”Model Validation” boxes refer to the corresponding figures explaining the designs of the models in that validation phase.

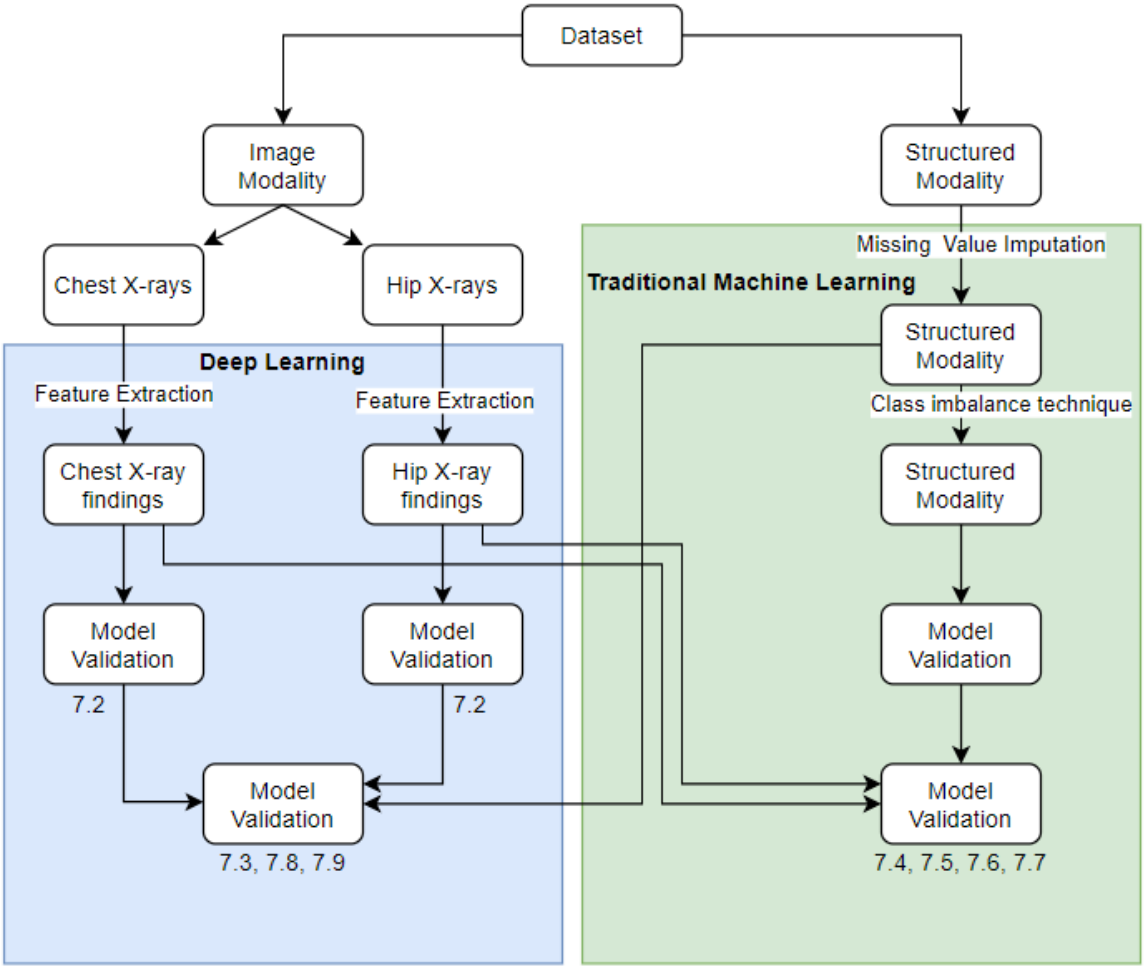


Figure 7.1: Diagram illustrating the outline of the experiments. Small numbers below the ”Model Validation” boxes refer to the corresponding figures explaining the designs of the models in that validation phase.

As the model training for different modalities is done separately, the dataset used during the training, validation, and testing was fixed in the first place. This is due to the fact that, once the output of a model with one modality is used on the training of other modality, there could be information leaks if the train, validation, and test sets are not fixed. This is also one of the main reasons why the cross-validation technique is not employed in most of the study as cross-validation mixes the train and validation sets on each fold.

7.1 Image Modality

This is the main part where the author employed deep learning with the help of transfer learning. The goal at this stage was first to extract features affecting 30-days mortality from the chest and hip x-rays and then also predict 30-day mortality by the use of extracted features. To this end, training at this stage was done with 30-days mortality labels of the patients. 4 convolutional neural network models that were pre-trained on ImageNet (dataset with 14 million images) were selected. These models were then trained with chest and hip x-rays separately. In order to achieve the best model, transfer learning was used in two ways; partial training and full training. Full training refers to train all the weights of a neural network whereas partial training refers to freezing some of the layers and training the weights of remaining(deeper) layers of the model. Once, the author validated models for both image series, these models(a model for chest x-rays and a model for hip x-rays) then tried to be improved by training simultaneously, which means that predecessor of the last layer of these models are concatenated horizontally in the end and backpropagation during the training was applied to both models from the same loss function at the same time. In other words, a bigger model was created which consists of both of the validated models and processes two image series for each patient. Figures 7.2 and 7.3 illustrates the architectures of experiments with image modality. In all the experiments with deep learning part, the class imbalance problem was tackled by the use of adjusted class weights. By that, the effect of positive samples and negative samples to the loss function was re-balanced as if they have the same sample size. Calculation of class weights for negative and positive class are given in equations 7.1 and 7.2 respectively. ”neg”, ”pos”, ”total” denote number of negative samples, number of positive samples and number of total samples respectively. Furthermore, initial bias is added to the last layer output which is useful for initial convergence in imbalanced datasets, calculation of initial bias is given in equation 7.3.

$$NegativeClassWeight = \frac{1}{neg} \times \frac{total}{2} \tag{7.1}$$

$$PositiveClassWeight = \frac{1}{pos} \times \frac{total}{2} \tag{7.2}$$

$$InitialBias = \ln(\frac{pos}{neg}) \tag{7.3}$$

An unsupervised learning technique, auto-encoders, were also tried to extract features from image modality, however, due to low potential, they were discarded in this study. Details on this could be found in appendix B.

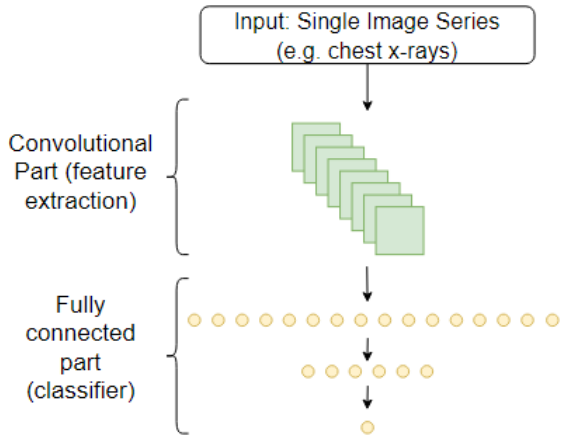


Figure 7.2: Diagram illustrating the architecture of an experiment with single image series

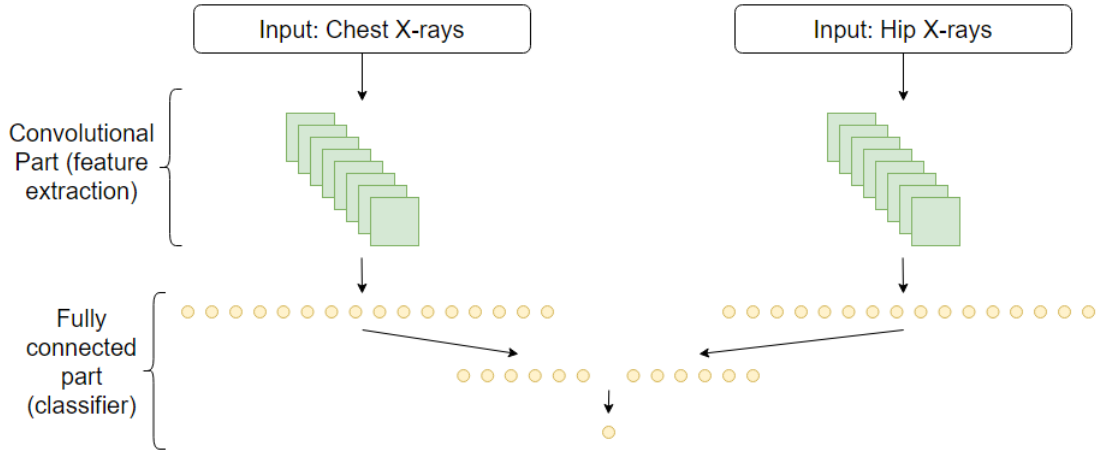


Figure 7.3: Diagram illustrating the architecture of an experiment where both of the image series are used

7.2 Structured Modality

Due to the extremely lower computation times, a higher number of experiments were undertaken with structured modality. These experiments were done as hyperparameter search. Due to the fact that, there are so many hyperparameters to optimize, this stage was split into sub-stages. First sub-stage was concerned on finding the most suitable:

- Missing value imputation technique
- Class imbalance handling technique
- Machine learning algorithm (classifier)

In order to prevent any information leakage, imputation techniques applied to different splits(train, validation, test) separately. For example, let's assume that the age of a patient is missing and for the sake of simplicity, let's assume that, the technique used to fill this value is the mean age of patients. If this patient was in the training set then, it would be filled by the mean age of patients in the training set and vice versa.

The second sub-stage was concerned about finding best hyperparameters on the missing value imputation technique. The third sub-stage was performed to optimize the hyperparameters of the classifier. The third sub-stage is the only part where cross-validation was applied to avoid overfitting. In later stages, cross-validated hyperparameters were used in training with initially fixed train, validation and test splits in order to avoid information leakage.

7.3 Multimodal Fusion

After validating unimodal predictors separately, next is to fuse multiple modalities into one model. This was done in several ways. Two model-agnostic approaches and one model-based approach were employed for this task. Model-agnostic approaches include early fusion and late fusion. Whereas the model-based approach is use of neural networks.

- Early Fusion: It is also logical to think of it as representation learning with respect to 30-days mortality, in this method, neuron outputs of an inner layer are gathered after the feature extraction and transferred to structured modality in order to use in the traditional machine learning algorithm. In this way, these neuron outputs are treated as structured findings related to mortality from image modality. As fully connected part of the networks is relatively smaller, the chosen inner layer(Dense_1 in table 8.22) is the one that is the predecessor of the output layer. Figure 7.4 and 7.5, illustrates the architecture for early fusion experiments.
- Late Fusion: In general, late fusion is done by combining decision outputs of unimodal predictors, however, in this research, it is done by using the output of the last layer from image modality in traditional machine learning algorithms alongside structured modality. Basically, the only difference with early fusion is that, instead of the inner layer(Dense_2 in table 8.22) output from image modality's predictors, the last layers' outputs(Output in table 8.22) are taken as a representation of findings related to 30-days mortality. Figure 7.6 and 7.7, illustrates the architecture for late fusion experiments.
- Multimodal Fusion with Neural Networks: In this method of multimodal fusion, structured modality is also fed into the neural network where image modality had been training. Similar to early fusion, right after the feature extraction part(convolutional parts of neural networks) ends, structured modality is merged with extracted features (Dense_2 in table 8.22) horizontally and then used in the rest of the neural network which is the part with only fully connected layers. One can think of this fully connected part as a classifier part of the neural network. If the training was not done simultaneously, then there would have been no difference with the early fusion method except the classifier used is a neural network instead of a traditional machine learning algorithm such as a decision tree or logistic regression. However, the training is done simultaneously for all of the parts of the network, meaning that, backpropagation from the loss function is done, when the model receives image modality and structured modality at the same time. Figure 7.8 and 7.9, illustrates the architecture for multimodal fusion experiments with neural networks.

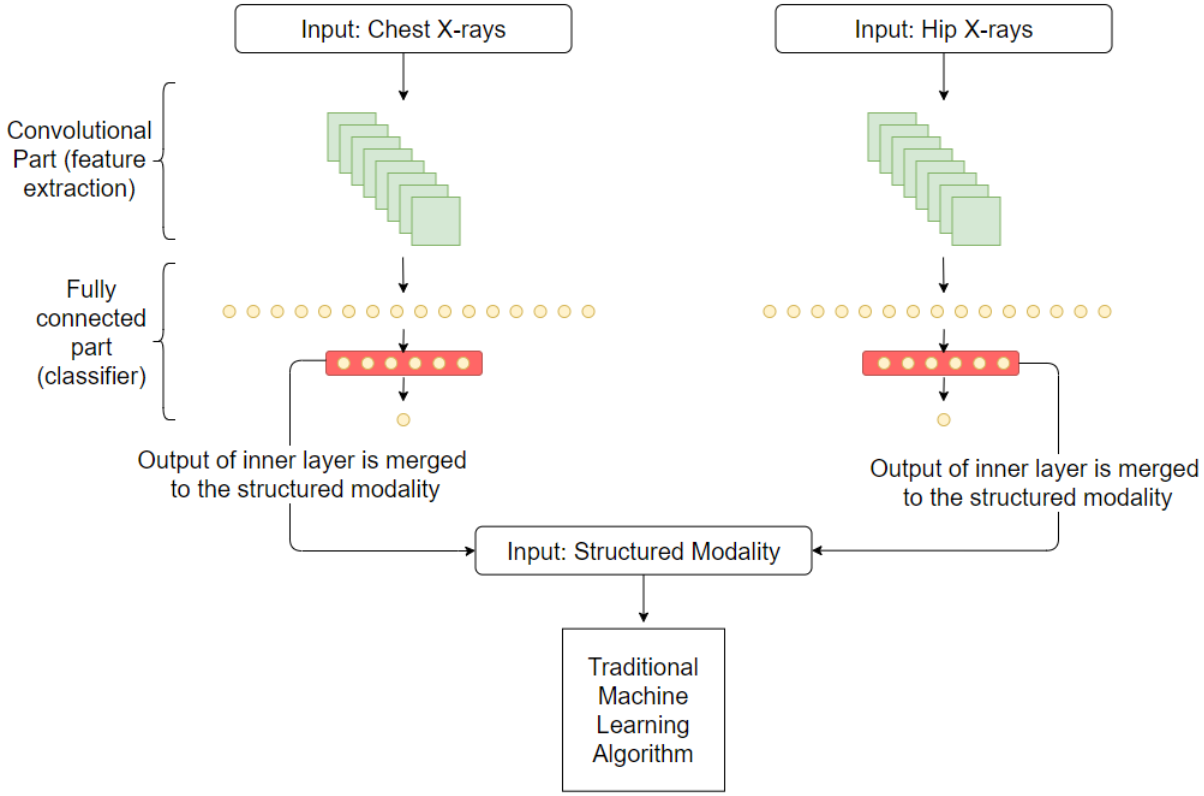


Figure 7.4: Diagram illustrating the early fusion where deep learning models for different image series are trained separately(independently)

