Prediction of Functional Outcome After Ischemic Stroke Using Deep and Interpretable Regression Models.

Master Thesis in Biostatistics (STA495)

by

Katrin Petermann, MSc
04-800-678

supervised by

Prof. Dr. Beate Sick
Lisa Herzog, MSc

University of Zurich[UZH]

Zurich, August 2021

Acknowledgements

This thesis would not have been possible without the support of many people. First I would like to thank my supervisors, Prof. Beate Sick and Lisa Herzog for their continuous support throughout the last year. Many thanks also go to the whole staff from the biostatistics department for the inspiring curriculum and for the possibility to complete the Masters program working part-time. I was lucky to find myself among many motivated students who helped and encouraged each other during the whole program. Special thanks go to Lucas Kook not only for his feedback during my thesis, but also for his critical mind and our friendship.
My parents and my brothers deserve endless gratitude for planting the seed of curiosity and intrinsic motivation.
Last but not least, I would like to thank my team at the movement disorders clinics at the Inselspital in Bern. Only their support made it possible for me to complete this Masters program in these difficult times during the COVID-19 pandemic.

Katrin Petermann
August 2021

Contents

Chapter 1

Introduction

1.1 Unstructured Data in Medicine

Within the last three decades, the field of statistics has undergone fundamental changes due to the increased availability of computational power. Models have become more complex, large datasets are widespread and statisticians spend most of their time programming. At the same time, the type of data we use for inference about the relationship between variables has stayed the same. When we think about a dataset, we think about tabular data, observations or measurements that can be described by numbers or simple characteristics. In medicine, popular measurements are blood pressure or weight, while a characteristic might be the gender of a patient. But what about other types of data? For example an EEG to measure brain activity, respiratory sounds detected by a stethoscope or radiological images?
For statistical analysis, these types of data are not usually used in the raw format as sound or image, but transformed to a number that then can be entered into a table. This feature engineering can be very time consuming and demands a lot of expert knowledge. However with the emergence of Artificial Neural Networks (ANNs) and Deep Learning (DL), methods for the analysis of complex or unstructured data have been dominating the field for the last few years (Goodfellow et al., 2016). Features of the data don't have to be predefined and extracted from raw data manually, but ANNs are able to automatically learn the relevant features that are hidden within the complex dataset. First commercial applications of ANNs in medicine were introduced recently, with a focus on disease detection on medical images (qure.ai[1], RetinAI[2], Zebra Medical Vision[3]). Today, the best performing algorithms for image classification (diseased vs. healthy) are Deep Convolutional Neural Networks (CNNs).
CNNs are a type of ANN consisting of convolutional layers, used for automatic feature extraction, followed by fully connected layers used for classification. Convolution is a process, where only a few neighboring cells are connected to the neuron in the next layer, which makes it possible to extract features containing information about groups of neighboring pixels instead of the image as a whole[4]. While CNNs show impressive performances in image classification, the interpretation of the automatically extracted features in not straight forward. And while CNNs are a powerful tool for 2D image classification, in medicine the acquired images are frequently of three-dimensional structure. While MRIs and CTs are standard of care in medicine, there are only few methods that use the three dimensional information for analysis. Compared to statistical regression models, where interpretability is key, NN are often called a *black box*. How much are we willing to trust a *black box* algorithm in medical decision making? Ideally, for medical applications, the combination of complex and tabular data into a single interpretable

[1] https://qure.ai/, Online; accessed 12-July-2021
[2] https://www.retinai.com/, Online; accessed 12-July-2021
[3] https://www.zebra-med.com/, Online; accessed 12-July-2021
[4] For a more detailed introduction to NN refer to Nielsen (2015) and Goodfellow et al. (2016)

model is desirable. In the case of ordinal outcomes this can be achieved through Ordinal Neural Network Transformation Models (ONTRAMs) (Kook & Herzog et al. (2020)).

1.2 Stroke

A stroke is a medical condition where the blood supply to the brain is interrupted, leading to neurological deficits. Most common symptoms include inability to move or feel parts of the body, slurred speech and loss of vision. A stroke is either of hemorrhagic or ischemic origin. While hemorrhagic strokes are caused by a brain bleed, ischemia is in most cases caused by a clogged blood vessel, leading to a reduction in blood flow, resulting in hypoxia and tissue damage. A Transient Ischemic Attack (TIA) has the same underlying mechanism as an ischemic stroke, but the symptoms typically disappear within one or two hours. TIA patients experience transient neurological deficits, but contrary to stroke patients, no persistent tissue damage can be detected on an Magnetic Resonance Image (MRI) (Stroke – Wikipedia).
In Switzerland about 16'000 people suffer a stroke each year, 85% of which are caused by ischemia (Swiss Neurological Society (2019)). Ischemic strokes are treated by unclogging the vessel, either through drugs or mechanically. Speed of diagnosis and correct treatment is crucial for survival and favorable outcome. But also factors like age and the size and localization of the stroke play a major role in recovery. In order to predict the functional outcome after stroke, one has to consider patient characteristics, risk factors and imaging data.

1.3 Objectives

For this Master Thesis a dataset consisting of unstructured data (MRIs) as well as tabular data from ischemic stroke and TIA patients was used. The overall goal of the thesis was to adapt ONTRAMs to 3D images, to predict functional outcome three months after ischemic stroke or TIA and to interpret the effect of the different risk factors. In order to do so, some intermediary steps had to be implemented first:

- Build a dataset out of brain MRIs that can be used for training a 3D CNN.
- Develop a 3D CNN capable to extract relevant features and detect ischemic strokes on MRIs.
- Integrate the 3D CNN into an ONTRAM model using unstructured and tabular data.
- Use 3D MRIs and tabular data to predict functional outcome three months after ischemic stroke or TIA.
- Evaluate the predictive performance of the ONTRAM model.
- Interpret the impact of the different model parts.

Chapter 2

Data and Methods

2.1 Data Preparation and Description

The data was collected retrospectively from 2013 to 2018 at the University Hospital in Zürich. The final dataset included a total of 497 patients. All of them arrived at the emergency department of the hospital showing neurological symptoms of a stroke. Only ischemic events, either transient or persistent (TIA or stroke) were included into our database. The Patient selection was not done systematically, but rather such that each patient, who had obtained a DWI sequence (see section 2.1.2) and was well enough to consent to the use of their medical records for research purposes, was included. Accordingly, there is a selection and a survival bias in our data, since patients with severe symptoms or deceased patients were not included.

2.1.1 Tabular Data

Tabular data was extracted from the Swiss Stroke Registry (Bonati (2015)), where baseline variables, tabular risk factors and outcome variables were selected.

Outcome Variables

For the three different experiments (see section 2.3), the following outcome variables were used:

- `Event`: binary variable, indicating if the patient suffered an ischemic *Stroke* or a *TIA*. The distribution of the variable can be seen in Figure 2.1 A.
- `mRS ordinal`: ordinal variable with 7 levels. The modified Rankin Scale (mRS) described in Table 2.1 is a measure for the degree of disability, evaluated at three months after the event. Figure 2.1 B shows the distribution of this variable.
- `mRS binary`: binary variable, dichotomized `mRS ordinal` into *good* (`mRS ordinal` ≤ 2) and *bad* (`mRS ordinal` > 2). Figure 2.1 C depicts the distribution of the dichotomized outcome variable.

Table 2.1: The modified Rankin Scale (mRS) is a scale used for measuring the degree of disability in activities of daily living of patients who have suffered a stroke. The scale runs from 0 to 6, running from perfect health without symptoms to death (Wilson et al., 2002).

Modified Rankin Scale (mRS)	
0	No symptoms.
1	No significant disability. Able to carry out all usual activities, despite some symptoms.
2	Slight disability. Able to look after own affairs without assistance, but unable to carry out all previous activities.
3	Moderate disability. Requires some help, but able to walk unassisted.
4	Moderately severe disability. Unable to attend to own bodily needs without assistance, and unable to walk unassisted.
5	Severe disability. Requires constant nursing care and attention, bedridden, incontinent.
6	Dead.

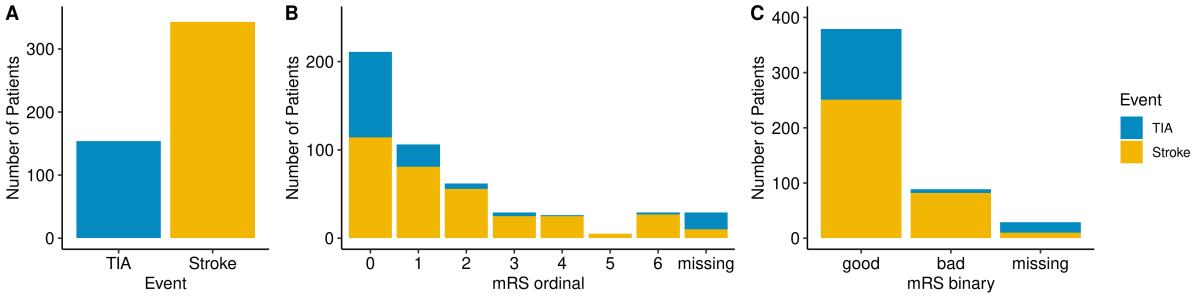


Figure 2.1: Distribution of the outcome variables. (A) Outcome `Event` with levels *TIA* and *Stroke*, (B) mRS at 3 months as ordinal variable called `mRS ordinal`, (C) mRS at 3 months is dichotomized into `mRS binary` with cutoff *good*: `mRS ordinal` ≤ 2. Patients with missing outcomes were removed from the datasets for experiments with outcome `mRS`.

Baseline Variables and Risk Factors

Table 2.2 describes the dataset and lists the explanatory variables extracted from the Swiss Stroke Registry. Baseline variables are *Age*, *Gender*, *mRS before Event*, *Previous TIA*, *Previous Ischemic Stroke* and *NIHSS at Baseline*. The National Institutes of Health Stroke Scale (NIHSS) is a scale used for quantification of the severity of a stroke through a neurological examination. The NIHSS ranges from 0 to 42 with high scores indicating high severity. For the purpose of this thesis, *NIHSS* and *mRS before Event* are treated as continuous variables. As risk factors, *High Cholesterol*, *Coronary Heart Disease*, *Atrial Fibrillation*, *Diabetes*, *High Blood Pressure* and *Smoker* were selected. The distributions of categorical baseline variables and risk factors can be seen in Figure 2.2

Missing Data

Out of the 497 included patients, 456 had no missing values. There were 88 missing variables for 41 patients, 29 for `mRS at 3 months` and 59 for various explanatory variables. The later were imputed using random forest imputation with the R package `missForest` (Stekhoven and Bühlmann, 2012) with default settings. 29 patients with missing outcome variables were removed from the experiments with outcome `mRS` resulting in a sample size of 468 patients. There were no missing `Event` labels.