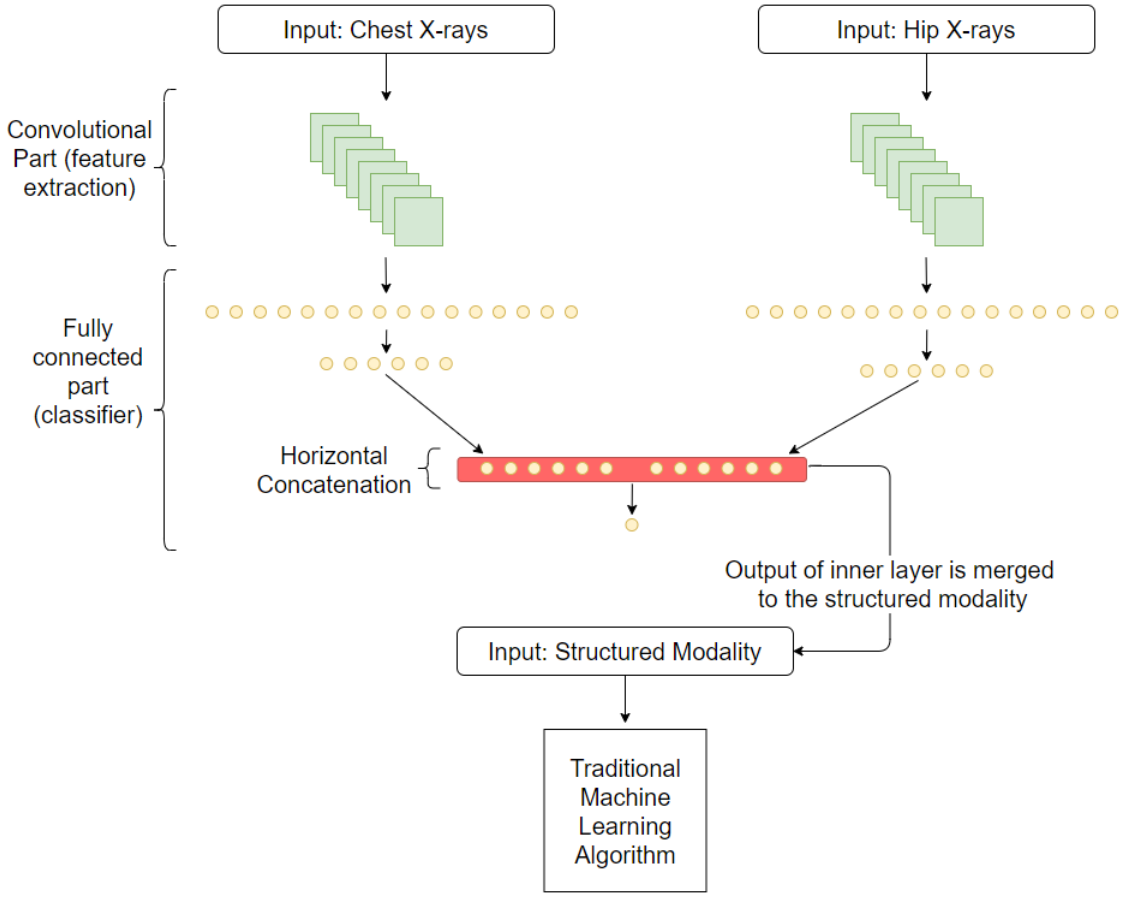


Figure 7.5: Diagram illustrating the early fusion where deep learning models for different image series are trained simultaneously

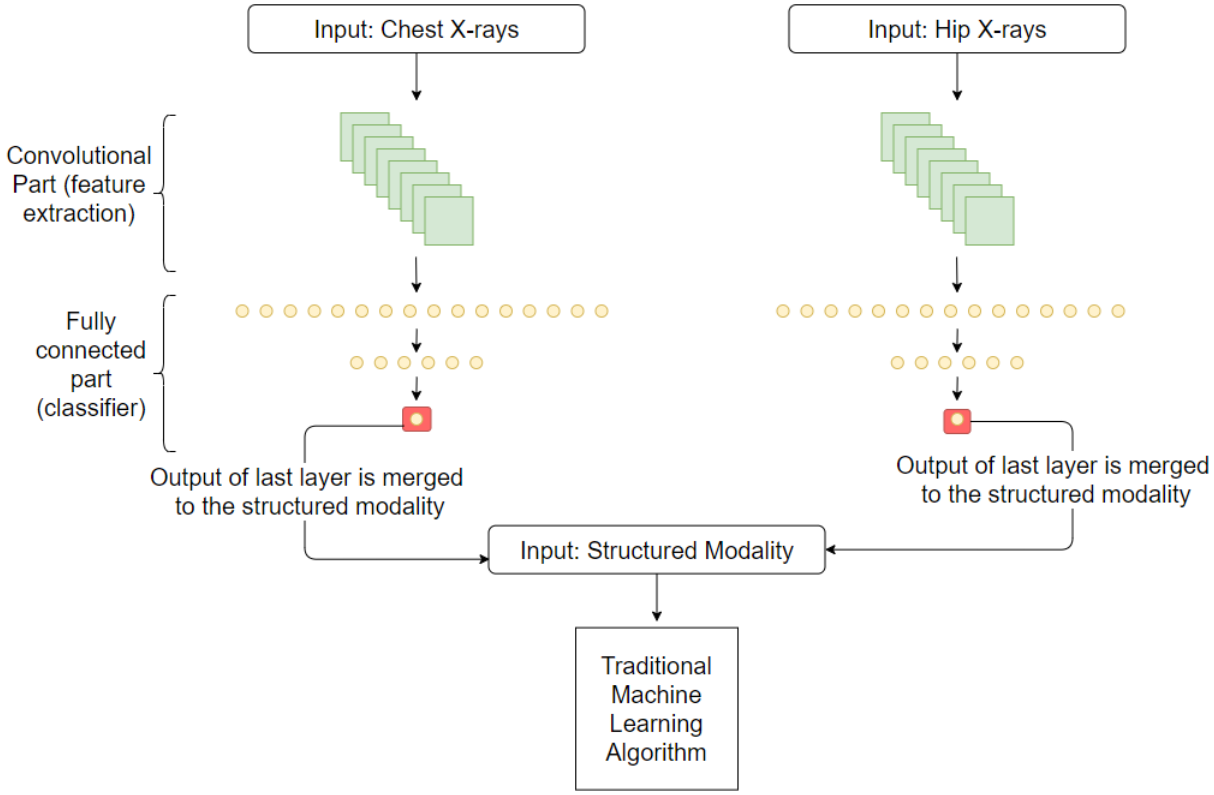


Figure 7.6: Diagram illustrating the late fusion where deep learning models for different image series are trained separately(independently)

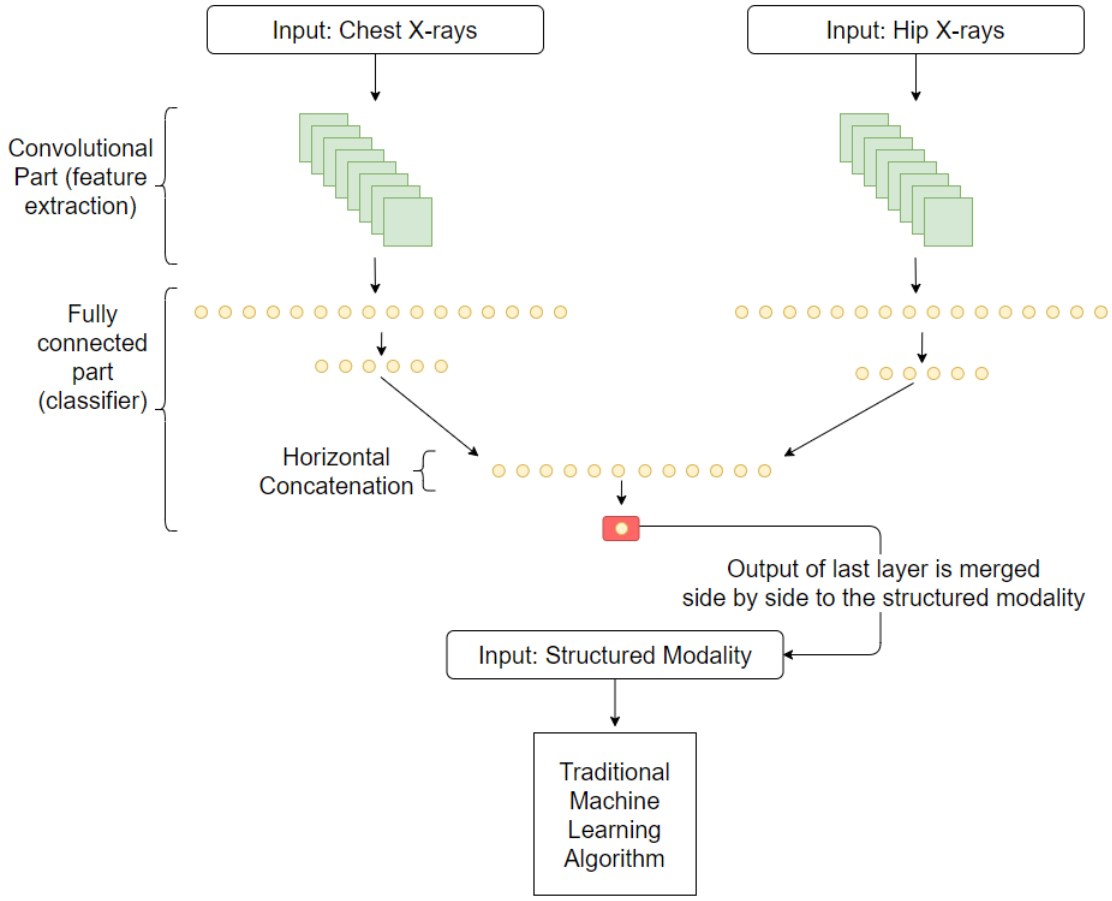


Figure 7.7: Diagram illustrating the late fusion where deep learning models for different image series are trained simultaneously

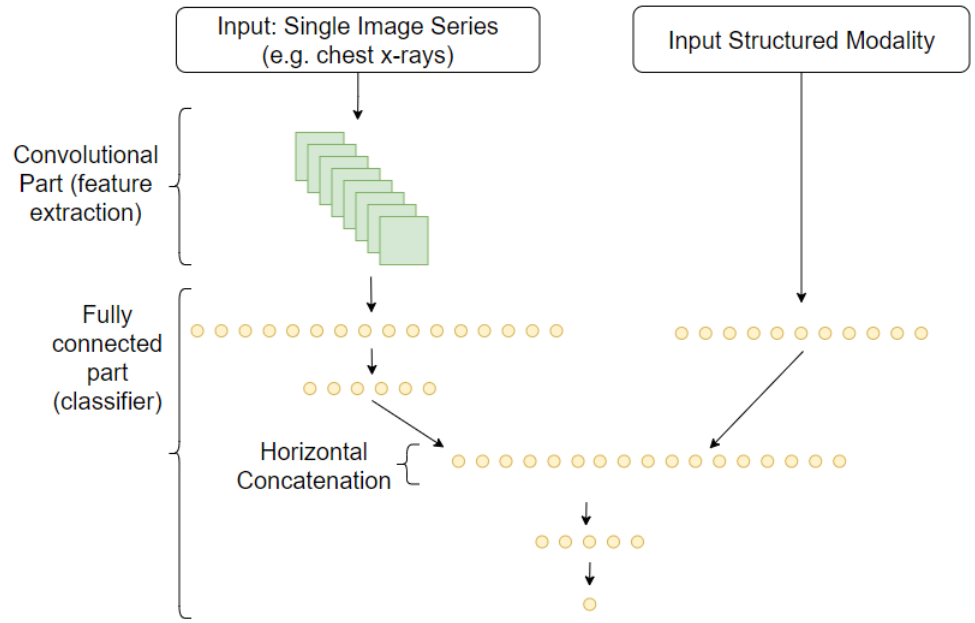


Figure 7.8: Diagram illustrating the architecture of multimodal fusion with neural network where only a single image series is used in an experiment

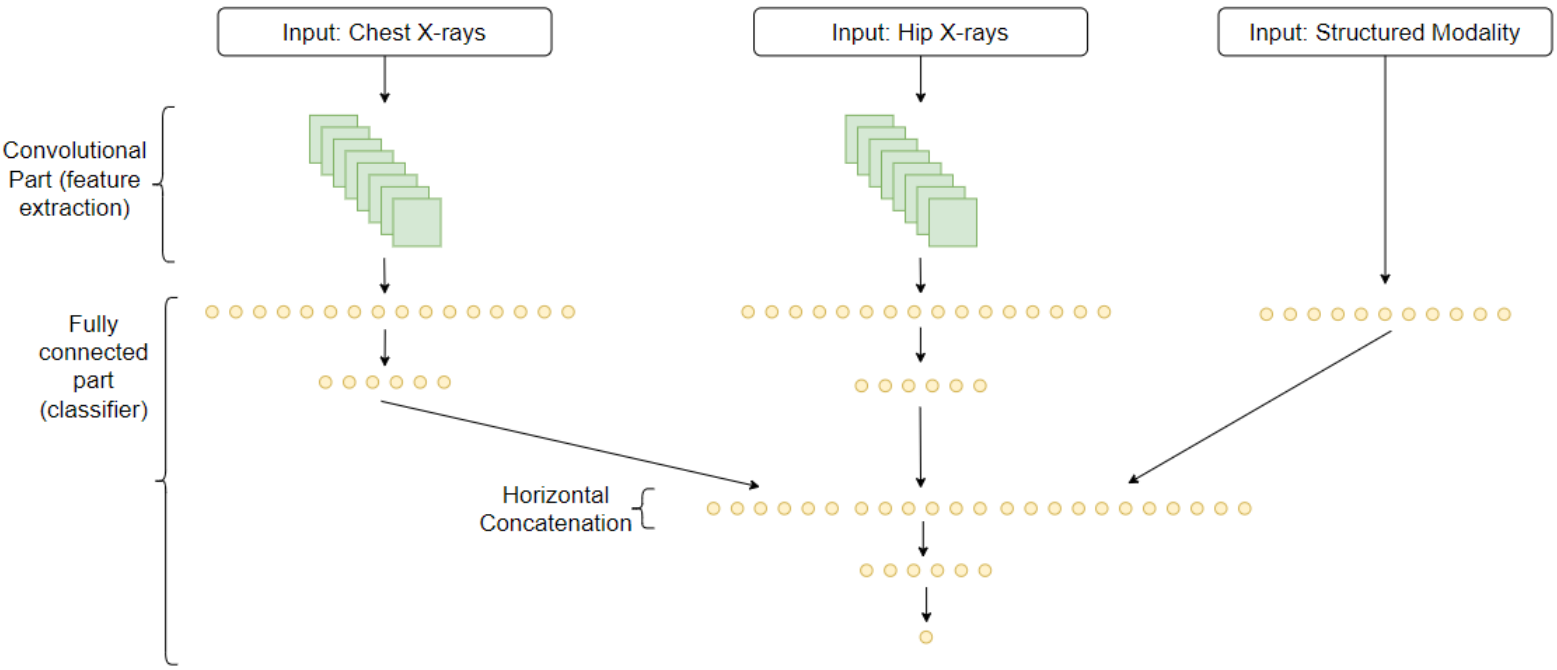


Figure 7.9: Diagram illustrating the architecture of multimodal fusion with neural network where both of the image series are used in an experiment

8 EXPERIMENTAL SETTINGS AND RESULTS

In this chapter, the author will explain the experiment settings and obtained results on each stage of experiments. Train, validation and test sets are split with 50%, 25% and 25% ratios respectively.