2.1.2 Imaging Data

Patients were selected for the study if they received a Diffusion Weighted magnetic resonance Image (DWI) when they first arrived at the emergency room. Each examination consisted of

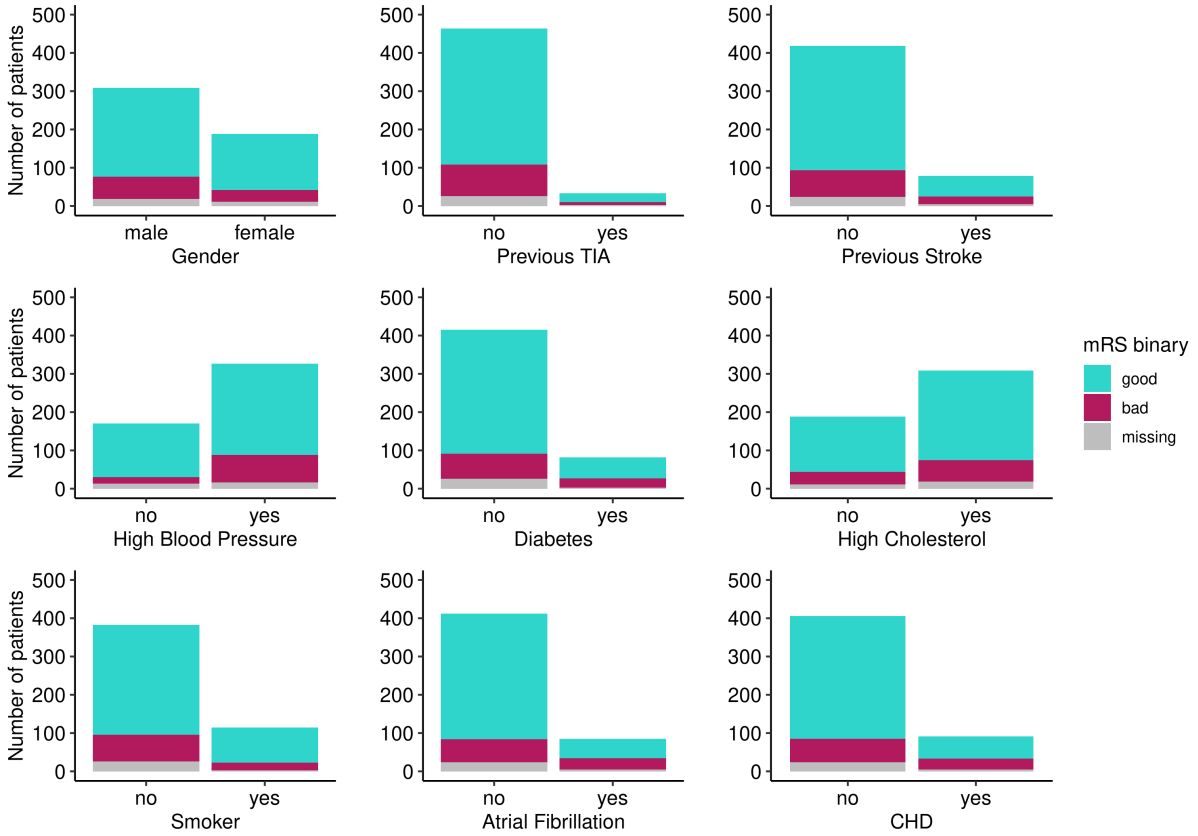


Figure 2.2: Categorical baseline variables and risk factors colored by `mRS binary`. There are *good* and *bad* outcomes for each level of each variable. CHD: Coronary Heart Disease

axial cross sections through the brain, with an average of 30 slices per patient. The slices were exported in JPG format and used as individual images in other studies (Herzog et al., 2020). However, when combining the individual images into a 3D stack, it became apparent that there were differences in brightness levels between adjacent slices, which is a problem when moving to the analysis of 3D images. Therefore, the DWIs were exported in DICOM[1] format to achieve a consistent brightness level over all slices. Figure 3.1 shows the difference between JPG and DICOM format.

Validation of DICOM Dataset

Image labels (*Stroke/TIA*) were assigned to each JPG image and had to be copied to the DICOM dataset. To make sure that the labels were correctly assigned, every image was visually checked and removed if JPG and DICOM data didn't match. To check if the new DICOM images are comparable to the original JPG dataset, both datasets were used to train a neural network previously described by Herzog et al. (2020). As evaluation metrics for the validation, the Negative Log Likelihood (NLL), Accuracy, Sensitivity, Specificity and Area Under the ROC Curve (AUC) were used. The different evaluation metrics are described in Appendix A.1.

[1] DICOM: Digital Imaging and Communications in Medicine. Worldwide standard for storing and transmitting medical images.

Table 2.2: Explanatory variables used in the experiments, stratified by `Event` (*TIA* vs. *Stroke*). For continuous variables (*Age* and *NIHSS*) mean and standard deviation (in brackets) are given, while categorical variables are listed in counts and percent (in brackets).

Variables	Levels	TIA	Stroke	% Missing
n		154	343	
Age		68.56 (14.78)	66.94 (15.35)	0.0
Gender	male	95 (61.7)	214 (62.4)	0.0
	female	59 (38.3)	129 (37.6)	
mRS before Event	0	116 (76.8)	269 (79.1)	1.2
	1	17 (11.3)	29 (8.5)	
	2	10 (6.6)	25 (7.4)	
	3	7 (4.6)	14 (4.1)	
	4	1 (0.7)	3 (0.9)	
NIHSS at Baseline		1.00 (1.87)	6.46 (6.33)	1.0
Previous TIA	no	136 (89.5)	322 (95.0)	1.2
	yes	16 (10.5)	17 (5.0)	
Previous Ischemic Stroke	no	122 (80.3)	290 (85.5)	1.2
	yes	30 (19.7)	49 (14.5)	
High Cholesterol	no	50 (32.9)	137 (40.4)	1.2
	yes	102 (67.1)	202 (59.6)	
Coronary Heart Disease	no	128 (84.2)	274 (80.8)	1.2
	yes	24 (15.8)	65 (19.2)	
Atrial Fibrillation	no	137 (90.1)	270 (79.6)	1.2
	yes	15 (9.9)	69 (20.4)	
Diabetes	no	130 (85.5)	280 (82.6)	1.2
	yes	22 (14.5)	59 (17.4)	
Smoker	no	126 (82.9)	252 (74.3)	1.2
	yes	26 (17.1)	87 (25.7)	
High Bloodpressure	no	54 (35.5)	115 (33.9)	1.2
	yes	98 (64.5)	224 (66.1)	

Praparing 3D Image Stacks

In order to feed a 3D CNN, the dimensions of the 3D images need to be constant over all patients. Linear interpolation was used to standardize the original volume (192×192 pixels, $24 - 46$ slices) to $128 \times 128 \times 30$ voxels. For linear 3D interpolation, the Python function `zoom` of the `ndimage` package in `SciPy` was used (Virtanen et al., 2020).

2.2 Methods

2.2.1 3D CNN

Architecture and Training

The development of the three dimensional CNN architecture was inspired by the work of other groups, who used 3D CNNs for classification of medical images (Kan et al., 2021; Zunair et al., 2020). The 3D CNN takes as input 3D images with dimensions width $\times$ height $\times$ depth $= x \times y \times z = 128 \times 128 \times 30$ and consists of 5 convolutional layers with max pooling, followed by 2 dense layers with dropout. Batch normalization was used in every layer. For the 3D convolution, kernels with dimensions $3 \times 3 \times 3$ were used with $32 - 32 - 64 - 64 - 128$ filters per layer. The dense layers consisted of 128 neurons each, and the number of outputs was $K - 1$ with K being the number of classes in the ordinal outcome. This architecture results in a total of almost 696'000 trainable parameters (also called weights), depending on the number of classes in the output. These weights are not interpretable but used to calculate the predicted probability of each class, depending on the input image. ReLU (Rectified Linear Unit) was used as activation function, the NLL (Negative Log Likelihood, see Equation 2.3) as loss function and the Adam optimizer for stochastic gradient descent (Kingma and Ba, 2017). Learning rate, number of epochs and

batch size were adjusted for each model and outcome, but not tested systematically. Figure 2.3 shows a schematic representation of the 3D CNN architecture. The Python implementation of the 3D CNN can be found on github: https://github.com/kilyth/MasterThesis.

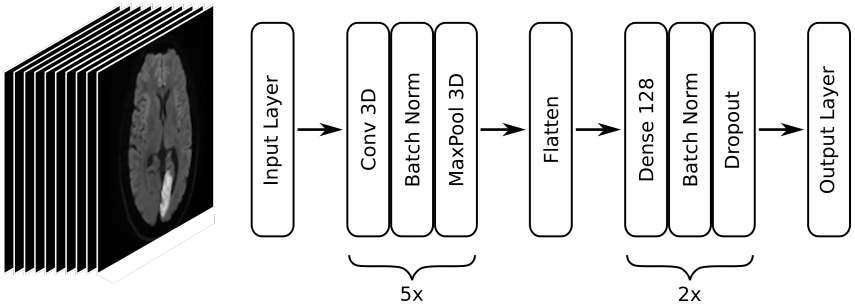


Figure 2.3: Schematic depiction of the 3D CNN architecture. For details refer to the Python implementation on github.

Image Preprocessing and Augmentation

After linear interpolation to a common image size of $128 \times 128 \times 30$ voxels (Section 2.1.2), the only preprocessing done before training was a standardization of voxel values per image to zero mean and unit variance.
The following augmentation strategies were implemented and combined to ensure that in each epoch unseen data was fed to the network:

- **zoom**: the image was zoomed in 3D, using random factors between 0.7 and 1.4, keeping the initial image size of $128 \times 128 \times 30$ voxels. Black voxels were added to the edges when the zoom factor was smaller than 1.
- **rotation**: the image was rotated in 3D, using random angles between -30 and 30 degrees for rotation around the z–axis and random angles between -10 and 10 degrees for rotation around the x– and y–axes. Note that rotation around x and y leads to a shear, since we don't have isotropic voxels for our 3D data.
- **shift**: the image was randomly shifted in 3D, with a shift between -20 and 20 pixels in x and y direction and between -5 and 5 slices in z direction. Where needed, black voxels were added to the image.
- **flip**: the image was randomly flipped, i.e. mirrored at the y–z plane.
- **gaussian filter**: an isotropic 3D Gaussian filter was applied, with random standard deviation between 0 and 0.2, leading to smoothing of the image.

2.2.2 Ordinal Regression Transformation Models

Transformation Models in General

In transformation models, we estimate the conditional outcome distribution $F_Y(y|\boldsymbol{x})$ through a transformation into a continuous conditional distribution function $F_Z(h(y|\boldsymbol{x}))$ (Hothorn et al., 2014).

$$F_Y(y|\boldsymbol{x}) = F_Z(h(y|\boldsymbol{x})). \tag{2.1}$$

Estimating $F_Y(y|\boldsymbol{x})$ is thus translated into a problem of estimating the parameters of a monotonically increasing transformation function $h(y|\boldsymbol{x})$ (Figure 2.4 B and D).