8.1 Structured Modality Experiments

8.1.1 Structured Stage 1

Initial experiments had started with structured modality. The first goal was to identify which machine learning algorithm, missing value imputation technique and class imbalance technique are more suitable in this context. To this end, number of candidates have been tried out. The evaluation of these candidates are done based on their performance on the validation set. Therefore, all the scores reported in this chapter will be validation scores unless stated otherwise.

Candidate machine learning algorithms:

- AdaBoost Classifier
- Support Vector Machine Classifier with Linear Kernel (LinearSVC)
- Logistic Regression
- Random Forest Classifier
- eXtreme Gradient Boosting Classifier (XGBClassifier)

Candidate class imbalance techniques:

- ADASYN
- Borderline SMOTE
- NearMiss-2
- Random Over Sampler
- Random Under Sampler
- SMOTE
- Adjusting Class weights

- None (as baseline)

For missing value imputation, a function, Iterative Imputer, from sklearn library is used. It starts with filling all missings with the mean value. Afterwards, with a machine learning algorithm chosen by the user, it imputates missing values based on correlation with other variables iteratively. On the other hand, mean strategy was also used where missing values are filled with the mean value of that variable. This was done in order to have benchmark as it is the simplest imputation technique. To prevent information leak, it is important that this procedure is done after the train, validation and test sets are split.

Candidate machine learning algorithms used for missing value imputation:

- Bayesian Ridge
- Decision Tree Regressor
- Extra Trees Regressor
- KNeighbors Regressor
- Filling with Mean value

With canditates for different task listed above, a grid-search was executed. Meaning that, all of the possible combinations of candidates have been tried out which results in $5 \times 8 \times 5 = 200$ runs. It must be noted that, at this stage, all of the algorithms are tried out with their default hyperparameters except following pre-selected hyperparameters for particular functions described in table 8.1.

Function Name	Hyperparameter Name	Chosen Value
IterativeImputer	maximum number of iterations	20
DecisionTreeRegressor	maximum features	$\sqrt{n\_features}$
KNeighborsRegressor	number of neighbors	20
Logistic Regression	solver	liblinear
Logistic Regression	maximum number of iterations	10000
LinearSVC	loss function	hinge
NearMiss	version	2
AdaBoostClassifier	base estimator	Decision Tree Classifier
AdaBoostClassifier	number of estimators	100
ExtraTreesRegressor	number of estimators	20

Table 8.1: Pre-selected hyperparameters for experiments, remaining hyperparameters were left as default

This stage was run to eliminate the candidates which show poor performance. In table 8.2, summarized results for candidate class imbalance techniques can be observed. Based on these results, it was decided to eliminate NearMiss-2 and Adjusted Class weights techniques for structured modality as they have significant lower results on AUC score. Table 8.3 summarizes the results for missing value imputation algorithms. It was observed that filling with mean values and DecisionTreeRegressor performed relatively worse. Moreover, although remaining algorithms do not result in big differences in terms of AUC scores, ExtraTreesRegressor has

extremely higher computation times and therefore it was also eliminated. Table 8.4 summarizes the results for classification algorithms. Based on the results, Logistic Regression and AdaBoost classifier were eliminated due to weaker performance.

Class Imbalance Handling Technique	Max AUC	Average AUC	Count of Runs
RandomUnderSampler	0.762	0.689	25
RandomOverSampler	0.762	0.677	25
SMOTE	0.762	0.681	25
None	0.754	0.680	25
ADASYN	0.752	0.675	25
BorderlineSMOTE	0.752	0.680	25
Adjusted class weights	0.738	0.653	25
NearMiss-2	0.609	0.518	25

Table 8.2: Summary of Results for Class Imbalance Techniques - Part1

Algorithm for Missing Value Imputation	Max AUC	Average AUC	Count of Runs
KNeighborsRegressor	0.762	0.667	40
BayesianRidge	0.762	0.654	40
DecisionTreeRegressor	0.75	0.64	40
ExtraTreesRegressor	0.752	0.651	40
Filling with Mean value	0.742	0.649	40

Table 8.3: Summary of Results for Missing Value Imputation Techniques - Part1

Algorithm Used in Classification	Max AUC	Average AUC	Count of Runs
RandomForestClassifier	0.762	0.7	40
LinearSVC	0.762	0.685	40
XGBClassifier	0.752	0.68	40
LogisticRegression	0.706	0.669	40
AdaBoostClassifier	0.631	0.536	40

Table 8.4: Summary of Results for Machine Learning Techniques used for classification - Part1

As techniques and algorithms which are not eliminated have very close results to each other, a second round of experiments were run to confirm which candidate is most suitable in their category. Regarding, classifiers, there were 3 algorithms remaining which had average AUC between 0.7 and 0.68. With respect to class imbalance handling, there were 6 techniques remaining which had average AUC between 0.67 and 0.69. Finally, there were 3 missing value imputation techniques left with average AUC range of 0.65-0.67. This resulted in $6 \times 3 \times 3 = 54$ runs. This time the evaluation was done in a sequential way. First, classification techniques were evaluated and the best performing one was selected. Afterwards, missing value imputation techniques were evaluated according to their performance with selected classifier and lastly, class imbalance handling method was evaluated according to their performance with selected classifier and missing value imputation method. According to results in table 8.5, Random Forest Classifier was selected as the most suitable machine learning algorithm for the classification task regarding the structured modality. In table 8.6, performance results for algorithms used in missing value imputation with the Random Forest Classifier are presented. According to the results, it was found that KNeighborsRegressor is the best candidate for this task. And lastly in table 8.7, although there were no big differences between different candidates, it can be seen

that SMOTE and RandomOverSampler has the highest score. However, due to the fact that Random Over Sampling is relatively more straight forward than SMOTE, RandomOverSampler was chosen as the most suitable candidate in this category.

Algorithm Used in Classification	Max AUC	Average AUC	Count of Runs
RandomForestClassifier	0.761	0.732	18
XGBClassifier	0.756	0.725	18
LinearSVC	0.74	0.699	18

Table 8.5: Summary of Results for Machine Learning Techniques used for classification - Part2

Algorithm for Missing Value Imputation	Max AUC	Average AUC	Count of Runs
KNeighborsRegressor	0.761	0.7486	6
DecisionTreeRegressor	0.755	0.732	6
BayesianRidge	0.723	0.7164	6

Table 8.6: Results for Missing Value Imputation Techniques with Random Forest Classifier - Part2

Class Imbalance Handling Technique	Max AUC	Average AUC	Count of Runs
RandomOverSampler	0.761	0.761	1
SMOTE	0.761	0.761	1
ADASYN	0.756	0.756	1
BorderlineSMOTE	0.751	0.751	1
None	0.748	0.748	1
RandomUnderSampler	0.714	0.714	1

Table 8.7: Results for Class Imbalance Techniques with Random Forest as Classifier and KNeighbors Regressor as missing value imputation algorithm - Part2

After finding out the best combination. This model was evaluated on the test set as an educated baseline model. It was found that AUC on test set is 0.7 which is 0.06 lower than the validation AUC. In order to find out why this is happening, two attempts, listed below, were tried out by adjusting validation and test splits.

- Switching validation and test set: Experiments executed till now were repeated to find the best model on the new validation set. It was observed that average AUC of all experiments is 0.678 on the new validation set which was 0.719 on the original validation set. Furthermore, the new best model which is consisted of RandomForestClassifier, KNeighborsRegressor and BorderlineSMOTE techniques, had 0.727 AUC on the new validation set and 0.751 on the new test set.
- Reshuffling dataset and generate splits again: Again, experiments were repeated as in the previous item. Best model this time was consisted of LinearSVC as classifier, RandomOverSampling as class imbalance technique and BayesianRidge as algorithm used in missing value imputation. This model had validation AUC of 0.749 and test AUC of 0.711. Average AUC on all experiments on the new validation set was 0.698.

It must be noted that Validation and Test sets were adjusted only for this part and original sets

were retrieved after this was over. In table 8.8 and 8.9, summarized results of this part can be found.

To conclude this stage, RandomForestClassifier was chosen as the classifier, KNeighborsRegressor was chosen as the algorithm used for missing value imputation and Random Over Sampling technique was chosen to deal with the class imbalance problem. The original validation set is relatively easier to predict. On the contrary, samples in the original test set are more difficult to predict. This is important to know for a fair evaluation of test results as the test scores depend highly on the test splits.

Splits	Max AUC	Average AUC
Original	0.761	0.719
Validation and Test switched	0.727	0.678
Reshuffled Dataset	0.749	0.698

Table 8.8: Overview of the validation results of the investigation why test score is significantly lower than validation score

Splits	Val AUC	Test AUC
Original	0.761	0.7
Validation and Test switched	0.727	0.751
Reshuffled Dataset	0.749	0.711

Table 8.9: Best model results from each attempt on investigating why test score is significantly lower than validation score

8.1.2 Structured Stage 2

In order to achieve better results with selected algorithms, it is usually useful to optimize the hyperparameters. This stage is concerned particularly with finding the best hyperparameters for the algorithm used in missing value imputation, namely, KNeighborsRegressor and iterative imputer function. In this context, There are two main hyperparameters to investigate:

- **Maximum number of iterations**, i.e. how many times the whole dataset has to be iterated to impute missing variables unless it is converged
- **The value for K**, i.e how many neighbor samples to use when calculating the missing variable for a sample

Till this stage, all the experiments were run with 20 maximum number iterations. However, the general situation was that, a convergence warning was received at the time when maximum number of iterations was reached, indicating that the stopping criteria was not yet below the tolerated rate. Tolerance was the default value which is 0.001. In order to overcome the convergence issue, an attempt to increase the maximum number of iterations was tried out. However, as in table 8.10 it decreased the AUC score from 0.761 to 0.731 and still the function could not converge. Therefore it was decided to keep the maximum number of iterations for the iterative imputer as 20. By the time of this project is executed, IterativeImputer function of sklearn library is still experimental, convergence problem might be due to this phase and might be mitigated in

the future versions. Regarding the k value of the KNeighborsRegressor, table 8.11 shows that best value was found as 10 neighbors.

Max Number of Iterations	Max AUC	Average AUC	Count of Runs
20	0.761	0.76	5
30	0.731	0.729	5

Table 8.10: Results regarding the search for maximum number of iterations of Iterative Imputer function

Number of Neighbors	Max AUC	Average AUC	Count of Runs
10	0.788	0.770	5
20	0.761	0.749	5
5	0.757	0.749	5

Table 8.11: Results regarding the search for number of neighbors of KNeighborsRegressor function

8.1.3 Structured Stage 3

In this stage, a grid-search was executed to find optimized hyperparameters for the Random Forest Classifier. Initially this search was validated on the validation set. However, this resulted in models that are extremely overfitting the validation set and showing drastically worse results on the test set. The reason for overfitting was that the model which has the highest validation AUC score has hyperparameters such that they work well with the validation set's specific data distribution but not really generalizable.

The author will first present the grid-search experiments which are validated with validation set in section 8.1.3.1 and then cross-validated experiments will be in section 8.1.3.2. The format of this stage is that, hyperparameter candidates which are given to grid-search and best hyperparameters are shown in the tables where the best score is noted in the caption of the tables.

8.1.3.1 Normal Validation Part

On each search, insights from the previous search were used to create candidates. In the end of each grid search, winners were used to narrow down the search range. Basically, the author created new arbitrary candidates which are closer to the winner of the previous search. This allowed to make a more detailed search in more relevant range without increasing the computation time significantly. By this way, a smarter search was achieved. At the point, where AUC score does not increase anymore, 4th grid-search to be precise, experiments stopped and model was tested on the test set. The model which had 0.803 AUC on the validation set, scored 0.67 AUC on the test set which is even worse than the educated baseline model which was built in Structured Stage 1. Therefore, it was decided to switch to cross validation to have a model which has a better generalizability.

Grid-Search - 1 - Best Model AUC 0.784

Hyperparameter	Candidates	Best Value
Number of trees	100, 200, 300, 400, 500, 1000, 1500	100
Criterion	Gini, Entropy	Gini
Maximum depth of trees	1, 3, 5, 7, 9, 10, 20, 30, 40, 50 ,60 ,70, 80, No limit	10
Minimum samples required to be a leaf	4, 8, 12, 16	8
Minimum samples required to split a node	4, 8, 12, 16	16
Maximum number of features to consider when splitting	$\sqrt{n\_features}$, $\log_2(n\_features)$, No limit	$\log_2(n\_features)$
Maximum leaf nodes	10, 30, 50, 70, 90, No limit	No limit
Enable bootstrap	True, False	False

Table 8.12: Hyperparameters optimization grid-search-1

Grid-Search - 2 - Best Model AUC 0.79

Hyperparameter	Candidates	Best Value
Number of trees	50, 100, 150, 200, 250, 300, 350, 400, 450	50
Criterion	Gini, Entropy	Gini
Maximum depth of trees	10, 12, 14, 16, 18	10
Minimum samples required to be a leaf	2, 4 ,6, 8	2
Minimum samples required to split a node	12, 16	12
Maximum number of features to consider when splitting	$\log_2(n\_features)$	$\log_2(n\_features)$
Maximum leaf nodes	10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95, No limit	60
Enable bootstrap	True, False	False

Table 8.13: Hyperparameters optimization grid-search-2

Grid-Search - 3 - Best Model AUC 0.803

Hyperparameter	Candidates	Best Value
Number of trees	10, 20, 30, 40, 50, 60, 70, 80, 90	50
Criterion	Gini	Gini
Maximum depth of trees	5, 6, 7, 8, 9, 10, 11, 12, 13, 14	12
Minimum samples required to be a leaf	1, 2, 3, 4	4
Minimum samples required to split a node	12, 24, 36	24
Maximum number of features to consider when splitting	$\log_2(n\_features)$	$\log_2(n\_features)$
Maximum leaf nodes	10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95, No limit	No limit
Enable bootstrap	True, False	True

Table 8.14: Hyperparameters optimization grid-search-3

Grid-Search - 4 - Best Model AUC 0.803

Hyperparameter	Candidates	Best Value
Number of trees	50, 55, 60, 65, 70, 75, 80	60
Criterion	Gini	Gini
Maximum depth of trees	10, 11, 12, 13, 14	12
Minimum samples required to be a leaf	2, 3, 4, 5, 6, 7	4
Minimum samples required to split a node	20, 22, 24, 26, 28	24
Maximum number of features to consider when splitting	$\log_2(n\_features)$	$\log_2(n\_features)$
Maximum leaf nodes	10, 20, 30, 40, 50, 60, 70, 80, 90, No limit	No limit
Enable bootstrap	True, False	True

Table 8.15: Hyperparameters optimization grid-search-4

8.1.3.2 Cross-Validation Part

In this part of the experiments, train and validation sets were merged together. Then, they were feed into 5-fold stratified cross-validation, resulting in 5 different splits of training and validation sets. An important detail of this part is that, on each split, missing value imputation for training and validation set and random over sampling for training set had to be applied.

The cross-validated hyperparameter search was run twice. Firstly, on the complete dataset then on the small dataset (mentioned in section 6.1). According to the results of this section, small dataset can achieve better AUC scores. Best hyperparameters for each dataset obtained in this stage will be used in the remainder of the thesis.

Cross-Validated Grid-Search on the complete dataset, average AUC achieved in this stage by the optimized hyperparameters is 0.755 which was 0.7 before the optimization

Hyperparameter	Candidates	Best Value
Number of trees	10, 20, 30, 40, 50, 60, 70, 80, 90, 200, 300, 500	70
Criterion	Gini	Gini
Maximum depth of trees	5, 6, 7, 8 ,9, 10, 11, 12, 13, 14	12
Minimum samples required to be a leaf	2, 3, 4, 5, 6, 7	6
Minimum samples required to split a node	12, 24, 36	24
Maximum number of features to consider when splitting	$\log_2(n\_features)$	$\log_2(n\_features)$
Maximum leaf nodes	20, 40, 60, 80, No limit	No limit
Enable bootstrap	True, False	False

Table 8.16: Cross-Validated Hyperparameters optimization with Grid-Search on the Complete Dataset

Split 0 Validation AUC	0.77
Split 1 Validation AUC	0.672
Split 2 Validation AUC	0.792
Split 3 Validation AUC	0.738
Split 4 Validation AUC	0.802
Standard Deviation AUC	0.047
Mean AUC	0.755
Mean AUC Before Optimization	0.7

Table 8.17: Results of Best Model from Cross-Validated Hyperparameters optimization with Grid-Search on the **Complete** Dataset

Cross-Validated-Grid-Search on the Small Dataset - Best Model Average AUC 0.788, Average AUC before optimization 0.766

Hyperparameter	Candidates	Best Value
Number of trees	10, 20, 30, 40, 50, 60, 70, 80, 90, 200, 300, 500	300
Criterion	Gini	Gini
Maximum depth of trees	5, 6, 7, 8 ,9, 10, 11, 12, 13, 14	13
Minimum samples required to be a leaf	2, 3, 4, 5, 6, 7	4
Minimum samples required to split a node	12, 24, 36	12
Maximum number of features to consider when splitting	$\log_2(n\_features)$	$\log_2(n\_features)$
Maximum leaf nodes	20, 40, 60, 80, No limit	80
Enable bootstrap	True, False	True

Table 8.18: Cross-Validated Hyperparameters optimization with Grid-Search on the **Small** Dataset

Split 0 Validation AUC	0.742
Split 1 Validation AUC	0.813
Split 2 Validation AUC	0.807
Split 3 Validation AUC	0.723
Split 4 Validation AUC	0.856
Standard Deviation AUC	0.049
Mean AUC	0.788
Mean AUC Before Optimization	0.766

Table 8.19: Results of Best Model from Cross-Validated Hyperparameters optimization with Grid-Search on the **Small** Dataset

8.2 Image Modality

As explained in section 7.1, experiments with only image modality have two types. First type uses single image series and second type uses both of the image series. Image modality experiments started with the first type. Then, based on the results, second type of experiments were executed. First type of experiments were executed to find best feature extraction method for each series. All the experiments in image modality were done with deep learning models. X-ray images used in Chest and Hip series are AP views. All the experiments were done with 30-days mortality labels. Before explaining different candidate models, the author will first give the general settings and hyperparameters used in all deep learning experiments and data augmentation settings for image data generation in table 8.20 and 8.21 respectively.

Setting/Hyperparameter	Value
Number of Epochs	100
Loss Function	Binary Cross Entropy
Optimizer	Adam
Learning Rate	0.001
Monitored Value for Early Stopping	Validation AUC
Patience for Early Stopping	20
Batch Size	20
Minimum delta required for improvement	0.0001
Monitored Value for Reducing Learning Rate	Validation AUC
Patience for Reducing Learning Rate	5
Factor by which the learning rate will be reduced	0.05
Weight for class 0(negative)	0.54
Weight for class 1(positive)	6.29

Table 8.20: Settings/Hyperparameters used in all Deep learning experiments

Data Augmentation Setting	Value
Rotation Range	20
Width Shift Range	0.2
Height Shift Range	0.2
Shear Range	0.2
Zoom Range	0.2
Channel Shift Range	10
Horizontal Flip	True
Vertical Flip	True
Fill mode	Nearest
Interpolation(also applies to validation & test)	Bicubic

Table 8.21: Data augmentation settings used in all Deep learning experiments for image data generation during training phase

In the scope of this study, 4 pre-trained convolutional neural network models on imagenet were included as candidates for feature extraction task from image modality. These models are DenseNet169 [26], ResNet152 [57], InceptionV3 [28], Xception [58]. These pre-trained models were obtained from Keras library. They come with an optional fully connected layer as the last layer however this is not useful for our task as the fully connected layer is used for classifying imagenet objects. Consequently this optional part was not retrieved. Instead of that, after the convolutional part ends on each model, 3 new fully connected layers are attached as the classifier part of the network. These 3 layers are shown in table 8.22. Hyperparameters which are not mentioned here, used as default from Keras library (e.g., kernel initializer = glorot uniform). During the training phase, there were two approaches followed regarding fine tuning the transfer learning model, namely, full training and partial training. In full training, all of the layers of a network is trained, in partial training, some of the convolutional part is freezed and remaining part of the network is trained. This is motivated by the fact that early convolutional layers capture more generic information where deeper convolutional layers have more context related specific information. To this end, each model was analyzed based on their size and architecture. Information of which layers of pre-trained models were frozen can be found in table 8.23, The column ”Partial Training Start Layer” refers to the split point of network, layers before this layer are frozen during partial training and layers after it are trainable. Mentioned blocks in column

”Trained Block Correspondence” refer to the blocks defined by pre-trained model developers.

Layer Name	Number of Neurons	Activation Function	Bias Initializer
Dense_1	1024	Relu	zeros
Dense_2	16	Relu	zeros
Output	1	Sigmoid	-2.45

Table 8.22: Fully Connected Layers which are added after the convolutional part of pre-trained models

Model Name	Total Number of Layers	Partial Training Start Layer	Trained Block Correspondence
DenseNet169	596	369	Last Convolutional Block
ResNet152	516	483	Last Convolutional Block
InceptionV3	312	249	Last 2 Convolutional Blocks
Xception	133	116	Last 2 Convolutional Blocks

Table 8.23: Information on layers of pre-trained models

8.2.1 Single Image Series

As experimental settings for image modality are now described, the author will present the results of the experiments with chest x-ray images and hip x-ray images in tables 8.24 and 8.25 respectively. It can be clearly seen that, Xception model with full training achieved the highest AUC of 0.701 on chest x-ray images where ResNet152 model with partial training performed best on hip x-ray images with 0.606 AUC.

Model Name	Training Mode	AUC Score
DenseNet169	Full	0.669
InceptionV3	Full	0.547
ResNet152	Full	0.5
Xception	Full	0.701
DenseNet169	Partial	0.5
InceptionV3	Partial	0.5
ResNet152	Partial	0.631
Xception	Partial	0.5

Table 8.24: Experiment results with Chest x-ray images

Model Name	Training Mode	AUC Score
DenseNet169	Full	0.588
InceptionV3	Full	0.57
ResNet152	Full	0.5
Xception	Full	0.562
DenseNet169	Partial	0.5
InceptionV3	Partial	0.498
ResNet152	Partial	0.606
Xception	Partial	0.532

Table 8.25: Experiment results with Hip x-ray images

After this stage, the best model on each image series tried to be improved further by training again with different methods. These different methods include dropout layers, regularizers and continuation. Experiment results including these methods are illustrated in table 8.26 for chest x-ray images and table 8.27 for hip x-ray images. The model for chest could not be improved more, however, AUC of hip model improved slightly from 0.606 to 614.

- Dropout Layers: In the classifier part, a dropout layer with a rate of 0.25, added after Dense_1 and Dense_2 layers.
- Regularizers: In the classifier part, L2 regularizers with a regularization rate of 0.001, added on Dense_1 and Dense_2 layers.
- Continuation: Continued training from the weights which achieved highest AUC on the task.

Model Name	Training Mode	Methods Used	AUC Score
Xception	Full	Continuation	0.693
Xception	Full	Dropout	0.612
Xception	Full	Dropout&Regularizer	0.522
Xception	Full	Dropout& Regularizer& Continuation	0.641

Table 8.26: Experiment results with improvement methods on Chest x-ray images

Model Name	Training Mode	Methods Used	AUC Score
ResNet152	Partial	Continuation	0.614
ResNet152	Partial	Dropout	0.568
ResNet152	Partial	Dropout& Regularizer	0.5
ResNet152	Partial	Dropout& Regularizer& Continuation	0.585

Table 8.27: Experiment results with improvement methods on Hip x-ray images

As the last experiment with single image series, models that are identified as best for their correspondings tasks trained again with Small Dataset in order to allow multimodal experiments also with Small Dataset. Xception model with full training achieved 0.642 AUC on chest series. On the other hand, ResNet152 model with partial training achieved 0.632 AUC on hip series. No further improvement was tried out on this stage.

8.2.2 Dual Image Series

Architectures of experiments with dual image series based heavily on the results of experiments with single image series. Therefore, Xception with full training is used for the feature extraction part of chest series and ResNet152 with partial training used for the feature extraction of hip series. The difference was that backpropagation was done simultaneously for both of the feature extractors from the same loss function. Concatenation of two models were done on "Dense_2" layer of each single image series models as shown in fig 7.3. Then, the same output layer with a sigmoid function added in the end. As in the single image series experiments, similar improvement methods applied here. Regarding the continuation method, weights were retrieved from the best models of single image series experiments to corresponding feature extractors. Results of experiments with dual image series are represented in table 8.28. None of the experiments succeded to outperform the Xception model which is using only chest series.

Models Used	Training Mode	Methods Used	AUC Score
Xception & ResNet152	Full & Partial	-	0.633
Xception & ResNet152	Full & Partial	Continuation	0.669
Xception & ResNet152	Full & Partial	Dropout & Continuation	0.697
Xception & ResNet152	Full & Partial	Dropout& Regularizer& Continuation	0.596

Table 8.28: Experiment results with both image series (chest & hip)

8.3 Multimodality

In multimodal experiments, there are early fusion, late fusion and neural network approaches. These approaches also have subdivisions where image series are trained separately and simultaneously. There are also experiments that use only single image series from image modality. The architectures of these approaches are illustrated in figures from 7.4 to 7.9. Early fusion and late fusion are model agnostic approaches, therefore, extracted features are merged with structured modality to create the new structured modality. Missing value imputation (iterative imputation with KNeighborsRegressor) and class imbalance handling (random over sampling) techniques were applied to the new structured modality and then finally fed into the model which performed best in structured modality section, namely, Random Forest Classifier. As the last experiments of structured modality, optimized hyperparameters were found for the Random Forest model with cross validation. Same hyperparameters were used again on multimodal experiments however this time validation was done with original validation set. Model-agnostic experiments run 5 times with varying seed values for the random number generation. By that, it was aimed to add randomization to experiments as it was not possible to do cross validation at this stage. Results for model agnostic approaches are in table 8.29. Eventhough, early fusion with separate feature extraction has the best performance, it was observed that, some of the neuron outputs used as findings from image modality are 0 for all of the patients. This could be a sign for further improvement opportunity of the model however, that will not be in the scope of this thesis.