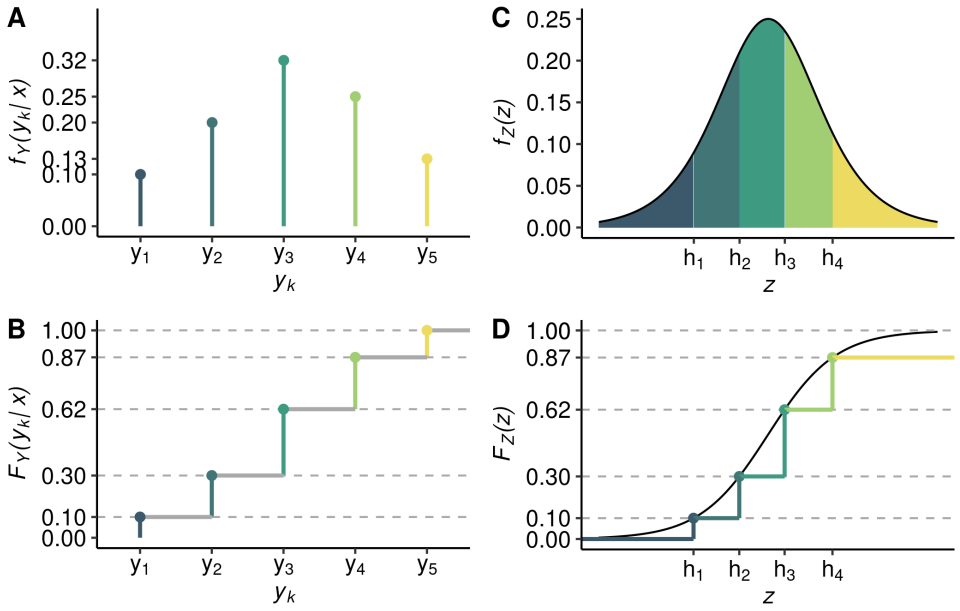


Figure 2.4: Ordinal Regression Transformation Model. Density and distribution functions for an ordinal outcome with $K = 5$ classes. Probability density functions (PDFs, panels A and C) and cumulative distribution functions (CDFs, panels B and D) of the variables Y and Z. The PDF corresponds to the probability to belong to class k, while the CDF describes the probability to belong to any class $\leq k$. The heights of the steps in the CDF correspond to the probability to belong to class k. $F_Y(y|\boldsymbol{x}) = F_Z(h(y|\boldsymbol{x}))$ can be seen by comparing Subfigures B and D. Note that there are only $K - 1 = 4$ cutpoints, since $h_5 = +\infty$. Figure adapted from Kook & Herzog et al. (2020).

In the case of ordinal regression, the transformation function $h(y_k|\boldsymbol{x})$ comprises of $K - 1$ points. It transforms the ordinal outcome y_k into cutpoints of a continuous latent variable F_Z, as can be seen in Figure 2.4.

Estimating the Transformation Function

After choosing F_Z, estimating regression parameters comes down to estimating the transformation function $h(y|\boldsymbol{x})$ via the maximum likelihood method. The likelihood contribution of a given observation $(y_{ki}, \boldsymbol{x}_i)$ is given by

$$
\begin{aligned}
\mathcal{L}_i(h; y_{ki}, \boldsymbol{x}_i) &= \mathbb{P}(Y = y_{ki}|\boldsymbol{x}_i) && \text{height of } f_Y, \text{ Figure 2.4 A} \\
&= F_Y(y_{ki}|\boldsymbol{x}_i) - F_Y(y_{(k-1)i}|\boldsymbol{x}_i) && \text{height of step in } F_Y, \text{ Figure 2.4 B} \\
&= F_Z(h(y_{ki}|\boldsymbol{x}_i)) - F_Z(h(y_{(k-1)i}|\boldsymbol{x}_i)) && \text{height of step in } F_Z, \text{ Figure 2.4 D} \\
&= \int_{h_{k-1}}^{h_k} f_Z(z)dz && \text{shaded area under } f_Z, \text{ Figure 2.4 C}
\end{aligned} \tag{2.2}
$$

In order to calculate the likelihood, we need to consider two consecutive cutpoints h_{k-1} and h_k. Thus the natural ordering of the outcome classes is taken into account.
Instead of maximizing the likelihood, we will be minimizing the Negative Log Likelihood (NLL) over all samples:

$$
-\frac{1}{n}\ell(h; y_{1:n}, \boldsymbol{x}_{1:n}) = -\frac{1}{n}\sum_{i=1}^{n} \log \mathcal{L}_i(h; y_i, \boldsymbol{x}_i) \tag{2.3}
$$

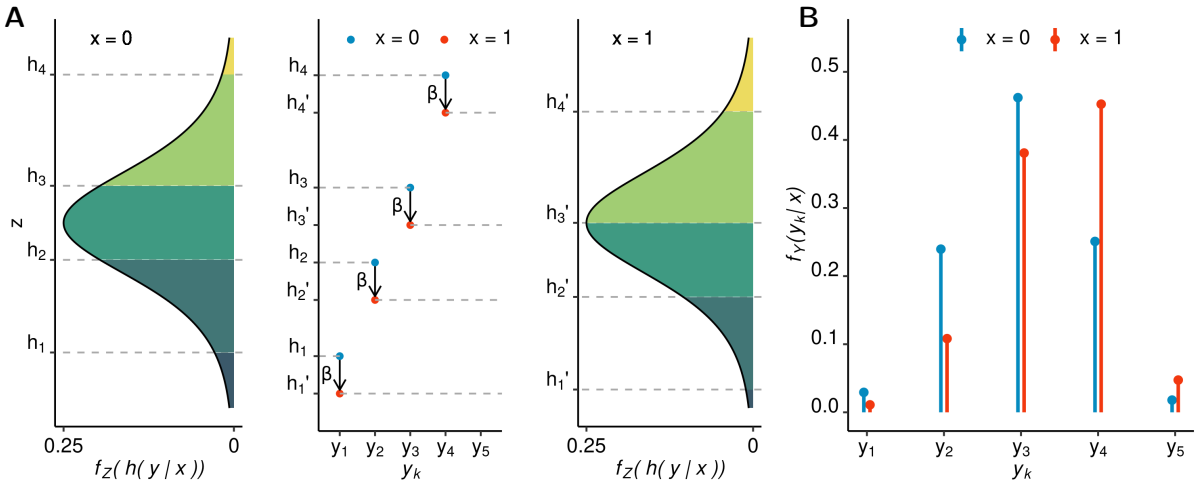


Figure 2.5: Linear shift transformation model for ordinal outcomes. (A) h_1 to h_4 are the $K-1$ cutpoints that define the transformation function in the case of an ordinal outcome with $K = 5$ classes. When the variable x is increased by one unit (from 0 to 1), all cutpoints get shifted by the same amount β. This induces a non linear shift of the probabilities for each class (area under the curve), that are colored with different shades of green. (B) The probabilities for each class shift in a non linear way when the variable x is increased by one unit. The heights of the probabilities f_Y correspond to the area under the curve of f_Z shown in panel (A).

Interpretability of Transformation Models

The interpretability of the parameters in a transformation model depends on the choice of F_Z and the structure of the transformation function $h(y|\boldsymbol{x})$. Since our primary outcome is on an ordinal scale, we will focus on ordinal regression transformation models with K classes. We choose for F_Z the standard logistic distribution F_L[2], while the transformation function is parametrized as a linear shift model:

$$h(y_k|\boldsymbol{x}) = \vartheta_k - \sum_{j=1}^{J} \beta_j x_j = \vartheta_k - \boldsymbol{x}^T\boldsymbol{\beta}, \quad j = 1, \ldots, J. \tag{2.4}$$

Then, the odds for the outcome to belong to a higher class than y_k can be written as:

$$\begin{aligned} \text{odds}(Y > y_k|\boldsymbol{x}) &= \frac{\mathbb{P}(Y > y_k|\boldsymbol{x})}{\mathbb{P}(Y \leq y_k|\boldsymbol{x})} = \frac{1 - F_Y(y_k|\boldsymbol{x})}{F_Y(y_k|\boldsymbol{x})} \\ &= \frac{1 - F_Z(h(y_k|\boldsymbol{x}))}{F_Z(h(y_k|\boldsymbol{x}))} = \frac{1 - F_L(\vartheta_k - \boldsymbol{x}^T\boldsymbol{\beta})}{F_L(\vartheta_k - \boldsymbol{x}^T\boldsymbol{\beta})}. \end{aligned} \tag{2.5}$$

If we increase the predictor x_j by one unit, holding all other predictors constant, we change $\boldsymbol{x}$ to $\boldsymbol{x}'$ and obtain

$$\begin{aligned} \text{odds}(Y > y_k|\boldsymbol{x}') &= \frac{1 - F_L(\vartheta_k - \boldsymbol{x}'^T\boldsymbol{\beta})}{F_L(\vartheta_k - \boldsymbol{x}'^T\boldsymbol{\beta})} = \frac{1 - F_L(\vartheta_k - \boldsymbol{x}^T\boldsymbol{\beta} - \beta_j)}{F_L(\vartheta_k - \boldsymbol{x}^T\boldsymbol{\beta} - \beta_j)} \\ &= \text{odds}(Y > y_k|\boldsymbol{x}) \exp(\beta_j). \end{aligned} \tag{2.6}$$

Thus, when increasing the predictor x_j by one unit, holding all other predictors constant, the odds to belong to a higher class than y_k change by a constant factor $\exp(\beta_j)$. The parameter β_j is independent of k and can be interpreted as a log-odds ratio:

$$\beta_j = \log\left(\frac{\text{odds}(Y > y_k|\boldsymbol{x}')}{\text{odds}(Y > y_k|\boldsymbol{x})}\right) = \log \text{OR}_{\boldsymbol{x} \to \boldsymbol{x}'} \tag{2.7}$$

[2] $F_L(z) = \dfrac{1}{1 + \exp(-z)}$

It is interesting to notice what happens to f_Z and f_Y when the predictor x_j changes. In Figure 2.5 we can see how the likelihood contributions change if we assume x_j to increase by one unit. The cutpoints $h_{1:4}$ all get shifted by the same distance β. Note that $h_5 = +\infty$. The change in $f_Y(y_k|x)$ is quite complex as can be seen in Figure 2.5 B.

Logistic Regression as a Special Case of Ordinal Regression

When the outcome variable is binary, we have two outcome classes and we need to estimate only one cutpoint h_1. This is the well known case of logistic regression.

2.2.3 Ordinal Neural Network Transformation Models (ONTRAM)

ONTRAMs combine ordinal regression models with deep neural networks through the integration of complex data like images (B) and/or tabular data ($\boldsymbol{x}$). As with the ordinal regression transformation model, the goal is to estimate a transformation function $h(y_k|\boldsymbol{x}, B)$ that now has been extended by some complex data B. ONTRAMs consist of one or several building blocks, that can be combined to complex models. All building blocks are calculated using neural networks and optimized by minimizing the NLL. For a complete description of ONTRAM models, please refer to Kook & Herzog et al. (2020).

Architecture Building Blocks of ONTRAMs

Only the building blocks relevant to this analysis are introduced here. A schematic representation of the different building blocks is shown in Figure 2.6.