Multimodal learning approach	Feature Extraction	Max AUC	Average AUC	Count of Runs
Only Structured Modality	-	0.775	0.751	5
Late Fusion	Separate	0.775	0.761	5
Early Fusion	Separate	0.811	0.801	5
Late Fusion	Simultaneous	0.794	0.774	5
Early Fusion	Simultaneous	0.783	0.775	5

Table 8.29: Multimodal experiment results with model agnostic approaches. Separate and simultaneous feature extraction refers to the training style of image modality as defined in section 7.1

A model based approach, namely, multimodal fusion with neural networks was also tried out for this task. Structured modality as an input layer was concatenated to Dense_2 layers of feature extractors as shown in figures 7.9 and 7.8. However, a couple more fully connected layers were attached after the concatenation. Information on layers after the concatenation layer can be found in table 8.30. As deep learning is once again employed, improvement methods from section 8.2 were used again. This was done mainly due to the fact that an overfitting pattern was observed, meaning that, training AUC was much higher than validation AUC. Experiment results of multimodal fusion with neural networks can be seen in table 8.31. It appears to be that multimodal fusion with neural network approach cannot outperform early fusion technique in this study. All experiments presented in this table used both structured modality and image modality. If "Models Used" column has Xception, this means, chest series were used from image modality, if it has ResNet152 then, hip series were used. If it has both of them, then both of the image series were used in the model. For the models who does not use continuation method, imagenet weights were used again as a starting point.

Layer Name	Number of Neurons	Activation Function	Bias Initializer
Dense_3	64	Relu	zeros
Dense_4	32	Relu	zeros
Dense_5	16	Relu	zeros
Dense_6	8	Relu	zeros
Output	1	Sigmoid	-2.45

Table 8.30: Information on layers after the concatenation of structured modality and image modality

Models Used	Training Mode	Methods Used	AUC Score
Xception(Chest) & ResNet152(Hip)	Full & Partial	-	0.689
Xception(Chest) & ResNet152(Hip)	Full & Partial	Continuation	0.735
Xception(Chest) & ResNet152(Hip)	Full & Partial	Dropout & Continuation	0.576
Xception(Chest) & ResNet152(Hip)	Full & Partial	Dropout& Regularizer& Continuation	0.599
ResNet152(Hip)	Partial	-	0.727
Xception(Chest)	Full	-	0.756

Table 8.31: Experiment results of multimodal fusion with neural networks. All experiments used structured modality beside the image series mentioned on each row.

8.4 Testing

Since validation phase of models had been completed. There was no point of keeping a separate validation set. Early fusion was validated as the best technique to fuse multiple modalities. From this point only, RandomForestClassifier will be trained and therefore there was no necessity for an early stopping set either. Therefore, the train set was redefined for this section as:

$$train_{new} = train_{old} + validation_{old} \tag{8.1}$$

Validated models from the previous stage were trained on these new train sets for both for the complete dataset and small dataset and their performance on the test sets is presented in table 8.32. Almelo Hip Fracture Score (AFHS), as a previous successful study with the same research question, was also used in order to have a benchmark. It has been reported that AHFS reached 0.82 AUC in [6]. However, due to the lack of some variables used in AHFS, the same version could not be replicated on the dataset of this study. Therefore, it will be called as AHFS-a(adjusted version). Variables used in this adjusted version are as follows: Age, Gender, CCI score, Prone to delirium, memory problems, KATZ ADL Score, ASA score, Pre-fracture living situation, Pre-fracture mobility, Cancer, HB, Prone to under-nutrition, Unintentional loss of weight, Decreased appetite, Drink or tube feeding, SNAQ Score. Furthermore, no class imbalance handling was applied to the AHFS-a model and missing value imputation is done by filling with mean values of the variables. The classifier used in AHFS-a is the Logistic regression method from sklearn library, with solver ”sag” and without any penalty for regularization.

Model	Dataset	AUC Score
Early Fusion	Complete	0.742
AHFS-a	Complete	0.706
Early Fusion	Small	0.769
AHFS-a	Small	0.729

Table 8.32: Test results of complete dataset and small dataset benchmarking with AHFS-a

Early fusion on complete dataset is considered to be the main product of this study. Therefore, the author will now describe this model further. Other performance metrics than AUC with the default decision threshold which is 0.5, are represented in table 8.33. Reader is advised to review section 2.6 to get an understanding of the mentioned metrics. These scores are all subject to change once the decision threshold is adjusted. One can do that by taking the ROC curve as reference which is shown in figure 8.1. Although deciding on decision threshold part was left out for the users of the model in real life, the author would like to show an example how model reacts to different decision thresholds. To this end, the author sets the decision threshold to 0.276 based on the ROC curve in figure 8.1. Results with the adjusted threshold are shown in table 8.34.

Metric	Value
Recall score	0.0408
Precision score	0.222
Specificity score	0.987
F1 score	0.069
Accuracy score	0.911
Area under the roc curve (AUC)	0.742

Table 8.33: Performance metrics with default decision threshold 0.5

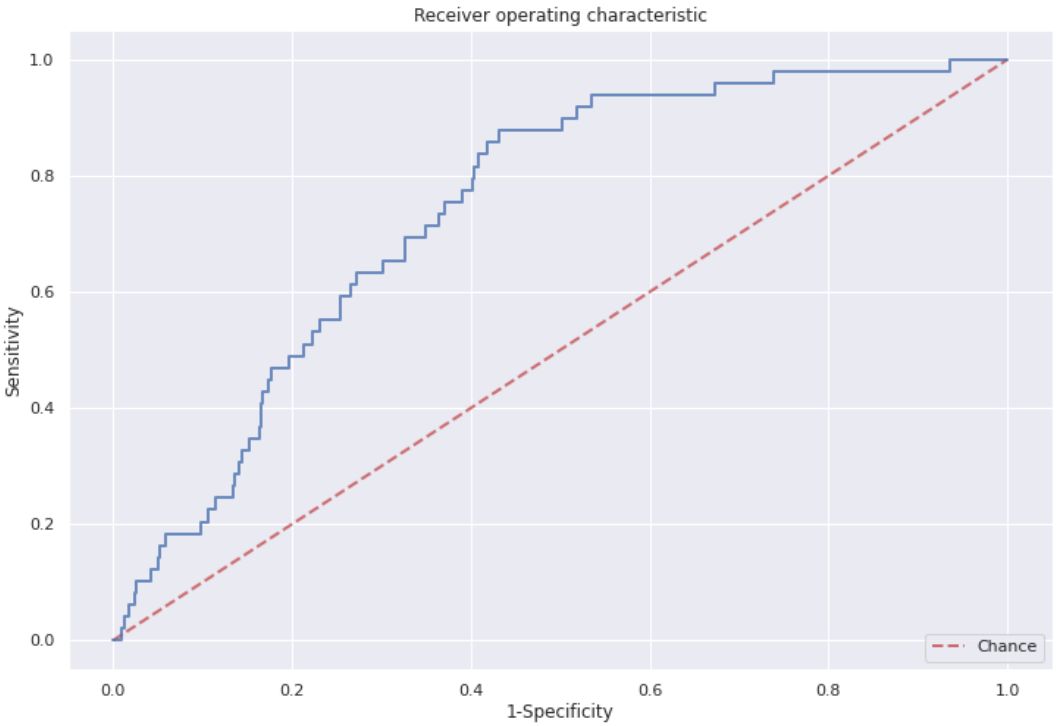


Figure 8.1: ROC curve of the main model, early fusion on complete dataset

Metric	Value
Recall score	0.49
Precision score	0.18
Specificity score	0.804
F1 score	0.264
Accuracy score	0.778
Area under the roc curve (AUC)	0.742

Table 8.34: Performance metrics with adjusted decision threshold 0.276

Moreover, the most important 20 features according to the main model are shown in the represented in figure 8.2 and the complete feature importance information can be found in tables 8.35,8.36 and 8.37. Variables called ”chest finding” and ”hip finding” refer to the extracted features from corresponding x-rays. Feature importances were calculated by the feature importance function of RandomForestClassifier, which measures features’ contribution in decreasing the impurity.

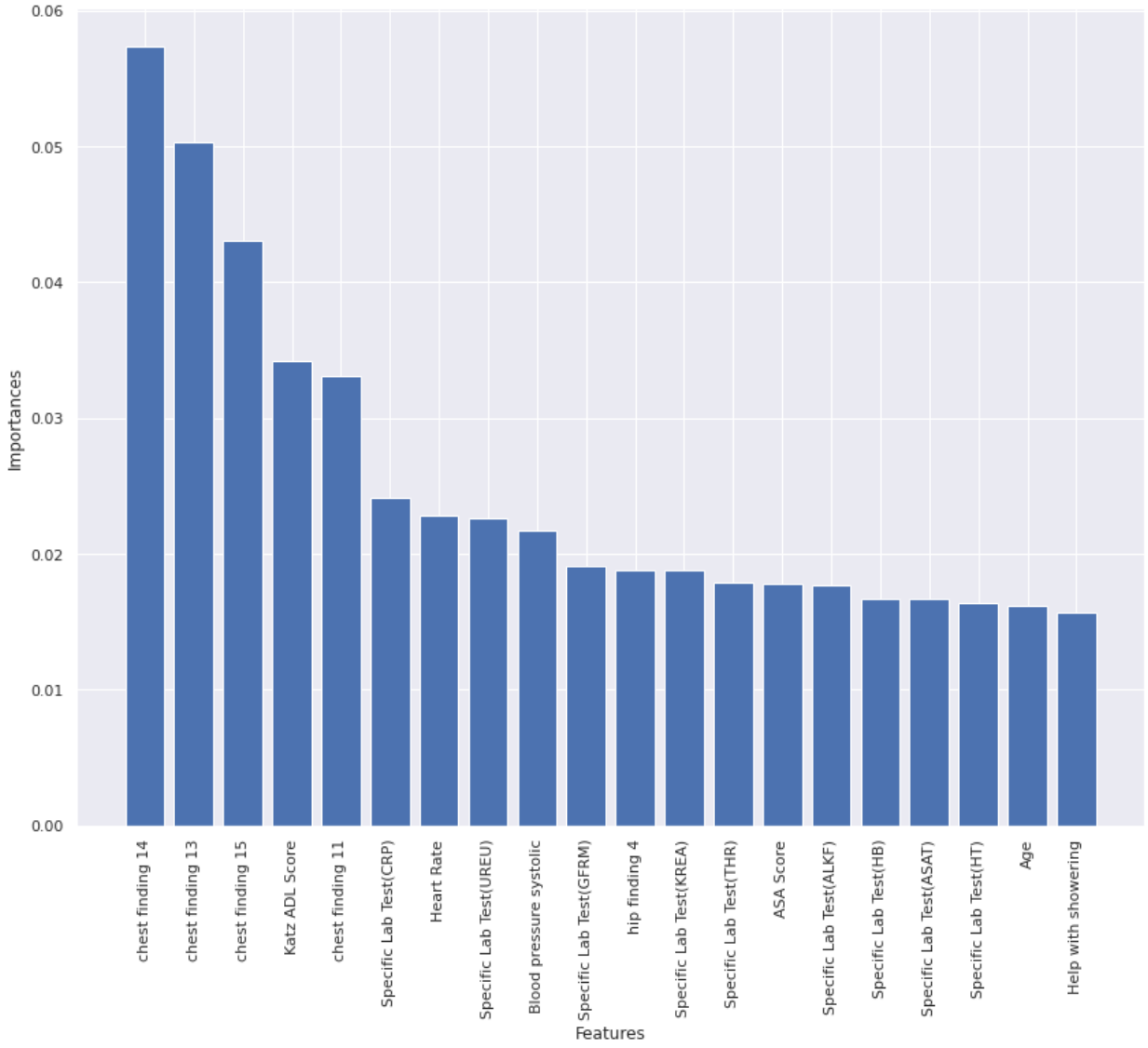


Figure 8.2: Top 20 Most Important Features of the main model

Feature	Importance
chest finding 14	0.057
chest finding 13	0.05
chest finding 15	0.043
Katz ADL Score	0.034
chest finding 11	0.033
Specific Lab Test(CRP)	0.024
Heart Rate	0.023
Specific Lab Test(UREU)	0.023
Blood pressure systolic	0.022
Specific Lab Test(GFRM)	0.019
hip finding 4	0.019
Specific Lab Test(KREA)	0.019
Specific Lab Test(THR)	0.018
ASA Score	0.018
Specific Lab Test(ALKF)	0.018
Specific Lab Test(HB)	0.017
Specific Lab Test(ASAT)	0.017
Specific Lab Test(HT)	0.016
Age	0.016
Help with showering	0.016
SNAQ Score	0.016
Width of QRS complex in ECG	0.016
Specific Lab Test(LDH1)	0.016
Help to dress	0.015
Blood pressure diagstolic	0.015
Specific Lab Test(GLUCGLUC)	0.014
Help with transfer from bed to chair	0.014
Help with self-care last 24 hours	0.014
Heart axis orientation in ECG	0.014
Specific Lab Test(XKA)	0.013
Specific Lab Test(LEUC)	0.013
Specific Lab Test(NA)	0.013
hip finding 12	0.013
Specific Lab Test(ALAT)	0.012
Specific Lab Test(GGT)	0.012
Respiration Parameter	0.012
CCI Score	0.012
Memory Problems	0.011
Medications(C07)	0.011
Medications(N05)	0.011
Prone to Delirium	0.011
hip finding 10	0.011
Fracture laterality is right	0.01
Unintentional loss of weight	0.01
Previous confusional state	0.01

Table 8.35: Complete table of feature importances according to the main model(Part 1).

Feature	Importance
Help with going to toilet	0.009
Incontinence Material Used	0.008
Medications(C03)	0.008
Medications(B01)	0.007
Prone to under-nutrition	0.007
Specific Lab Test(BLGR) is 0	0.007
fall risk	0.007
Use of Bloodthinners	0.007
pre-fracture living situation - nursing care home	0.006
Medications(C10)	0.006
Medications(C09)	0.006
Medications(M01)	0.006
Medications(A10)	0.005
Decreased appetite	0.005
Gender Male	0.005
Medications(B02)	0.004
Medications(R03)	0.004
pre-fracture living situation - independent	0.004
Binary SNAQ Score	0.004
Medications(C08)	0.004
Gender Female	0.004
Medications(A02)	0.004
Help with eating	0.004
pre-fracture mobility - mobile without tools	0.003
Medications(C01)	0.003
CCI Comorbidities (CHF)	0.003
drink or tube feeding	0.003
Fracture Type - mediale collumfracturen	0.003
Fall in last 6 months	0.003
Medications(B03)	0.003
Type of Surgery - Internal fixation intertrochanteric and subtrochanteric femur fracture	0.002
Fracture Type - pertrochantaire fractures	0.002
pre-fracture living situation - independent with help	0.002
CCI Comorbidities (MSRD)	0.002
pre-fracture living situation - retirement home	0.002
CCI Comorbidities (DIACC)	0.002
CCI Comorbidities (CVD)	0.002
CCI Comorbidities (CVE)	0.002
chest finding 7	0.002
Type of Surgery - Internal fixation for femoral neck fracture	0.002
Type of Surgery - Endoprosthesis for femoral neck fracture	0.002
pre-fracture mobility - mobile outdoor with 1 tool	0.001
Medications(L04)	0.001
CCI Comorbidities (DEM)	0.001
pre-fracture mobility - mobile outdoor with 2 tools or frame(e.g. rollator)	0.001

Table 8.36: Complete table of feature importances according to the main model(Part 2).

Feature	Importance
CCI Comorbidities (CAN)	0.001
CCI Comorbidities (RD)	0.001
pre-fracture mobility - mobile indoor, never out without help	0.001
Type of Surgery - Other	0.001
CCI Comorbidities (CPD)	0.001
CCI Comorbidities (MI)	0
hip finding 2	0
CCI Comorbidities (PVD)	0
CCI Comorbidities (DIA)	0
CCI Comorbidities (MLD)	0
pre-fracture mobility - no functional mobility	0
Specific Lab Test(IRAI) is positive	0
Fracture Type - subtrochantaire femur	0
CCI Comorbidities (PUD)	0
CCI Comorbidities (LEU)	0
CCI Comorbidities (LYM)	0
CCI Comorbidities (MSLD)	0
pre-fracture living situation - others	0
chest finding 0	0
chest finding 1	0
chest finding 2	0
chest finding 3	0
chest finding 4	0
chest finding 5	0
chest finding 6	0
chest finding 8	0
chest finding 9	0
chest finding 10	0
chest finding 12	0
hip finding 0	0
hip finding 1	0
hip finding 3	0
hip finding 5	0
hip finding 6	0
hip finding 7	0
hip finding 8	0
hip finding 9	0
hip finding 11	0
hip finding 13	0
hip finding 14	0
hip finding 15	0

Table 8.37: Complete table of feature importances according to the main model(Part 3).

9 DISCUSSION, FUTURE WORK AND LIMITATIONS

9.1 Discussion and Future Work

In this section, the author will discuss the results gathered in the experimenting stage. This part will go with the same chronological order of experiments but provide a more general overview of results and their interpretation. Future work is integrated into the discussion of relevant parts.

9.1.1 Selection of Classifier, Class Imbalance, and Missing Value Imputation Technique

Although the first and second stages of structured modality experiments did not use cross-validation, they were run multiple times. The decisions were made based on average outcomes. This helped with not choosing a technique that has a really high score in one experiment just by luck. As the traditional machine learning algorithm for the classification task, Random Forest Classifier was chosen as it achieved the highest average validation AUC 0.761. As the function to use in iterative imputation, KNeighborsRegressor was found to be the most convenient technique with the highest average validation AUC 0.749. In general, the improvement by class imbalance techniques are not much higher than no handling. Therefore, it might also be acceptable to not apply any technique to handle the class imbalance. Even though different sampling techniques had similar average results, Random Over Sampling was chosen to deal with class imbalance due to its higher compatibility with the Random Forest Classifier as it achieved the best validation AUC 0.761. Moreover, this class imbalance handling technique, by its nature, is simpler to implement.

9.1.2 Investigation of the Test Set

After the discovery of the best combination of techniques regarding, missing value imputation, class imbalance handling, and classification, the model was applied on the test set as an educated baseline model and it was observed that AUC dropped to 0.7 which was 0.761 on the validation set. This decline was further investigated with two attempts, including switching validation and test sets, and reshuffling dataset and generating new train, validation, and test splits. This investigation showed that the original validation set is much easier to predict than the original test set. Moreover, this particular test set is in fact quite difficult to predict in general. This finding will be mentioned later in this chapter while discussing the final test results. To conclude, changing the test set can result in different outcomes as test results are highly dependent on the test set. Generally, this could be avoided by running with multiple test sets and average the results to decrease the error. However, this was not possible in the due to architecture of this study. The model training for different modalities is done separately. Once the output of a

model with one modality is used on the training of other modality which is the case during Early Fusion, there could be information leaks if the test set is not fixed.

9.1.3 Optimizing Hyperparameters

Before going through the third stage of structured modality experiments, one should note that there were two datasets used in this study. The first dataset(2404 samples) is in fact the complete dataset with all patients throughout the study period. The second dataset(1654 samples), which is called as the small dataset, includes patients only after 01-04-2012, thus resulting in a dataset with much less missing values. Therefore, the test sets of these two datasets are different than each other. It should be acknowledged that test results depend highly on the split. Most of the validation experiments were undertaken with the complete dataset. The second dataset, however, used only after validating most of the techniques in the methodology except hyperparameter optimization of the classifier. This was due to the fact that different hyperparameters might optimize a model which is fed with fewer data. On the third stage of structured modality experiments, a couple of grid-searches took place to optimize the hyperparameters of the Random Forest Classifier. Grid-search experiments were first validated on the normal validation set however, after applying the validated model on the test set, it was found that hyperparameter optimization overfits the validation set because AUC on the test was 0.67 whereas it was 0.80 on validation set which is an unacceptable decrease. Therefore, the last grid-search was repeated with cross-validation in order to avoid overfitting on the validation set. Cross-validation results were evaluated based on their average AUC scores. The best model achieved Average AUC of 0.755. For the sake of benchmarking, the model before the hyperparameter optimization was cross-validated as well, it could achieve Average AUC of 0.70. This showed that hyperparameter optimization improved the AUC remarkably by 0.055. Subsequently, the cross-validated grid-search was applied to the small dataset as well. The mean AUC score before optimizing hyperparameters was 0.766, after the grid-search, the mean AUC score increased to 0.788. It should be noted that cross-validated grid-search improved both models although resulting optimized hyperparameters for two datasets were different. One of the biggest difference in optimized hyperparameters were the number of trees used in the random forest model. The model which trained with the complete dataset had 70 trees in the forest, whereas the model trained on the small dataset had 300 trees. In general, a higher number of trees help Random Forest models to generalize better and avoid overfitting. Considering that the small dataset has a smaller sample size, the fact that it requires more trees to generalize better makes sense. After exploring the optimized hyperparameters, all experiments regarding structured modality was over.

9.1.4 Image Modality

After finishing structured modality, experiments on image modality were executed in order to find signs related to 30-days mortality. First of all, correct images had to be selected for image modality experiments, to this end, a side-project had to be executed which built a model to discriminate images with different views or different body parts. Details on this side project can be found in appendix A. This model could also be used for more general tasks within the hospital. After finalizing the image dataset, 30-days mortality modeling started. Transfer learning was used in the convolutional part of neural networks for that purpose. By means of fine-tuning, a couple of different configurations were tried out. Even though this study is concerned with hip fractures, image modality experiments showed that there is more predictive

power in chest x-ray images compared to hip x-ray images. By the use of chest x-ray images, a model could reach up to 0.701 AUC score on the validation set, whereas this was only 0.614 with hip x-ray images. Logistic Regression, on structured modality experiments, had a maximum AUC of 0.706 with the same validation set which is very close to what has been achieved with the chest x-ray model. This is an important finding from the clinical perspective as it brings evidence, however, signs found in chest x-rays, linked to 30-days mortality, are not yet directly transferable to humans within the scope of this thesis. Yet, the explanation should be followed up in future work. In the convolutional part of these models, Xception [58] model was the best performing on the chest with full training, and ResNet152 [57] was on the hip with partial training. Furthermore, including both image series in the same model and backpropagating at the same time did not really improve the AUC score, the maximum score achieved on that way was 0.697 on the validation set. However, when training the best model again on the small dataset, the highest AUC achieved on the validation set with chest x-ray images was 0.642. Therefore, it can be confidently claimed that training with more data helps the model significantly on image modality. This could be also in the structured modality experiments but the number of missings did not allow the model to train in the most successful way. Based on this, the author suggests that, as future work, decreasing the number of missings with the help of other modalities or sources could help to achieve better performance.

9.1.5 Multimodal Learning

After finalizing the validation of unimodal predictors, multimodal learning was applied in three different approaches, namely early fusion, late fusion, and neural networks. The most successful attempt was with early fusion. This approach had an average AUC score of 0.801 on the validation set. The early fusion model is in fact, the Random Forest model from the structured modality experiments that is using also the findings from image modality. These findings are the neuron outputs of the predecessor layer of the last layer of image modality models. In this version of early fusion, two image models trained separately from each other. Although the inclusion of these findings improved the average AUC of the model from 0.751 to 0.801 on the validation set, not all of the variables carried information. It was observed that some of the neuron outputs are constantly 0 for all patients. This shows that at least the classifier part of the image modality model is suboptimal. Thus, by simplifying the fully connected layer architecture, further improvements could be possible and should be investigated as future work.

9.1.6 Factors Affecting 30-Days Mortality

It is quite difficult to compare the variables found to be important in this study directly to the ones in the literature due to the nature of the study. The inclusion of image modality and a much higher number of variables in the prediction model for 30-days mortality of elderly hip fracture patients brought a different approach than the ones in the literature. None the less, as these findings from image modality improved the prediction performance when combined with structured modality, it would be very valuable to know what the findings actually are. Especially, in order to have it working on a daily health care system, an artificial intelligence model should be as explainable as possible. Currently, this is possible with the early fusion model to some extent. It is known by the feature importances function(see figure 8.2 and tables 8.35,8.36,8.37) of the random forest model that, chest findings which carry information are the most valuable variables of the random forest model, meaning that, they decrease the impurity most. Additionally, activities of daily living and lab tests contribute substantially to the prediction of 30-days

mortality, yet comorbidities extracted from diagnostic treatment codes did not show any significant contribution to this prediction task. On the other hand, ASA score and Age was in the top 20 most important features of the model developed in this study, the literature suggests that these variables also appear to be significant factors affecting 30-days mortality in elderly hip fracture patients. As future work, heat maps from images could be generated to find out which areas in the images contribute the most in findings. This would then let the users of the model see why the model is making a decision in one particular direction. Moreover, another future work could be to investigate the feature importances in another perspective, by setting a threshold and removing the variables with importance below that threshold and see if the model can perform as good.

9.1.7 Testing and Benchmark

Finally, the validated early fusion model was tested on the test set in order to show its performance on unseen data. As was already mentioned earlier in this chapter, the test split of the complete dataset is actually quite difficult to predict than usual. Therefore, the results reported on the test set could be slightly underestimating the performance of the model. In order to have a better benchmark, an adjusted version of AHFS(Almelo Hip Fracture Score) was replicated. It was reported that the AHFS model in [6] achieved AUC of 0.82. However, AHFS-a (Adjusted Almelo Hip Fracture Score) could reach only AUC of 0.706 on the test set of this study, where the early fusion model had AUC of 0.742. The same comparison was executed also on the small dataset, AHFS-a had AUC of 0.729 where the early fusion model had 0.769. From these results, one can see that the test split of the complete dataset is difficult as both models perform worse in the complete dataset. Moreover, the early fusion model outperformed the AHFS-a model on both datasets. It should be noted that missing value imputation by means of regression and random oversampling for class imbalance handling was not applied when building the AHFS-a model as they are part of the early fusion model. Missing values were imputed by mean values instead which is a much simpler technique. The ROC curve of the final model was provided as guidance for the future users of the model for setting the decision threshold. This adjustment would allow one to re-balance the trade-off between the number of correctly predicted positive samples and correctly predicted negative samples in the desired way.

9.2 Limitations

The limitations encountered in this study are listed as follows.

- According to the protocol, all patients admitted to Emergency Room with an acute hip fracture, should have full pelvis x-ray images with AP view, however, this protocol is not always followed due to the specific request of the surgeon. Therefore, for those patients that full pelvis x-ray is missing, one-sided (right or left) hip x-ray with AP view is used. This might have resulted in a drop in the quality of the dataset. If the full pelvis x-ray image for all patients were available, it could have a positive impact on the performance by improving the quality of features extracted from hip x-rays.
- Extracting comorbidities with diagnostic treatment codes, do not give us which year the disease occurred, so it is possible that irrelevant comorbidities are considered in the prediction. As an example, a patient could have a lung disease 10 years ago but now, they are fully recovered, this should not affect the 30-days mortality, at least as much as a

disease happened in the last year. But, this discrimination was not possible and therefore was a limitation of this study.

- Another limitation with the extraction of comorbidities from the diagnostic treatment codes is that, identification of cancer state. More precisely, cancer which is on metastasis and normal phase cannot be discriminated which is in fact a huge difference regarding the Charlson Comorbidity Index.

10 CONCLUSION

So far, series of experiments were done to find appropriate techniques for missing value imputation, handle class imbalance problems, extract features from image modality, and fuse image modality with structured modality to predict 30-days mortality of elderly hip fracture patients. 2 datasets were employed in development of models, the first dataset is the complete dataset whereas the second one (small dataset) excluded patients admitted before 2012-04-01 and consequently much less missing values. The author will conclude the study by presenting the clinical implications and answering the research questions.

10.1 Clinical Implications

Hip fractures are a major health care problem in society. This study presents a prediction model for 30-days mortality of elderly hip fracture patients by following a multimodal machine learning approach, in order to guide the decision-making process with respect to the treatment of the patient. This approach fuses the image modality with the structured modality for the prediction task. At the same time, it also addresses the problems related to the class imbalanced dataset and the high number of missing values. The proposed model outperforms a replicated version of the Almelo Hip Fracture Score (AHFS-a) with an AUC score of 0.742 vs 0.706. However, due to the fact that the test is particularly harder to predict, these scores might be underestimating the actual model performances. Although an AUC score of 0.742 is considered as acceptable discrimination, by itself, it is still not good enough to decide directly on which patients should be operated or not. Yet, it can be used as a tool to identify high-risk patients. Finally, by the analysis of feature importances, this study also demonstrates that chest x-ray images contain important signs related to 30-days mortality of the patients.