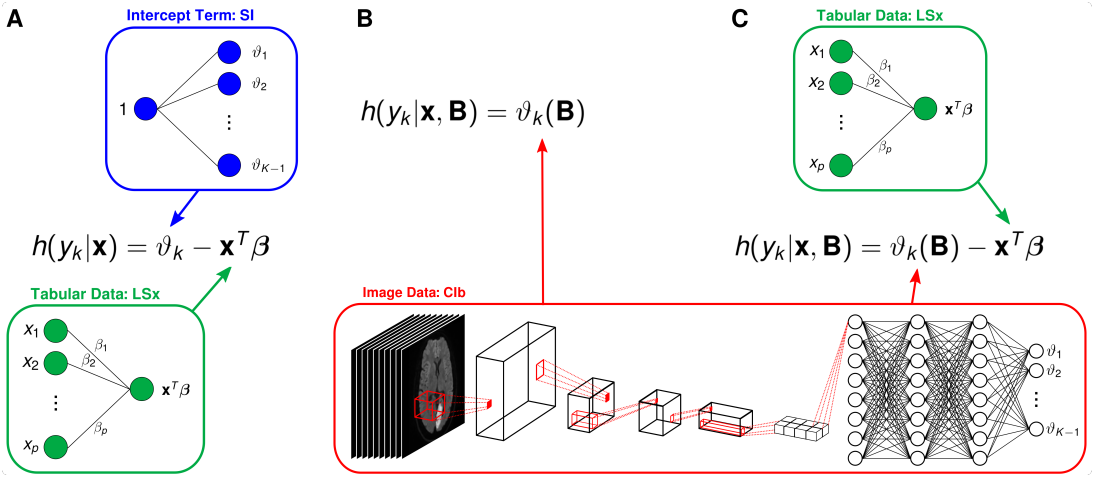


Figure 2.6: Architecture building blocks of ONTRAMs. (A) SI LSx: simple Intercept, linear shift for tabular data. The model is implemented as a single layer neural network. (B) CIb: complex intercept for image, (C) CIb LSx: complex intercept for image, linear shift for tabular data. The complex intercept for the unstructured data is implemented as a 3D CNN and combined with a single layer NN for the LSx.

Simple Intercepts (SI) ϑ_k, $k = 1, \ldots, K-1$ are independent of the input data and can be modeled as a single layer neural network with a single unit input 1 and $K-1$ output units γ_k with linear activation function. To ensure that the transformation function is monotonically increasing, the outputs are transformed as follows:

$$\begin{aligned} \vartheta_k &= \vartheta_1 + \sum_{i=2}^{k} \exp(\gamma_i), \qquad k = 2, \ldots, K-1 \\ \vartheta_0 &= -\infty, \quad \vartheta_1 = \gamma_1, \quad \vartheta_K = +\infty. \end{aligned} \tag{2.8}$$

Linear Shifts (LS) $\boldsymbol{x}^T\boldsymbol{\beta}$ are used for tabular predictors. A single layer neural network is used for modeling, comprising of P input neurons, one for each predictor, and a single output unit with linear activation function. Note that there is no bias term. We are interested in the optimized weights of this network $\beta_{1:P}$, as they can be interpreted as the log-odds ratios described in Section 2.2.2.

- **SI LSx** Model: The simple intercept (SI) can be combined with a linear shift for the tabular data (LSx). It does not depend on any unstructured data and when optimized using the NLL, leads to the same results as an ordinal regression model. In this case, $\boldsymbol{\beta}$ can be interpreted as cumulative log odds-ratios, since we chose $F_Y = F_L$. When using a small enough learning rate and trained for long enough, this model is equivalent to a Proportional Odds Logistic Regression (POLR). When the outcome is binary, this model leads to the same results as a logistic regression (here called LogReg), since both POLR and LogReg minimize the convex NLL. (Figure 2.6 A)

$$h(y_k|\boldsymbol{x}) = \vartheta_k - \boldsymbol{x}^T\boldsymbol{\beta}. \tag{2.9}$$

Complex Intercepts (CI) $\vartheta_k(B)$, $k = 1, \ldots, K-1$, unlike SIs, do depend on the input data. For the current analysis, we model a complex intercept that depends on the image data, by using 3D CNNs as described in Section 2.2.1. As with the SIs, after a linear activation function, the $K-1$ last layer outputs get transformed as described in Equation 2.8.

- **CIb** Model: The complex intercept for image data is the simplest model for complex data. It achieves classification for image data through a 3D CNN with $K-1$ outputs. Since it doesn't depend on tabular data, there will be no coefficients for this model (Figure 2.6 B).

$$h(y_k|B) = \vartheta_k(B). \tag{2.10}$$

- **CIb LSx** Model: Complex intercept for image data, linear shift for tabular data, is the integration of image and tabular data into a single model. It still has high interpretability, since the weights $\boldsymbol{\beta}$ of the linear shift term still can be interpreted as cumulative log odds-ratios (Figure 2.6 C).

$$h(y_k|\boldsymbol{x}, B) = \vartheta_k(B) - \boldsymbol{x}^T\boldsymbol{\beta}. \tag{2.11}$$

2.3 Experiments

The code for all experiments is accessible on github: https://github.com/kilyth/MasterThesis

2.3.1 Models

The data was used to fit three different models introduced in the previous section: SI LSx, simple intercept, linear shift for tabular data, integrating only tabular data (Figure 2.6 A), CIb, complex intercept for image data, integrating only image data (Figure 2.6 B) and CIb LSx, complex intercept for image data, linear shift for tabular data, is the combination of tabular and image data (Figure 2.6 C).

2.3.2 Cross-Validation

5-fold Cross-Validation (CV) was used with a 60%/20%/20% split for train/validation/test, resulting in approximately 300 training, 100 validation and 100 test samples per fold. Each patient is part of the test split exactly once, such that the prediction performance can be calculated over the whole dataset.

2.3.3 Ensembles

Deep ensembles are used to increase prediction accuracy and to produce uncertainty estimates (Lakshminarayanan et al., 2017; Gal, 2016). Each fold is trained 5 times (5 runs) using different starting weights. The overall predicted probability for each patient is calculated as the mean probability per outcome class over 5 runs. An uncertain estimate is expected to show a higher variability between runs.
Partnering CV and ensembles results in a total of 25 networks per model.

2.3.4 Weight Initialization

The weights for the SI LSx and CIb models are initialized with a HeNormal (He et al., 2015) distribution (truncated normal distribution centered around zero) with a different seed per run. To speed up the training, the weights for the CIb LSx models were initialized with the following strategy:

- **Complex Intercept**: the pretrained weights from the CIb model were used for initialization. The corresponding model for each fold and each run was chosen, such that information leakage between folds was avoided.

- **Linear Shift**: a logistic regression for binary outcomes or a proportional odds logistic regression for ordinal outcomes was fit to the training data of each fold. The resulting coefficients were used for the weight initialization of the linear shift term. In each run, random normal noise was added to the coefficients before training to ensure some variability between runs.

2.3.5 Performance Metrics

For the binary outcomes the following evaluation metrics were used: Negative Log Likelihood (NLL), Accuracy, Sensitivity, Specificity and Area under the ROC Curve (AUC)
For the ordinal outcome, the evaluation metrics were NLL, Accuracy, Ranked Probability Score (RPS) and Quadratic Weighted Kappa (QWK). A detailed description of the different metrics can be found in Appendix A.1. While NLL and RPS are the only proper scores, we still included the other performance scores because of their wide use in classification. Proper scoring rules result in honest probabilistic predictions, because they are optimized when the conditional outcome distribution corresponds to the data generating distribution. For a brief discussion of the pitfalls of improper scores, see Kook & Herzog et al. (2020).

2.4 Software

For reproducibility, all code is accessible on github: `https://github.com/kilyth/MasterThesis`

2.4.1 Importation of DICOM Data

The preparation of the DICOM dataset was semi-automatized with a Python script allowing for manual verification of image labels.

2.4.2 ONTRAM

ONTRAMS were implemented in Python 3.6.9, models are written in `Keras` based on `TensorFlow` backend 2.4.0 and trained on a GPU.

2.4.3 Analysis

Analysis and visualization of Results was done in R (R Core Team, 2021), knitr and LaTeX. Logistic regressions were fitted using the function `stats::glm()`, the POLR model was fitted with `MASS::polr()` (Venables and Ripley, 2002).

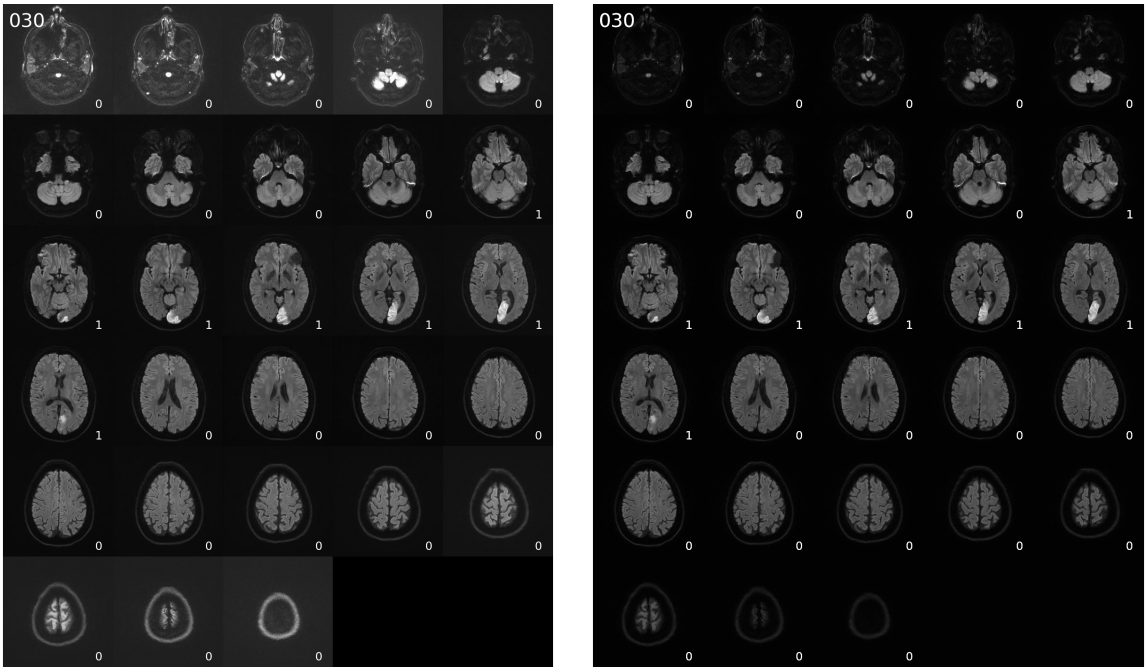
R version and packages used to generate this report:
R version: R version 4.1.0 (2021-05-18)
Base packages: stats, graphics, grDevices, utils, datasets, methods, base
Other packages: xtable 1.8-4, tableone 0.13.0, RColorBrewer 1.1-2, psych 2.1.6, pROC 1.17.0.1, ontram 0.1.0, tensorflow 2.5.0, keras 2.4.0, missForest 1.4, itertools 0.1-3, iterators 1.0.13, foreach 1.5.1, randomForest 4.6-14, MASS 7.3-54, latticeExtra 0.6-29, labelled 2.8.0, gridExtra 2.3, ggpubr 0.4.0, caret 6.0-88, ggplot2 3.3.5, lattice 0.20-44, boot 1.3-28, biostatUZH 1.8.0, survival 3.2-11, knitr 1.33

Chapter 3

Results and Discussion

3.1 Validation of DICOM Data

To build a 3D dataset with constant brightness levels over all slices, we needed to switch from JPG to DICOM format. For the original JPG dataset, the labels for each slice and patient were assigned by a neurologist. To get the equivalent dataset in DICOM format, the labels had to be transferred to the DICOM images. There was one label per slice for a total of 503 patients and 15'191 images. Pairs of JPG and DICOM images were checked visually to make sure that the assignment of labels was done correctly. Figure 3.1 shows the difference between JPG and DICOM image quality.