10.2 Research Questions-Answers

1. **To what extent, one can predict 30-days mortality of the elderly hip fracture patients after surgery using machine learning with pre-operative variables?**

   Test results showed that the early fusion model developed in this study can achieve AUC score of 0.742 on the complete dataset and 0.769 on the small dataset. Although the results are similar to what had been achieved in the literature, there is enough evidence to think that the score on the complete dataset might be underestimating the model performance. See discussion in chapter 9 to have a better overview of the benchmark of the model.

   (a) **With respect to the class imbalanced dataset, to what extent, class imbalance**

handling techniques are useful to preprocess the data for classification?

Oversampling techniques, namely Random Over-Sampling, SMOTE, Borderline-SMOTE, ADASYN improved the classification to a similar extent. However, Random Oversampling was chosen due to its compatibility with the chosen classifier. Adjusting class weights was used in deep learning. However,it showed poor results when applied with traditional machine learning algorithms. Additionally, the NearMiss-2 technique showed significantly poor results. Finally, the improvement by these techniques are not much higher than no handling. Therefore, it might also be acceptable to not apply any technique to handle the class imbalance.

(b) Due to the high amount of missing values, to what extent, the missing value imputation techniques are suitable in predicting 30-days mortality of the elderly hip fracture patients?

KNeighborsRegressor algorithm was found to be the best regression model to impute missing values. Although this technique performed significantly better than filling with mean values, the missing value imputation task was not successful as desired. Both validation and test results on the small dataset had better scores than the complete dataset. Thus, it can be concluded that Partial Listwise Deletion (creating small dataset) method performed better than missing value imputation.

(c) Which machine learning algorithm performs best in the classification task?

RandomForestClassifier outperformed the other candidates during validation with AUC of 0.761, even though LinearSVC (Support Vector Machine with linear kernel) and XGBClassifier were performing quite well around 0.75-0.74. However, Logistic Regression and AdaBoost Classifier showed remarkably worse results.

2. As the literature suggests that multimodal machine learning showed good results in the medical domain, it is important to question whether different modalities would contribute to predicting 30-days mortality of elderly hip fracture patients.

(a) To what extent, one can predict 30-days mortality by using chest and hip X-ray images?

On the complete dataset, the model using chest x-ray images achieved 0.70 validation AUC whereas the model with hip x-ray images achieved only 0.614. On the small dataset, the chest model reached 0.642 AUC and hip had 0.632. This means that chest x-ray images contain valuable information and have remarkable prediction power with respect to 30-days mortality.

(b) Different variable groups have difficulties in the collection and extraction phases and might be costly as well. However, if it was possible to extract these variables from x-ray images, it would be less costly and easier. To what extent, extracted features from x-ray images can be used to replace structurally collected variables?

Feature importances of the final model showed that features extracted from the chest and hip x-ray images have much higher contributions to the prediction of 30-days mortality. In fact, four features extracted from chest x-ray images were in the top 5 most important features regarding the prediction of 30-days mortality. On the contrary, comorbidity features extracted from diagnostic treatment codes had the least contribution in the prediction. Therefore, it can be concluded that imaging data, especially chest x-rays contain valuable information with respect to the 30-days mortality of patients, and can be used to replace comorbidity variables extracted from diagnostic treatment codes.

(c) What is the most suitable way to fuse image modality and structured modality when predicting 30-days mortality of the elderly hip fracture patients?

Early fusion (separate training for imaging) appeared to be the best way to fuse image modality and structured modality, as it had 0.801 average validation AUC. Late fusion had 0.77 average validation AUC, the neural network approach achieved 0.756 validation AUC. Early fusion, first takes the inner layer output from convolutional neural networks which are used to extract features from images separately for each image series. Next, it uses these neuron outputs as structured variables in the Random Forest model alongside other structured variables.

(d) To what extent, multimodal fusion improves the prediction on 30-days mortality when compared to prediction with only structured modality?

Without fusing imaging features, the Random Forest model had 0.751 average validation AUC. This score improved to 0.801 with the early fusion of image modality into the Random Forest model.

REFERENCES

[1] B. Gullberg, O. Johnell, and J. A. Kanis. World-wide projections for hip fracture. *Osteoporosis International*, 7(5):407–413, 1997.

[2] Fangke Hu, Chengying Jiang, Jing Shen, Peifu Tang, and Yan Wang. Preoperative predictors for mortality following hip fracture surgery: A systematic review and meta-analysis. *Injury*, 43(6):676–685, 6 2012.

[3] G. Holt, R. Smith, K. Duncan, D. F. Finlayson, and A. Gregori. Early mortality after surgical fixation of hip fractures in the elderly: An analysis of data from the Scottish hip Fracture Audit. *Journal of Bone and Joint Surgery - Series B*, 90(10):1357–1363, 10 2008.

[4] Lisa L Kirkland, Deanne T Kashiwagi, M Caroline Burton, Stephen Cha, and Prathibha Varkey. The Charlson Comorbidity Index Score as a predictor of 30-day mortality after hip fracture surgery. *American journal of medical quality : the official journal of the American College of Medical Quality*, 26(6):461–7, 11 2011.

[5] M J Maxwell, C G Moran, and I K Moppett. Development and validation of a preoperative scoring system to predict 30 day mortality in patients undergoing hip fracture surgery. *BJA: British Journal of Anaesthesia*, 101(4):511–517, 8 2008.

[6] W. S. Nijmeijer, E. C. Folbert, M. Vermeer, J. P. Slaets, and J. H. Hegeman. Prediction of early mortality following hip fracture surgery in frail elderly: The Almelo Hip Fracture Score (AHFS). *Injury*, 47(10):2138–2143, 10 2016.

[7] Julian Karres, Noera Kieviet, Jan-Peter Eerenberg, and Bart C. Vrouenraets. Predicting Early Mortality After Hip Fracture Surgery. *Journal of Orthopaedic Trauma*, 32(1):27–33, 1 2018.

[8] Cornelis Lp van de Ree, Taco Gosens, Alexander H van der Veen, Cees Jm Oosterbos, Martijn W Heymans, and Mariska Ac de Jongh. Development and validation of the Brabant Hip Fracture Score for 30-day and 1-year mortality. *Hip international : the journal of clinical and experimental research on hip pathology and therapy*, page 1120700019836962, 3 2019.

[9] Martyn Parker and Antony Johansen. Hip fracture: Clinical Review. *British Medical Journal (BMJ)*, 333(7557):27–30, 6 2006.

[10] Katie Jane Sheehan, Boris Sobolev, and Pierre Guy. Mortality by Timing of Hip Fracture Surgery. *The Journal of Bone and Joint Surgery*, 99(20):e106, 10 2017.

[11] E. C. Folbert, J. H. Hegeman, M. Vermeer, E. M. Regtuijt, D. van der Velde, H. J. ten Duis, and J. P. Slaets. Improved 1-year mortality in elderly patients with a hip fracture following integrated orthogeriatric treatment. *Osteoporosis International*, 28(1):269–277, 1 2017.

[12] J R Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

[13] Steven L Salzberg. C4.5: Programs for Machine Learning by J. Ross Quinlan. Morgan Kaufmann Publishers, Inc., 1993. *Machine Learning*, 16(3):235–240, 1994.

[14] Leo Breiman, Jerome H Friedman, and Charles J Olshen Richard A }and Stone. *Classification and regression trees*. The Wadsworth statistics/probability series. Wadsworth & Brooks/Cole Advanced Books & Software, Monterey, CA, 1984.

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

[16] Yoav Freund and Robert E Schapire. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 8 1997.

[17] Jerome H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29(5):1189–1232, 2001.

[18] Christopher M Bishop. *Pattern recognition and machine learning*. Information science and statistics. Springer, New York, NY, 2006.

[19] Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 162(1):83–112, 2017.

[20] Aaron Defazio, Francis R Bach, and Simon Lacoste-Julien. SAGA: A Fast Incremental Gradient Method With Support for Non-Strongly }Convex Composite Objectives. *CoRR*, abs/1407.0202, 2014.

[21] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A library for large linear classification. *Journal of machine learning research*, 9(Aug):1871–1874, 2008.

[22] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[23] Rikiya Yamashita, Mizuho Nishio, Richard Kinh Gian Do, and Kaori Togashi. Convolutional neural networks: an overview and application in radiology. *Insights into Imaging*, 9(4):611–629, 2018.

[24] S J Pan and Q Yang. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[25] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In Z Ghahramani, M Welling, C Cortes, N D Lawrence, and K Q Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3320–3328. Curran Associates, Inc., 2014.

[26] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely Connected Convolutional Networks. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-January:2261–2269, 8 2016.

[27] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. International Conference on Learning Representations, ICLR, 9 2015.

[28] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the Inception Architecture for Computer Vision. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2016-December, pages 2818–2826. IEEE Computer Society, 12 2016.

[29] Ueli Masci Jonathan
}and Meier, Cireşan Dan, and Schmidhuber Jürgen. Stacked Convolutional Auto-Encoders for Hierarchical Feature Extraction. In Włodzisław Honkela Timo
}and Duch, Girolami Mark, and Kaski Samuel, editors, *Artificial Neural Networks and Machine Learning – ICANN 2011*, pages 52–59, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[30] AlindGupta. ML | Auto-Encoders.

[31] T Baltrušaitis, C Ahuja, and L Morency. Multimodal Machine Learning: A Survey and Taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2):423–443, 2019.

[32] Y Bengio, A Courville, and P Vincent. Representation Learning: A Review and New Perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

[33] Pradeep K Atrey, M Anwar Hossain, Abdulmotaleb El Saddik, and Mohan S Kankanhalli. Multimodal fusion for multimedia analysis: a survey. *Multimedia Systems*, 16(6):345–379, 2010.

[34] Haibo He and Edwardo A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 9 2009.

[35] Inderjeet Mani and I Zhang. kNN approach to unbalanced data distributions: a case study involving information extraction. In *Proceedings of workshop on learning from imbalanced datasets*, volume 126, 2003.

[36] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[37] Hui Han, Wen Yuan Wang, and Bing Huan Mao. Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. In *Lecture Notes in Computer Science*, volume 3644, pages 878–887. Springer, Berlin, Heidelberg, 2005.

[38] Haibo He, Yang Bai, Edwardo A. Garcia, and Shutao Li. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In *Proceedings of the International Joint Conference on Neural Networks*, pages 1322–1328, 2008.

[39] David W Hosmer Jr, Stanley Lemeshow, and Rodney X Sturdivant. *Applied logistic regression*, volume 398. John Wiley & Sons, 2013.

[40] STANLEY LEMESHOW and DAVID W HOSMER JR. A REVIEW OF GOODNESS OF FIT STATISTICS FOR USE IN THE DEVELOPMENT OF LOGISTIC REGRESSION MODELS1. *American Journal of Epidemiology*, 115(1):92–106, 1 1982.

[41] Hosmer–Lemeshow test - Wikipedia.

[42] Paul Lodder. To impute or not impute: That's the question. *Advising on research methods: Selected topics. Huizen: Johannes van Kessel Publishing*, 2013.

[43] Stef van Buuren and Karin Groothuis-Oudshoorn. mice: Multivariate imputation by chained equations in R. *Journal of Statistical Software*, 45(3):1–67, 12 2011.

[44] Henk Jan Schuijt, Jelle Bos, Diederik Pieter Johan Smeeing, Olivia Geraghty, and Detlef van der Velde. Predictors of 30-day mortality in orthogeriatric fracture patients aged 85 years or above admitted from the emergency department. *European Journal of Trauma and Emergency Surgery*, pages 1–7, 12 2019.

[45] Julian Karres, Nicole A. Heesakkers, Jan M. Ultee, and Bart C. Vrouenraets. Predicting 30-day mortality following hip fracture surgery: Evaluation of six risk prediction models. *Injury*, 46(2):371–377, 2 2015.

[46] T. C. Marufu, S. M. White, R. Griffiths, S. R. Moonesinghe, and I. K. Moppett. Prediction of 30-day mortality after hip fracture surgery by the Nottingham Hip Fracture Score and the Surgical Outcome Risk Tool. *Anaesthesia*, 71(5):515–521, 5 2016.

[47] Hong X Jiang, Sumit R Majumdar, Donald A Dick, Marc Moreau, James Raso, David D Otto, and D William C Johnston. Development and Initial Validation of a Risk Score for Predicting In-Hospital and 1-Year Mortality in Patients With Hip Fractures. *Journal of Bone and Mineral Research*, 20(3):494–500, 2005.

[48] Mary E. Charlson, Peter Pompei, Kathy L. Ales, and C. Ronald MacKenzie. A new method of classifying prognostic comorbidity in longitudinal studies: Development and validation. *Journal of Chronic Diseases*, 40(5):373–383, 1 1987.

[49] K Mohamed, G P Copeland, D A Boot, H C Casserley, I M Shackleford, P G Sherry, and G J Stewart. An assessment of the POSSUM system in orthopaedic surgery. *The Journal of bone and joint surgery. British volume*, 84(5):735–739, 2002.

[50] Yoshio Haga, Satoshi Ikei, and Michio Ogawa. Estimation of physiologic ability and surgical stress (E-PASS) as a new prediction scoring system for postoperative morbidity and mortality following elective gastrointestinal surgery. *Surgery Today*, 29(3):219–225, 1999.

[51] K. L. Protopapa, J. C. Simpson, N. C. E. Smith, and S. R. Moonesinghe. Development and validation of the Surgical Outcome Risk Tool (SORT). *British Journal of Surgery*, 101(13):1774–1783, 12 2014.

[52] Heung Il Suk, Seong Whan Lee, and Dinggang Shen. Hierarchical feature representation and multimodal fusion with deep learning for AD/MCI diagnosis. *NeuroImage*, 101:569–582, 11 2014.

[53] Kenneth Er, Acharya U. Rajendra, N. Kannathal, and Lim Choo Min. Data fusion of multimodal cardiovascular signals. In *Annual International Conference of the IEEE Engineering in Medicine and Biology - Proceedings*, volume 7 VOLS, pages 4689–4692. Conf Proc IEEE Eng Med Biol Soc, 2005.

[54] Roger Bramon, Imma Boada, Anton Bardera, Joaquim Rodríguez, Miquel Feixas, Josep Puig, and Mateu Sbert. Multimodal data fusion based on mutual information. *IEEE Transactions on Visualization and Computer Graphics*, 18(9):1574–1587, 2012.

[55] Nelson Nunes, Bruno Martins, Nuno André da Silva, Francisca Leite, and Mário J. Silva. A multi-modal deep learning method for classifying chest radiology exams. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 11804 LNAI, pages 323–335. Springer Verlag, 9 2019.