(a) JPG **(b)** DICOM

Figure 3.1: Patient 030: 28 axial DWI slices. Comparison of JPG and DICOM image format. Changing brightness levels between adjacent slices in the JPG format are seen in Subfigure (a). After changing from JPG to DICOM, brightness levels are constant over all slices. Each slice is labeled with 1 if a stroke lesion is visible (bright area) and with 0 otherwise.

From the original 503 patients and 15'191 images, 1 patient (30 images) was removed because JPG and DICOM didn't show the same patient. 3 patients (90 images) were removed because the MRI sequence was different between JPG and DICOM. 2 patients (53 images) were removed because the order of slices between JPG and DICOM was different. An additional 219 DICOM

images were removed because they were missing in the JPG dataset and thus didn't have an matching image label. After cleaning, the resulting dataset included 497 patients and 14'800 images.
For the validation of the DICOM dataset a previously described 2D CNN was used (Herzog et al., 2020). The binary labels used for training were *Stroke* versus *TIA* on each image slice. There was no aggregation of the predicted labels on patient level. Figure 3.2 shows the results of the 5-fold cross-validation for both datasets. The results were consistent and the prediction performance is very similar between JPG and DICOM, with a slightly better performance of sensitivity for the DICOM data.

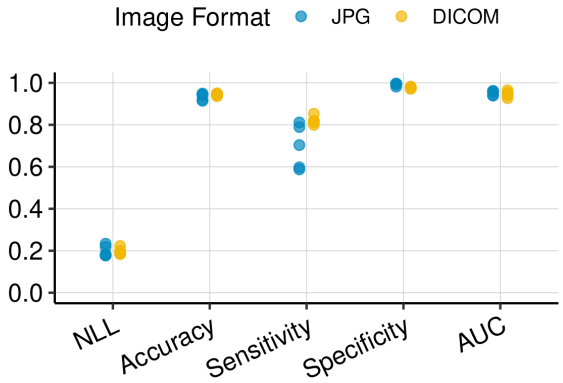


Figure 3.2: Validation of DICOM dataset. JPG and DICOM data was used on a previously described 2D CNN (Herzog et al., 2020). Each point is the result from one fold of the 5-fold cross-validation. NLL: Negative Log Likelihood, AUC: Area Under the ROC Curve.

3.2 Binary Patient Outcome «Event»

To get a prediction per patient and not per image slice, we either need to aggregate the data from a 2D CNN to a patient specific outcome (as done in Kook & Herzog et al. (2020)), or use a 3D CNN from the beginning. Using a 3D CNN has the advantage, that 3D information from the MRI can be fully integrated. In doing so, the main disadvantage is the sample size reduction from 14'800 2D images to 497 3D images.
Before running the ONTRAM models, the dataset was restructured from 2D to 3D. The labels per image slice were no longer used, but only the type of `Event` per patient (*Stroke* vs. *TIA*) was used as label.
For the outcome `Event` three different ONTRAM models were used, as described in the methods in Section 2.3. For each model a 5-fold cross-validation was used, which allowed us to have every patient in the test fold exactly once. The 5 ensemble runs per fold led to 5 predictions for each of the 497 patients in the dataset. To get an overall prediction per patient, the mean of the predicted probabilities over all runs was calculated.

SI LSx Model Simple intercept, linear shift for tabular data.
As described in the methods in Section 2.2.3 the SI LSx model should lead to the same results as the logistic regression LogReg when the outcome is binary, since both models minimize the convex NLL. Prediction performance for the different models can be seen in Figure 3.3. The small differences between the SI LSx and LogReg models are due to either too large learning rates or not enough number of epochs. Accordingly, Figure 3.4 (A) shows that coefficients calculated with the logistic regression are the same as the ones calculated with the SI LSx model. Because the NLL of a logistic regression model has only one minimum, the 5 estimates from each run of the SI LSx model lie exactly on top of each other.

CIb Model Complex intercept for image data.
Since for the outcome `Event` all information is available within the image, we expect this model to perform well. Prediction performance is significantly better than for the SI LSx model as can be seen in Figure 3.3. Since this model doesn't depend on tabular data, we don't get interpretable coefficients.

CIb LSx Model Complex intercept for image data, linear shift for tabular data.
Adding the tabular data to the image model leads to a further decrease of the NLL. The combination of the two types of data does not result in a better prediction accuracy, but leads to a model with higher confidence for the predictions it makes. The weights for the CIb LSx model were initialized with the results from the logistic regression for the linear shift and the pretrained 3D CNN from the CIb model for the complex intercept. Here we allow the image to add additional information and in return, we see that the estimated coefficients shift from their original value (Figure 3.4 A).

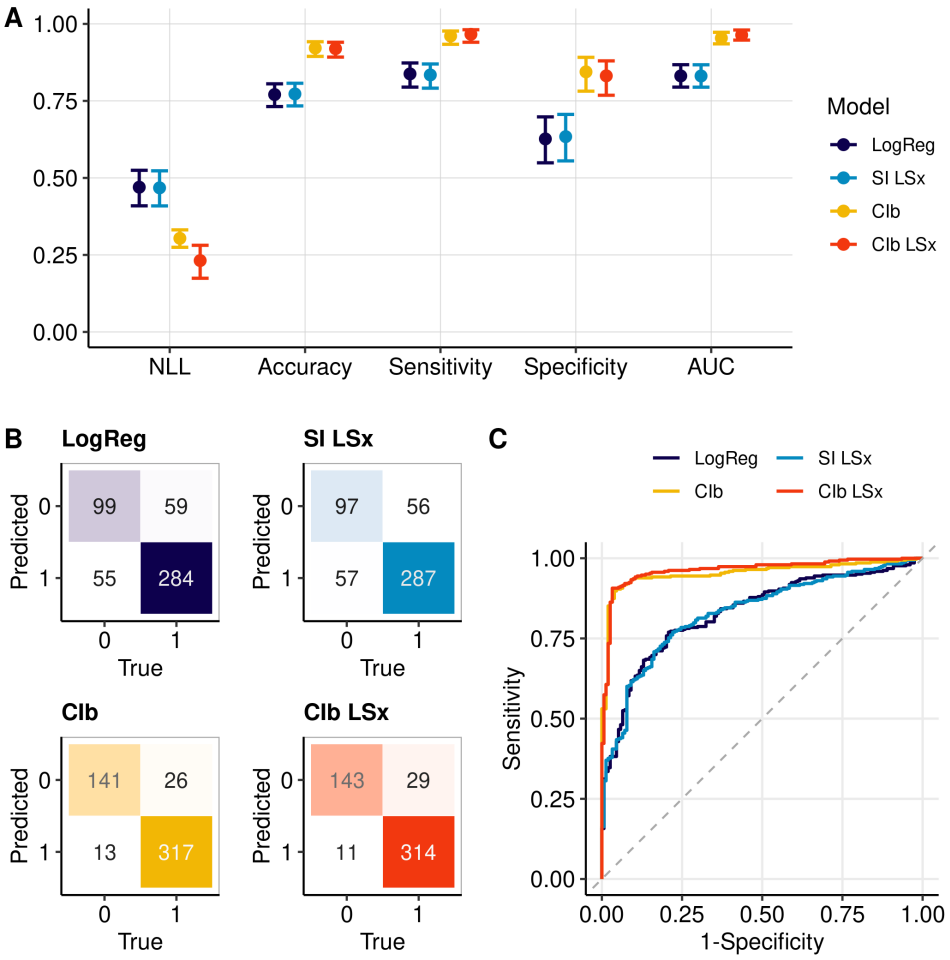


Figure 3.3: Outcome `Event`: (*Stroke* vs. *TIA*)
Prediction performance of the different models. 5-fold CV for the LogReg model and 5-fold CV with 5 ensembles for the ONTRAM models were calculated. (A) Prediction performance and 95% confidence intervals for the different models. (B) Confusion Matrices with labels: 0 = *TIA*, 1 = *Stroke*. (C) ROC curves. LogReg: logistic regression, SI LSx: simple intercept, linear shift for tabular data, CIb: complex intercept for imaging data, CIb LSx: complex intercept for imaging data, linear shift for tabular data. NLL: negative log likelihood, AUC: area under the ROC curve.

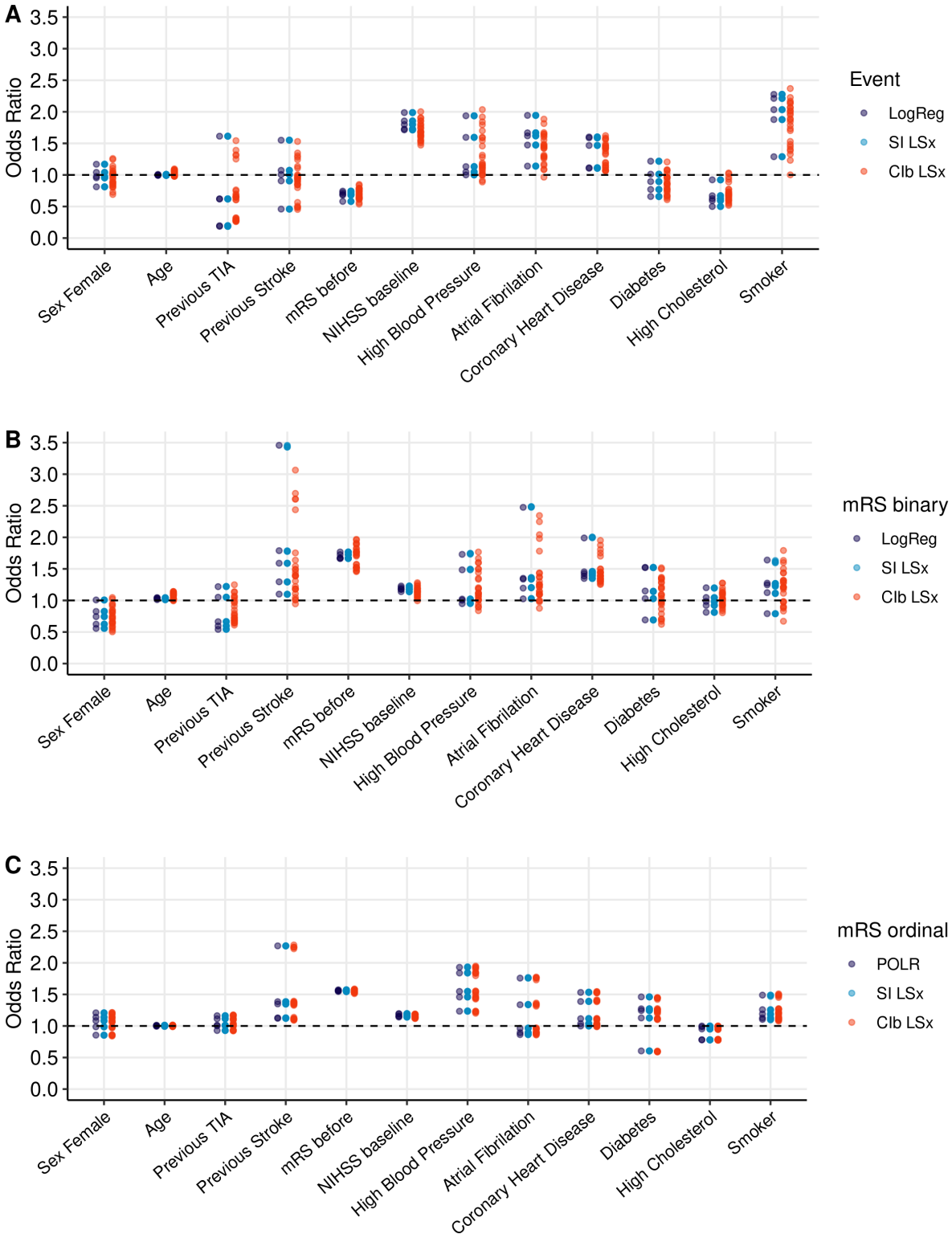


Figure 3.4: Estimated coefficients from all experiments.
(A) Outcome `Event`: *Stroke* vs. *TIA* (B) Outcome `mRS binary`: *good* vs. *bad* (C) Outcome `mRS ordinal`. 5-fold CV for the LogReg and POLR models and 5-fold CV with 5 ensembles for all ONTRAM models were calculated. The coefficients from the SI LSx ensemble runs all lie on top of each other, such that only the five different folds can be seen. These correspond to the coefficients from the LogReg model for the binary outcomes `Event` and `mRS binary` or to the coefficients of the POLR model for outcome `mRS ordinal`. For the CIb LSx models, all 25 estimates can be seen.

Interpretation of Coefficients The estimated coefficients for the LogReg, the SI LSx and the CIb LSx models are shown in Figure 3.4 A. Using 5-fold cross-validation with 5 ensemble runs gives us an idea about the uncertainty of the estimates, although we did not calculate point estimates with confidence intervals. Wide spread values like the risk factor *Smoker* seem to have a higher uncertainty in their estimate than for example the variable *Age*.
Interpretation of the magnitude of the different estimates is only of limited use here and has to be done with great caution. This is due to the fact, that there is a selection and a survival bias in our dataset, which could have a significant influence on the estimated coefficients. As an example, if we had better data to begin with, the interpretation of the coefficient for *Smoker* would be as follows: The odds to have a ischemic stroke and not a TIA when presenting neurological symptoms of a stroke, is $\text{OR}_{\text{smoker}} = \exp(\beta_{\text{smoker}})$ times higher when the patient is a smoker, holding all other predictors constant. See Section 2.2.2 for an explanation about the interpretability of the ordinal transformation models. Since for the risk factor *Smoker* the odds-ratio is larger than one, the odds to have a ischemic stroke instead of a TIA is higher for patients who smoke (holding all other factors constant). This is a result that we would expect, knowing that smoking is bad for our health. However when we look at the risk factor *High Cholesterol*, the interpretation is not as clear. Since the estimate is smaller than one, we would conclude that patients with high cholesterol have lower odds of suffering a ischemic stroke instead of a TIA. One interpretation would be, that patients who have high cholesterol know about it and have already adapted their lifestyles. Another interpretation is, that patients with high cholesterol are underrepresented in our database, since we have a selection bias for patients with good outcomes. Because of this selection and survival bias in our database, it is not possible to draw meaningful conclusions about the effect size of the different coefficients.

3.3 Outcome «mRS binary»

The modified Rankin Scale (mRS) described in Table 2.1 is a measure for the degree of disability for patients who suffered a stroke. The scale contains 7 ordered categories from 0 to 6, ranging from perfect health to death, with small outcomes being better. The same experiment as with outcome `Event` was repeated with outcome `mRS binary`. To this end, the ordinal outcome `mRS ordinal` was dichotomized into *good* (`mRS ordinal` ≤ 2) and *bad* (`mRS ordinal` > 2). The 29 patients with missing outcomes were removed, leading to a dataset with 468 patients.

SI LSx Model Simple intercept, linear shift for tabular data.
As with the binary outcome `Event`, we expect the SI LSx and the LogReg models to lead to the same results, if we use a small enough learning rate and train the network for many epochs. Prediction performance for the different models can be seen in Figure 3.5. Again there is little variability between the two models, presumably due to suboptimal training parameters. All coefficients from all five ensemble runs lie on top of each other and only show minimal deviations from the coefficients from the LogReg model (Figure 3.4 B).

CIb Model Complex intercept for image data.
For this outcome the information is not directly detectable in the image as it was the case with the outcome `Event`. To train a CNN to extract features for an outcome that lies in the future is a much harder task. Therefore the CIb model is expected to have lower prediction performance than for the outcome `Event`. This could indeed be observed as shown in Figure 3.5. The CIb model performs not as good the tabular data alone.

CIb LSx Model Complex intercept for image data, linear shift for tabular data.
Combining the image and the tabular data improves the prediction performance compared to

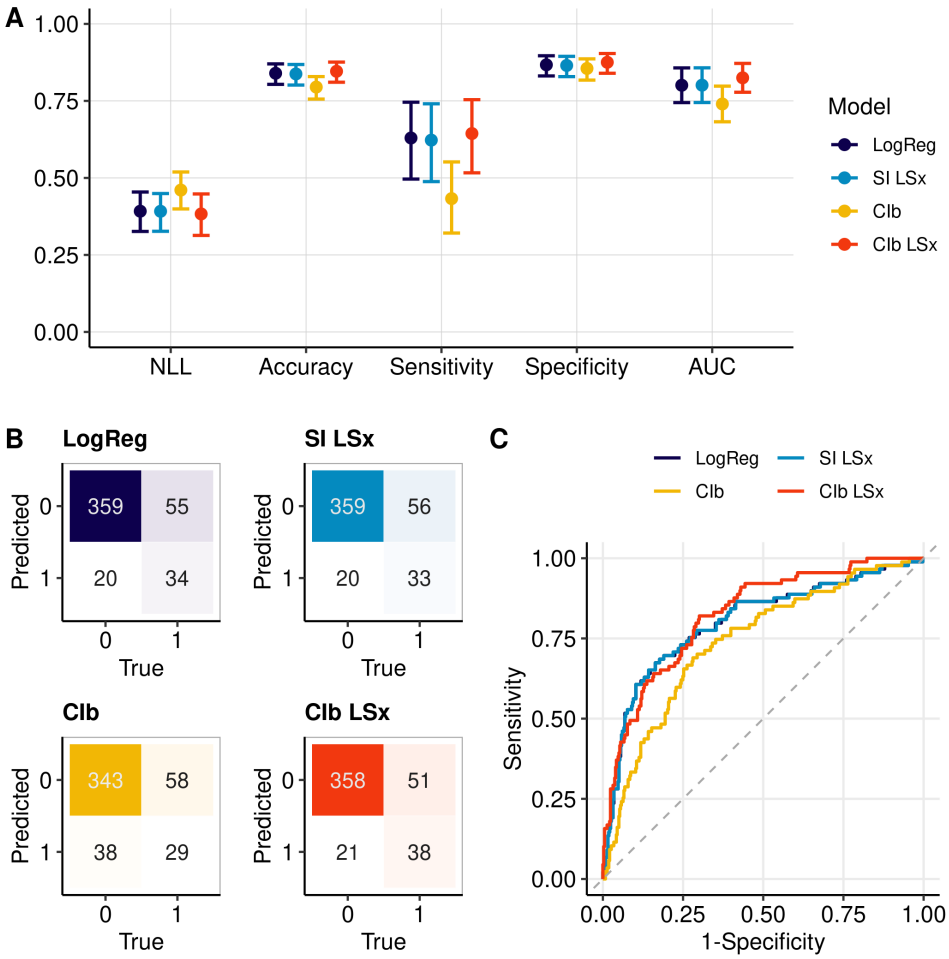


Figure 3.5: Outcome `mRS binary`: (*good* vs. *bad*)
Classification performance of the different models. 5-fold CV for the LogReg model and 5-fold CV with 5 ensembles for the ONTRAM models were calculated. (A) Prediction performance and 95% confidence intervals for the different models. (B) Confusion Matrices with labels: 0 = *good*, 1 = *bad*. (C) ROC curves. The curves for LogReg and SI LSx lie exactly on top of each other. LogReg: logistic regression, SI LSx: simple intercept, linear shift for tabular data, CIb: complex intercept for imaging data, CIb LSx: complex intercept for imaging data, linear shift for tabular data. NLL: negative log likelihood, AUC: area under the curve.

the CIb model, with a slight tenancy to perform even better than the tabular data alone (see Figure 3.5). Especially, since we initiate the weights of the CIb LSx with the coefficients from the logistic regression and the previously trained CIb (see Section 2.3.4), we expect the more complex model to perform at least as good as the better performing of the simpler models.
Again we can see that when adding the image to the tabular data, estimated coefficients shift from their original value (Figure 3.4 B).

3.4 Outcome «mRS ordinal»

We ran the same experiments as before, this time not using a binary, but an ordinal outcome. The mRS score contains 7 ordered categories ranging from 0 to 6, with small outcomes being better. For ordinal outcomes, we cannot use the same classification metrics than for the binary outcomes. The evaluation metrics used here were NLL, Accuracy, Ranked Probability Score (RPS) and Quadratic Weighted Kappa (QWK). While NLL and RPS are proper scores, the

other performance metrics are still shown here, because of their wide use in classification. For a description of the used metrics, please refer to Appendix A.1 and Kook & Herzog et al. (2020).

SI LSx Model Simple intercept, linear shift for tabular data.
The SI LSx model was compared to a Proportional Odds Logistic Regression (POLR). As for the previous experiments, the SI LSx model should lead to the same results as the POLR, if trained with enough epochs and small enough learning rate. Again, there is a small variability between the two models, that can be explained by suboptimal training parameters of the SI LSx model. A total of 3 patients were classified differently between the two models. Interestingly, classes 3 to 5 were never predicted, although there are 60 patients with mRS 3 to 5 at three months after the ischemic event (Figure 3.6 B).
All SI LSx coefficients from each run lie on top of each other (Figure 3.4 C) and match the coefficients from the POLR model.