[56] W S Nijmeijer, E C Folbert, M Vermeer, M M R Vollenbroek-Hutten, and J H Hegeman. The consistency of care for older patients with a hip fracture: are the results of the integrated orthogeriatric treatment model of the Centre of Geriatric Traumatology consistent 10 years after implementation? *Archives of Osteoporosis*, 13(1):131, 2018.

[57] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2016-December, pages 770–778. IEEE Computer Society, 12 2016.

[58] François Chollet. Xception: Deep Learning with Depthwise Separable Convolutions. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-January:1800–1807, 10 2016.

A IMAGE SELECTION

Imaging data in medical domain usually is in DICOM format. This format lets user to add meta-data and labels to the image so that it could be stored and processed in an efficient manner. Some of the important features for the identification of image are Study Description, Series Description, Body Part Examined and Accession Number. Study Description describes the kind of study such as Bekken/Heup(Hip) or Thorax(Chest), similar to that Body Part Examined specifies obviously which body part was examined during that study. Series description is used to describe the view of the study such as AP (Anterior-Posterior) or lateral. However, a study could have more than one series. Imaging data of a particular patient is retrieved by the ”accession number” which is referring to the study. Therefore, when an accession number is used to extract a study of a patient, usually one can end up with multiple series, i.e. multiple images of the study with different views. The problem at this point is that, not all the meta-data are correct. Sometimes, data registry goes wrong in the radiology and one can end up with a chest x-ray from a study of hip or vice versa. Furthermore, it could also be that an image with a series description ”AP” might be in fact lateral. On the other hand, these labels are not perfectly standardized, meaning that, one can register a thorax AP image with series description ”ap” or ”thorax ap” or ”pa” and so on. All of these mentioned issues makes the image data extraction by query extremely difficult and impossible to end up with desired result. For this reason, an Artificial Intelligence had to be developed to select the desired images for each patient. Desired images in this project are one ”AP” view of Thorax and one ”AP” view of pelvis or hip. The other views are discarded due to lower quality.

First step was to develop two datasets, one for thorax studies and one for hip studies.

A.1 Hip Dataset Creation

Filters used for ”StudyDescription” to include in hip dataset: ’Bekken+heup links’, ’Bekken+heup rechts’, ’Bekken’, ’Heup rechts’, ’Heup links’, ’Bekken+heup beiderzijds’, ’Bekken,femur,knie rechts’. After applying this filter to the ”StudyDescription”, ’PELVIS’ filter was applied to ”BodyPartExamined”.

From the remaining images, desired images had to be selected and labeled as correct so that they can be used in supervised learning on the next stage. This is done by first selecting images which contain ’AP’ (none case-sensitive) in ”SeriesDescription”. Finally, images were grouped by the accession number and the groups which have more than 1 member were discarded. By this way, studies with only single ’AP’ series were obtained. After adding few images by manual observation, positive samples of this dataset was completed, namely correct images.

Afterwards, negative samples had to be collected, namely incorrect images. For this class, any x-ray image which is not a hip study with ap view can be eligible. Therefore, images having

'StudyDescription' one of 'Thorax', 'Thorax liggend op bed', 'Thorax op bed' were put in this class, except the ones having "BodyPartExamined" = 'PELVIS'. This filter contained mainly chest x-rays, and thus helped the model to distinguish between hip images and chest images. Moreover, images with "SeriesDescription" containing one of 'lau', 'ax', 'lat' were also marked as incorrect and put in this class. This filter helped the model to distinguish between 'AP' and other views of hip x-ray images.

In the end, the dataset consisted 2680 correct and 7006 incorrect images. Train, validation and test split took place with ratios of 50%, 40% and 10% respectively.

A.2 Chest Dataset Creation

During chest dataset creation, a similar procedure was followed as in hip dataset creation. Filters used for "StudyDescription" to include in chest dataset: 'Thorax', 'Thorax liggend op bed', 'Thorax op bed'. After applying this filter to the "StudyDescription", 'CHEST' filter was applied to "BodyPartExamined".

From the remaining images, desired images had to be selected and labeled as correct so that they can be used in supervised learning on the next stage. This is done by first selecting images which contain 'AP' or 'PA' (none case-sensitive) in "SeriesDescription". Finally, images were grouped by the accession number and the groups which have more than 1 member were discarded. By this way, studies with only single 'AP' series were obtained. After adding few images by manual observation, positive samples of this dataset was completed, namely correct images.

Afterwards, negative samples had to be collected, namely incorrect images, this part was slightly different than hip dataset creation. First, correct images of the hip dataset was used as incorrect images of the chest dataset. Afterwards, if a image has "BodyPartExamined" = 'HIP' and "StudyDescription" one of 'Thorax', 'Thorax liggend op bed', 'Thorax op bed', then they were marked as incorrect. Furthermore, images with "BodyPartExamined" = 'HIP' or 'CHEST', and contain 'ax' or 'la' in their series description were also marked as incorrect.

In the end, the dataset consisted 2932 correct and 6861 incorrect images. Train, validation and test split took place with ratios of 50%, 40% and 10% respectively.

A.3 Modelling

Convolutional Neural Networks were used during the modelling phase of image selection. With respect to the implementation, Keras library was used with TensorFlow backend. The first attempt was using DenseNet169 [26] by means of transfer learning for the convolutional part of the network. However, it achieved extremely good results and therefore no further experiment had to be executed. DenseNet169 model was fine tuned seperately for each of the datasets. Fine tuning was done by adding fully connected and dropout layers at the end of CNN and training with correct/incorrect labels. Information on added layers can be found in table A.3. Data augmentation and hyperparameters used were completely same in modelling of both datasets and can be found in table A.2 and A.1. Validation and test results for chest and hip models can be found in table A.4. After testing the models, all of the images were fed into the models and got predicted, i.e. received a score between 1 and 0 regarding their correctness. Finally,

for each patient, image with highest chest score was selected as the correct chest image and image with highest hip score was selected as the correct hip image for that patient. By this way, it was achieved to associate almost all patients with one chest 'AP' and one pelvis/hip 'AP' scan.

Setting/Hyperparameter	Value
Number of Epochs	100
Loss Function	Binary Cross Entropy
Optimizer	Adam
Learning Rate	0.001
Monitored Value for Early Stopping	Validation Accuracy
Patience for Early Stopping	10
Minimum delta required for improvement	0.0001
Monitored Value for Reducing Learning Rate	Validation Accuracy
Patience for Reducing Learning Rate	6
Factor by which the learning rate will be reduced	0.1

Table A.1: Settings/Hyperparameters used in Experiments

Data Augmentation Setting	Value
Rotation Range	20
Width Shift Range	0.1
Height Shift Range	0.1
Shear Range	0.1
Zoom Range	0.1
Channel Shift Range	10
Horizontal Flip	True
Fill mode	Nearest
Interpolation(also applies to validation & test)	Bicubic

Table A.2: Data augmentation settings used in experiments for image data generation during training phase

Layer Name	Number of Neurons	Activation Function	Bias Initializer	Dropout Rate
Dense_0	1024	Relu	zeros	-
Dropout_0	-	-	-	0.5
Dense_1	512	Relu	zeros	-
Dropout_1	-	-	-	0.5
Dense_2	256	Relu	zeros	-
Dropout_2	-	-	-	0.5
Dense_3	128	Relu	zeros	-
Dropout_3	-	-	-	0.5
Output	1	Sigmoid	zeros	-

Table A.3: Fully Connected Layers which are added after the convolutional part of pre-trained models

Dataset	Validation Accuracy	Test Accuracy
Chest	100%	99.89%
Hip	99.26%	99.39%

Table A.4: Results of image selection models

B CONVOLUTIONAL AUTO-ENCODERS

Auto-encoders are a technique of unsupervised learning. As this study is concerned with multimodality, learning a representation of image modality in an unsupervised way was also one of the approaches. However, the end goal was to use this feature set which represents image modality, in traditional machine learning algorithms such as Random Forest, Logistic regression. This requires such an extraction that with only a few variables, signs of early mortality should be summarized. To this end, convolutional auto-encoders were employed to extract features from image modality. Experimental settings used on auto-encoders can be seen in table B.1

The performance of auto-encoders in this context is evaluated by their ability to reconstruct images from the encoded features. However, when the target encoding dimensions decrease, more and more information loss occurred as the quality of the reconstructed image dropped significantly. In table B.2, layer architecture can be seen sequentially. In the encoding part, each block consists of one convolutional layer and one max-pooling layer, whereas in the decoding part, the upsampling layer is used as the opposite operation of max-pooling. This example shows an architecture of an auto-encoder with 3 blocks on each encoding and decoding parts. After the encoding, the feature set has a dimension shape of $32 \times 32 \times 64 = 65536$ which is extremely big for use in traditional machine learning. Other architectures were also tried out with more blocks or with different numbers of kernels. Original images of 3 patients can be seen in figure B.1. Reconstructed images from different experiments are in shown figures B.2 - B.7. In figures B.4 and B.7, the reconstruction of three different images ended up in the same black image which does not carry any information. The conclusion was that this approach will not contribute to 30-days mortality prediction in a practical way due to the high amount of information loss and necessity of a high number of features to represent image modality.

Setting/Hyperparameter	**Value**
Number of Epochs	200
Loss Function	Mean Squared Error
Optimizer	Adam
Learning Rate	0.01
Monitored Value for Early Stopping	Validation Loss
Patience for Early Stopping	20
Batch Size	20
Initial Image Dimension	256x256
Color mode	Gray Scale

Table B.1: Settings/Hyperparameters used in auto-encoder experiments

Layer	Kernel size	Nr. of Kernels	Activation	Padding	Interpolation
Convolutional2D	3x3	64	Relu	zero padding	-
MaxPooling2D	2x2	-	-	zero padding	-
Convolutional2D	3x3	64	Relu	zero padding	-
MaxPooling2D	2x2	-	-	zero padding	-
Convolutional2D	3x3	64	Relu	zero padding	-
MaxPooling2D	2x2	-	-	zero padding	-
Convolutional2D	3x3	64	Relu	zero padding	-
UpSampling2D	2x2	-	-	-	nearest
Convolutional2D	3x3	64	Relu	zero padding	-
UpSampling2D	2x2	-	-	-	nearest
Convolutional2D	3x3	64	Relu	zero padding	-
UpSampling2D	2x2	-	-	-	nearest

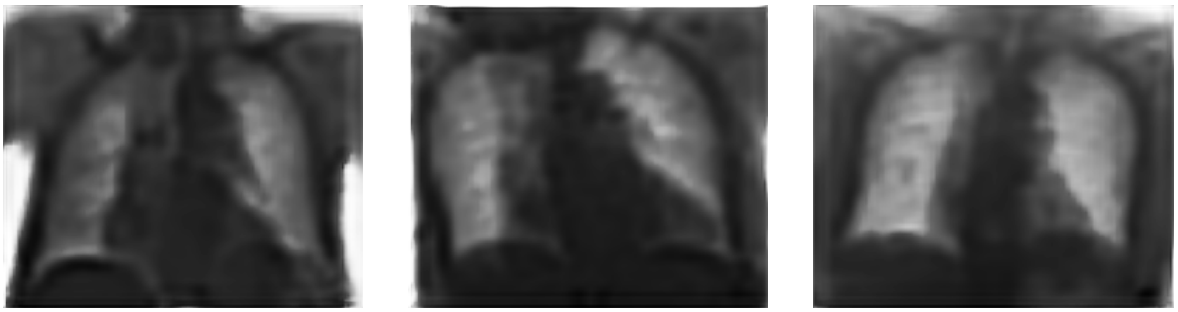
Table B.2: Example layer architecture of an auto-encoder with 3 blocks and 64 kernels on each convolutional layer

(a) Patient 1 (b) Patient 2 (c) Patient 3

Figure B.1: Example chest x-ray scans of patients, original version

(a) Patient 1 (b) Patient 2 (c) Patient 3

Figure B.2: Reconstructed images on auto-encoder with 3 blocks and 64 kernels on each convolutional layer, number of features after encoding is 65536



(a) Patient 1 (b) Patient 2 (c) Patient 3

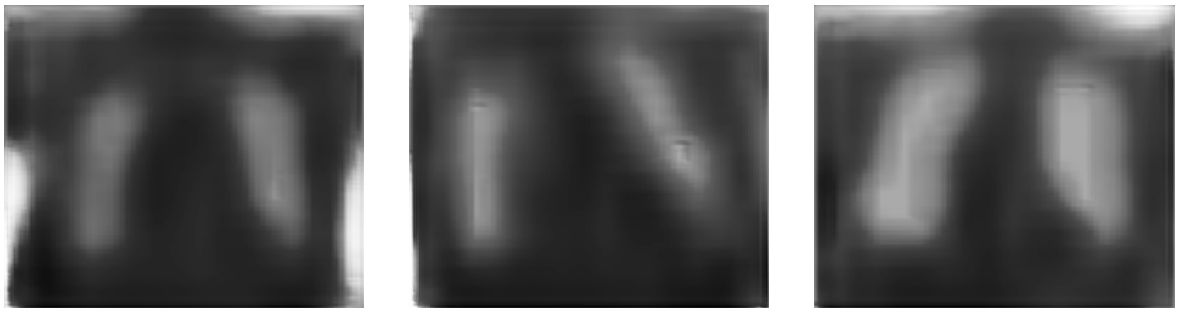
Figure B.3: Reconstructed images on auto-encoder with 5 blocks and 64 kernels on each convolutional layer, number of features after encoding is 4096

(a) Patient 1 (b) Patient 2 (c) Patient 3

Figure B.4: Reconstructed images on auto-encoder with 7 blocks and 64 kernels on each convolutional layer, number of features after encoding is 256

(a) Patient 1 (b) Patient 2 (c) Patient 3

Figure B.5: Reconstructed images on auto-encoder with 3 blocks and 16 kernels on each convolutional layer, number of features after encoding is 16384



(a) Patient 1 (b) Patient 2 (c) Patient 3

Figure B.6: Reconstructed images on auto-encoder with 5 blocks and 16 kernels on each convolutional layer, number of features after encoding is 1024

(a) Patient 1 (b) Patient 2 (c) Patient 3

Figure B.7: Reconstructed images on auto-encoder with 7 blocks and 16 kernels on each convolutional layer, number of features after encoding is 64