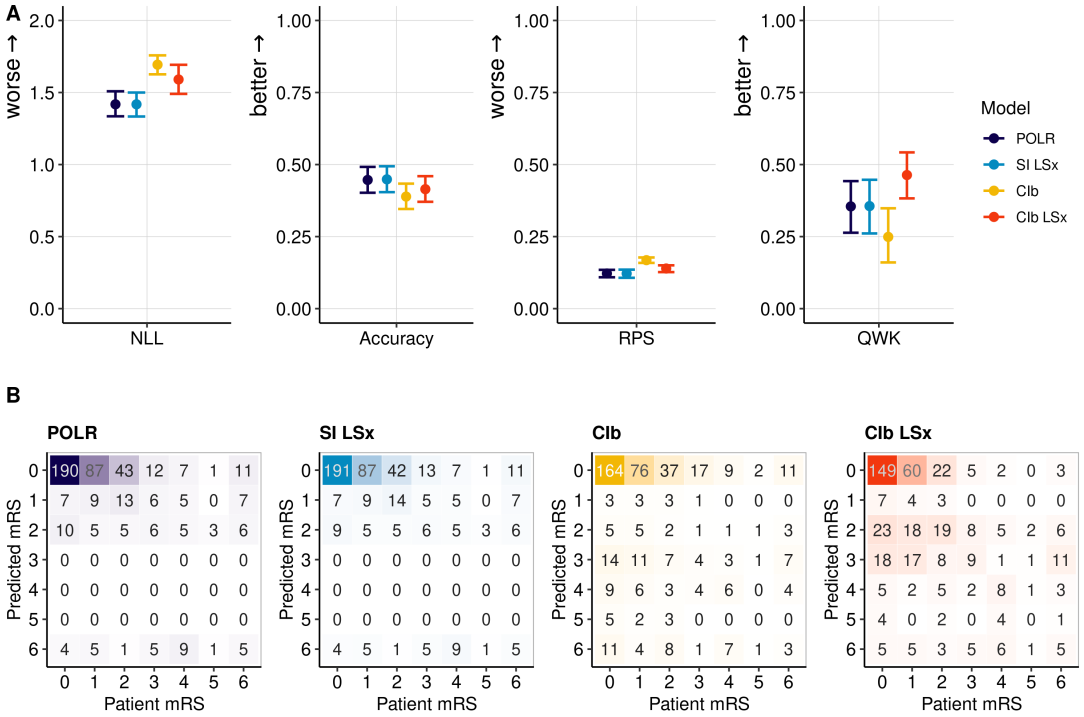


Figure 3.6: Outcome `mRS ordinal`
(A) Prediction performance and (B) confusion matrices for the different models. All models were calculated using a 5-fold CV and with 5 ensemble runs for the ONTRAM models. POLR: proportional odds logistic regression, SI LSx: simple intercept, linear shift for tabular data, CIb: complex intercept for imaging data, CIb LSx: complex intercept for imaging data, linear shift for tabular data, NLL: negative log likelihood, RPS: ranked probability score, QWK: quadratic weighted kappa.

CIb Model Complex intercept for image data.
As already observed with the dichotomized outcome `mRS ordinal`, the prediction performance of the CIb model is worse than the SI LSx model for all performance metrics. We try to predict an outcome that lies in the future compared to when the MRI was taken, which makes this classification a much hard task. In contrast to the predictions of the SI LSx model, classes 3 to 5 are frequently predicted.

CIb LSx Model Complex intercept for image data, linear shift for tabular data.
For the proper scores NLL and RPS and the improper Accuracy, the CIb LSx model gains in performance compared to the CIb model, however it is not as good as the SI LSx model. Since all ONTRAM models are trained with a NLL loss, we would expect the more complex CIb LSx model to perform at least as good as the better performing simpler model, however this is not the case here. This might be due to initialization of the weights, such that the model reaches a different local minimum, or because of bad combination of learning rate, batch size and number of epochs. Interestingly, when we look at the QWK, the CIb LSx model gains in performance over all other models. In contrast to the other performance metrics, the QWK takes into account the degree of miss-classification, penalizing classifications that are further away from the true value. As observed with the previous outcomes, when combining tabular and imaging data, the coefficients start to deviate from the values given by the LogReg model. When we compare the coefficients from the two experiments with outcome `mRS`, we see that the coefficients lie within a similar range for both the ordinal and the dichotomized outcome (Figure 3.4 B and C).

Interpretation of Coefficients Again, interpretation of the magnitude of the different estimates is only of limited use here and has to be done with great caution. Compared to binary outcomes, the interpretation of the coefficients has to be adapted slightly for ordinal outcomes: If the patient is a smoker, holding all other factors constant, the odds to belong to a higher class than y_k (0 to k vs. $k+1$ to K) change by a constant factor $\text{OR}_{\text{smoker}} = \exp(\beta_{\text{smoker}})$. See Section 2.2.2 for an explanation about the interpretability of the ordinal transformation models.

Chapter 4

Summary and Outlook

It is well known that functional outcome after ischemic stroke depends not only on *time to treatment* but also on patient characteristics and risk factors like age or diabetes (Weimar et al., 2002). When we want to predict the functional outcome based on medical data, we want to build a model that integrates these variables and predicts the outcome as accurately as possible. This can be done with well known statistical models, like the logistic regression for binary outcomes (LogReg), or the proportional odds logistic regression (POLR) for ordinal outcomes. But in the case of stroke, the size and location of the ischemia is very much relevant to the outcome as well. We are therefore in need for a model that is able to make predictions, not solely based on tabular data, but is capable to integrate image data as well. This can be done with the recently implemented ONTRAMs described in Kook & Herzog et al. (2020). In this thesis, I used three different ONTRAMs with different complexity and interpretability, on three different outcomes each – two binary outcomes and one ordinal – and compared them to the classical logistic regression and proportional odds logistic regression models.
For the ONTRAM to work, a well performing CNN is needed, capable to extract relevant image features from the data. While 2D CNNs are widely established and show astonishing performance in image classification tasks, 3D CNNs are rare. The downsides of using 3D over 2D data are the need for larger computational power to fit 3D models and the vast reduction in size of the dataset. For example, the dataset used for this thesis was reduced from 14'800 2D images with image labels to 497 3D images with patient labels. Using 3D information from MRIs however has important advantages: we can extract three-dimensional information from our data and we get rid of the need to aggregate several image predictions to a patient level prediction. Preparing a dataset with 3D images was a large part of this thesis. Because of inconsistent brightness levels in the previously used JPG dataset, we decided to re-import all data from DICOM format and transfer image labels from the original dataset.
To make use of the 3D image data within an ONTRAM model, a 3D CNN was developed. The resulting network was inspired by other groups who built 3D CNNs for medical data (Kan et al., 2021; Zunair et al., 2020). The resulting network was not very deep, but contained only 5 convolutional layers and two dense layers. Even so, it showed good classification performance for stroke detection of 92% accuracy and an AUC of 0.95 and was thus able to extract relevant image features from MRIs. The same architecture was then used to predict patient outcome three months after stroke.

4.1 ONTRAM

Three ONTRAM models were implemented: SI LSx (simple intercept, linear shift for tabular data) taking only tabular data as input, CIb (complex intercept for image data) using only image data and CIb LSx (complex intercept for image data, linear shift for tabular data) for the combi-

nation of tabular and image data as described in Section 2.3. During training of the ONTRAMs, the NLL was minimized by gradient descent. When using a batch size equal to the size of the training data, the SI LSx model is no longer optimized with stochastic gradient descent, but calculates the exact gradient over the whole training data. Since this is a convex problem, we can reach the global optimum if we use a small learning rate and train the network for many epochs. The global optimum is equivalent to the solution of the LogReg model if the outcome is binary, or the solution of a POLR model for ordinal outcomes. It could be shown that the neural network based SI LSx models yield the same results as the corresponding LogReg and the POLR models that were fitted in R.
The CIb model only uses image data and demands high computational power. Image augmentation and 3D architecture make the computations slow, such that one model needs up to four hours for 200 epochs. The CIb LSx model allows for the integration of unstructured data besides tabular data and at the same time estimates coefficients for the later, that can be interpreted as log-odds ratios as described in Section 2.2.2.
All models were evaluated with 5-fold cross-validation. Each patient was part of the test dataset exactly once, such that there was a predicted outcome for each patient. Additionally, for every fold, 5 ensemble runs were calculated, resulting in 25 runs per model. Ensembles allow to get better prediction performance (Lakshminarayanan et al., 2017) and also give us an idea about the uncertainty and variability of the calculated coefficients.

4.2 Experiments

For the first experiment, a binary outcome `Event` (*Stroke* vs. *TIA*) was chosen. The label is based on the patient's image, if there is a stroke lesion visible on the MRI. It is not surprising that the CIb model performs well for this outcome, since the whole information about the label is based on the image. The SI LSx model, based on tabular data only, has a significantly lower prediction performance than the models containing image information. The small variation between results from the SI LSx and the LogReg model is presumably due to suboptimal training parameters for the SI LSx model. The more complex CIb LSx model, integrating tabular and image data, has significantly better prediction performance than the tabular data alone. Looking at the NLL, it performs even better than the CIb model, and gives us an idea about the influence of the tabular data.
The second experiment had the same implementation as the first one, but used the dichotomized modified Rankin Scale `mRS binary` (Section 2.1.1) as patient outcome. Predicting the functional outcome is a harder task than discriminating between TIA and stroke because the information about the outcome in not readily available within the image and lies in the future compared to the date when the image was acquired. Therefore we didn't expect the CIb model to perform as well as in the first experiment. The tabular data alone had a better prediction performance than the image alone. However when combining the tabular data with the image data in the CIb LSx model, the prediction performance is as good as with the simpler SI LSx model (Figure 3.5).
For the third experiment, the ordinal outcome `mRS ordinal` containing 7 categories was used. The 3D CNN architecture was the same as in the other experiments and the NLL was again used as loss function. Evaluating ordinal classification performance is not as straight forward as for binary outcomes. Accuracy, although shown here as a performance metric, is not recommended for assessing classification performance for ordinal outcomes. It only takes into account true and false predictions without considering the predicted probabilities of the outcome. The NLL and RPS are both proper scoring rules (see Gneiting and Raftery (2007) and Kook & Herzog et al. (2020) for a discussion about scoring rules) but contradict QWK in the performance evaluation for this experiment. The QWK is a performance measure that was developed for ordinal classification, penalizing miss-classifications farther away from the observed class. However it is not a

proper scoring rule and was not used for optimization in our experiments.

4.3 Outlook

Since our models were trained with a NLL loss, we would expect the more complex CIb LSx model to have a similar NLL than either the SI LSx or the CIb model – whichever performed better. If the image does not contribute to a better prediction performance, we would expect the ONTRAM to set the weights from either CIb or LSx to zero, such that it reaches at least the same NLL than the corresponding simpler model. However, this was not the case for the third experiment, where the CIb LSx model performed worse than the simpler SI LSx model. This might be due to initialization of the weights, such that the model reaches a different local minimum, or because of bad combination of learning rate, batch size and number of epochs. This is an issue that needs further investigation.
The models based on tabular data only (SI LSx, LogReg or POLR) are much faster to train than the models including unstructured data. Thus, it only makes sense to include image data, when the performance of the model can be significantly increased. If this is the case, using the more complex CIb LSx model over the simpler CIb model is preferable. It yields interpretable estimates about the influence of the tabular predictors, while the CIb model only predicts the outcome. Therefore the CIb LSx model is no longer a *black box*, but allows for the estimation of interpretable coefficients, which makes it possible to gain important knowledge about the data.
Another open question is the estimation of confidence intervals for the calculated coefficients. Even though the combination of cross-validation and ensembles gives us an idea about the variability and uncertainty of the estimates, the aggregation to a valid confidence interval is not obvious. It is in general an unsolved problem how to set valid confidence intervals in neural network based model that combine single-layer and deep neural networks to estimate model parameters. Since a single training run for the CIb LSx model takes up to 4 hours, bootstrap confidence intervals cannot be computed as for example done in random forest classification.
For this thesis, the focus of the different models lied on prediction of the outcome, since there is limited use in the interpretation of the magnitude of the coefficients. As explained in the Methods 2.1 the data was not collected in a systematic way nor do we have a consecutive patient cohort, resulting in a selection bias. It is therefore not possible to draw meaningful conclusions about the effect size of the different coefficients. In a future study, it would be important to define clear inclusion criteria and avoid selection bias in order to make interpretation of the results possible. Conducting a randomized clinical trial with systematic collection of all confounders would lead to improved data quality. In addition to patient characteristics and risk factors, it would then be possible to add treatment variables to our model in order to quantify a treatment effect. Let's assume a patient comes into the emergency room with an ischemic stroke. A properly trained ONTRAM model would allow for selection of the best treatment for a specific patient, depending on the patients baseline variables, risk factors and the MRI.

Bibliography

Bonati, L. für das Swiss Stroke Registry Steering Committee (2015). Minimaler Datensatz Swiss Stroke Registry. https://www.neurovasc.ch/fileadmin/files/arbeitsgruppen/SSR_MinimalerDatensatz_2015-3-18.pdf, accessed 13-July-2021. 5

Brown, L. D., Cai, T. T., and DasGupta, A. (2001). Interval estimation for a binomial proportion. *Statistical Science*, 16(2):101–133. https://doi.org/10.1214/ss/1009213286. 31, 32

Davison, A. C. and Hinkley, D. V. (1997). *Bootstrap Methods and Their Applications*. Cambridge University Press, Cambridge. ISBN 0-521-57391-2. 31, 32

Gal, Y. (2016). Uncertainty in deep learning. PhD thesis, University of Cambridge. https://mlg.eng.cam.ac.uk/yarin/thesis/thesis.pdf. 14

Gneiting, T. and Raftery, A. E. (2007). Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378. https://doi.org/10.1198/016214506000001437. 26

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press. http://www.deeplearningbook.org. 3

Haile, S. R., Held, L., Meyer, S., Rueeger, S., Rufibach, K., and Schwab, S. (2019). *biostatUZH: Misc Tools of the Department of Biostatistics, EBPI, University of Zurich*. R package version 1.8.0/r82, https://R-Forge.R-project.org/projects/ebuzh/. 31, 32

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. https://openaccess.thecvf.com/content_iccv_2015/html/He_Delving_Deep_into_ICCV_2015_paper.html. 14

Herzog, L., Murina, E., Dürr, O., Wegener, S., and Sick, B. (2020). Integrating uncertainty in deep neural networks for MRI based stroke analysis. *Medical Image Analysis*, 65:101790. https://doi.org/10.1016/j.media.2020.101790. 7, 18

Hothorn, T., Kneib, T., and Bühlmann, P. (2014). Conditional transformation models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(1):3–27. https://doi.org/10.1111/rssb.12017. 9

Kan, M., Aliev, R., Rudenko, A., Drobyshev, N., Petrashen, N., Kondrateva, E., Sharaev, M., Bernstein, A., and Burnaev, E. (2021). Interpretation of 3D CNNs for brain MRI data classification. In *Recent Trends in Analysis of Images, Social Networks and Texts*, pages 229–241. Springer International Publishing. ISBN 978-3-030-71214-3, https://doi.org/10.1007/978-3-030-71214-3_19. 8, 25

Kingma, D. P. and Ba, J. (2017). Adam: A method for stochastic optimization. arXiv. https://arxiv.org/abs/1412.6980. 8

Kook, L., Herzog, L., Hothorn, T., Dürr, O., and Sick, B. (2021). Deep and interpretable regression models for ordinal outcomes. arXiv. https://arxiv.org/abs/2010.08376. 4, 10, 12, 14, 18, 23, 25, 26, 32

Kook, L. H. (2021). *ontram: Ordinal transformation model neural networks*. R package version 0.1.0, https://github.com/LucasKookUZH/ontram-pkg. 32

Lakshminarayanan, B., Pritzel, A., and Blundell, C. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. arXiv. https://arxiv.org/abs/1612.01474. 14, 26

Nielsen, M. A. (2015). *Neural Networks and Deep Learning*, volume 25. Determination Press San Francisco, CA. http://neuralnetworksanddeeplearning.com/. 3

R Core Team (2021). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/. 15

Robin, X., Turck, N., Hainard, A., Tiberti, N., Lisacek, F., Sanchez, J.-C., and Müller, M. (2011). pROC: an open-source package for R and S+ to analyze and compare ROC curves. *BMC Bioinformatics*, 12:77. https://doi.org/10.1186/1471-2105-12-77. 31

Stekhoven, D. J. and Bühlmann, P. (2012). MissForest — non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1):112–118. https://doi.org/10.1093/bioinformatics/btr597. 6

Swiss Neurological Society (2019). Schlaganfall. https://www.swissneuro.ch/view/Content/schlaganfall accessed 15-April-2021. 4

Venables, W. N. and Ripley, B. D. (2002). *Modern Applied Statistics with S*. Springer, New York, fourth edition. ISBN 0-387-95457-0, http://www.stats.ox.ac.uk/pub/MASS4/. 15

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and SciPy 1.0 Contributors (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272. https://doi.org/10.1038/s41592-019-0686-2. 8

Weimar, C., Ziegler, A., König, I. R., and Diener, H.-C. (2002). Predicting functional outcome and survival after acute ischemic stroke. *Journal of neurology*, 249(7):888–895. https://doi.org/10.1007/s00415-002-0755-8. 25

Wikipedia contributors (2021). Stroke — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Stroke&oldid=1032912414, accessed 12-July-2021. 4

Wilson, J. L., Hareendran, A., Grant, M., Baird, T., Schulz, U. G., Muir, K. W., and Bone, I. (2002). Improving the assessment of outcomes in stroke: use of a structured interview to assign grades on the modified rankin scale. *Stroke*, 33(9):2243–2246. https://doi.org/10.1161/01.STR.0000027437.22450.BD. 6

Zunair, H., Rahman, A., Mohammed, N., and Cohen, J. P. (2020). Uniformizing techniques to process CT scans with 3D CNNs for tuberculosis prediction. In *Predictive Intelligence in Medicine*, pages 156–168. Springer International Publishing. ISBN 978-3-030-59354-4, https://doi.org/10.1007/978-3-030-59354-4_15. 8, 25

Appendix A

Appendix

A.1 Evaluation Metrics

A.1.1 Evaluation Metrics for Binary Outcomes

For binary outcomes, the following evaluation metrics were used:

Negative Log Likelihood (NLL): Always given as the mean NLL per observation, i.e. the NLL divided by the number of samples, which is thus equivalent to the crossentropy. Reversed quantile confidence intervals were calculated using the R package `boot` (Davison and Hinkley, 1997).

$$\begin{aligned} NLL &= -\frac{1}{n}\sum_{j=1}^{n}(y_j \log(p_1(x_j))) + (1-y_j)log(1-p_1(x_j)) \\ &= -\frac{1}{n}\left[\sum_{\text{j for y = 0}} \log(p_0(x_j)) + \sum_{\text{j for y = 1}} \log(p_1(x_j))\right] \end{aligned}$$

Accuracy[1]:

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

Sensitivity

$$\text{Sensitivity} = \frac{TP}{TP+FN}$$

Specificity

$$\text{Specificity} = \frac{TN}{TN+FP}$$

For accuracy, sensitivity and specificity, Wilson confidence intervals were calculated with the R function `biostatUZH::confIntProportion` (Brown et al., 2001; Haile et al., 2019).

Area Under the ROC Curve (AUC)
ROC curves, AUC and corresponding bootstrap confidence intervals were calculated using the R Package `pROC` (Robin et al., 2011).

[1] TP: true positive, TN: true negative, FP: false positive, FN: false negative

A.1.2 Evaluation Metrics for Ordinal Ourcomes

For ordinal outcomes, the evaluation metrics listed below were used. For a more detailed description of ordinal evaluation metrics refer to Kook & Herzog et al. (2020).

Negative Log Likelihood (NLL): Always given as the mean NLL per observation, i.e. the NLL divided by the number of samples, which is thus equivalent to the crossentropy. Reversed quantile confidence intervals were calculated using the R package `boot` (Davison and Hinkley, 1997).

$$NLL = -\frac{1}{n}\left[\sum_{\text{j for y = 0}} \log(p_0(x_j)) + \sum_{\text{j for y = 1}} \log(p_1(x_j)) + \cdots + \sum_{\text{j for y = K-1}} \log(p_{k-1}(x_j))\right]$$

Accuracy: calculated with the R function `biostatUZH::confIntProportion` with Wilson confidence intervals (Brown et al., 2001; Haile et al., 2019).

$$\text{Accuracy} = \frac{\text{number of correct classifications}}{\text{total number of classifications}}$$

Ranked Probability Score (RPS)

$$\text{RPS}(p; y) = \frac{1}{K-1}\sum_{k=1}^{K}\left(\sum_{j=1}^{k} p_j - \sum_{j=1}^{k} e_j\right)$$

where K is the number of classes, p_j and e_j are the predicted and actual probability of class j. e_j is given by the j^th entry of the one-hot encoded outcome. The RPS was calculated in R using the function `ontram::rps()` (Kook, 2021). Reversed quantile confidence intervals were calculated using the R package `boot` (Davison and Hinkley, 1997).

Quadratic Weighted Kappa (QWK)

$$\kappa_w = \frac{p_{obs}(w) - p_{exp}(w)}{1 - p_{exp}(w)}$$

where $p_{obs}(w)$ and $p_{exp}(w)$ are the sum of observed and expected probabilities for each combination of true versus predicted class. The QWK and its confidence intervals was calculated in R using the function `biostatUZH::confIntKappa()` (Haile et al., 2019).

A.2 List of Terms

ANN	Artificial Neural Network
AUC	Area Under the ROC Curve
CDF	Cumulative Distribution Function
CHD	Coronary Heart Disease
CNN	Convolutional Neural Network
CV	Cross-Validation
DICOM	National Institutes of Health Stroke Scale
DL	Deep Learning
DWI	Diffusion Weighted MRI
EEG	Electroencephalography
ICH	Intra-Cerebral Hemorrhage
LogReg	Logistic Regression
MRI	Magnetic Resonance Image
mRS	modified Rankin Scale
NIHSS	National Institutes of Health Stroke Scale
NLL	Negative Log Likelihood
ONTRAM	Ordinal Neural Network Transformation Model
PDF	Probability Density Function
POLR	Proportional Odds Logistic Regression
QWK	Quadratic Weighted Kappa
ReLU	Rectified Linear Unit
ROC	Receiver Operating Characteristic
RPS	Ranked Probability Score
TIA	Transient Ischemic Attack