UNIVERSITY OF TARTU
Faculty of Science and Technology
Institute of Computer Science
Computer Science Curriculum

Siim Kurvits

Prediction Models of Ischemic Stroke Using Deep Neural Networks

Master's Thesis (30 ECTS)

Supervisor(s): Toomas Haller, PhD
Ardi Tampuu, PhD

Tartu 2021

Prediction models of ischemic stroke using deep neural networks

Abstract: The ischemic stroke is one of the leading causes of death worldwide. Although, there are many known risk factors for the disease the growing amount of electronic medical data available gives opportunities for creating novel models for personal risk prediction. Usage of deep neural network (DNN) for developing such models can offers many benefits such as potential to encode multiple types of data, less feature selection and engineering required, and sometimes even an increased prediction accuracy. This Thesis focuses on developing a model for ischemic stroke prediction using electronic health record (EHR) data. I show that TabNet, a state-of-the art DNN architecture for tabular data analysis outperforms a simpler method, the FastAI tabular learner. Still, neither of the DNN methods achieved better results than the Random Forest. The ensemble models using Random Forest and DNN models were tested but only a small increase in the performance was achieved compared to the singular model. These results indicate that an ensemble-based methods such as Random Forest is sufficient for the data used. Nevertheless, with increased number of features and addition of more complex data types methods such as TabNet could still become valuable. All models developed resulted with high prediction power for ischemic stroke. This indicates that personal risk predictions for ischemic stroke can be given and the clinical utility of the models should be evaluated further.

Keywords: machine learning, neural networks, ischemic stroke

CERCS: P176

Tehisnärvivõrkudel põhinevad isheemilise insuldi ennustusmudelid

Lühikokkuvõte: Isheemiline insult on üks maailma juhtivatest surma põhjustest. Kuigi on teada mitmed isheemilise insuldi riskifaktoreid, siis aina kasvav elektrooniliste meditsiiniandmete hulk loob võimalusi uudsete personaalsete riskimudelite arendamiseks. Süvanärvivõrkudel põhinevatel mudelitel on mitmeid eeliseid - võimalus kodeerida erivaid andmetüüpe, väiksem vajadus tunnuste eelnevaks valimiseks ja modifitseerimiseks, mõnikord ka suurenenud mudeli täpsus. Käesoleva töö eesmärgiks on isheemilise insuldi ennustusliku mudeli loomine kasutatades elektroonilisi terviseandmeid. Süvanärvivõrkudena kasutatakse TabNet ja FastAI tehnoloogial baseeruvaid mudeleid, millest esimene saavutab parema täpsuse. Siiski jäävad mõlemad süvanärvivõrkudel baseeruvad mudelid alla juhumetsa metoodikal loodud mudelile. Mitme mudeli kooskasutus andis väikese edu võrreldes ühe mudeli kasutamisele. Seega on juhumetsa kasutamine käesolevas töös uuritava andmete jaoks piisav, kuid andmete lisandumisel võivad süvanärvivõrkudel põhinevad mudelid osutuda kasulikeks. Kõik töö raames loodud mudelid ennustasid isheemilist insulti hästi ja mudelite kliinilise kasutuse võimalusi tuleks edasi uurida.

Võtmesõnad: masinõpe, tehisnärvivõrgud, isheemiline insult

CERCS: P176

Contents

List of Abbreviations

ANN artificial neural network

ATC Anatomical Therapeutic Chemical

AUC area under the curve

CKD-EPI Chronic Kidney Disease Epidemiology Collaboration

DNN deep neural network

EHR electronic health record

ICD-10 International Classification of Diseases version 10

LOINC Logical Observation Identifier Names and Codes

ML Machine Learning

ROC receiver operating characteristic curve

STACC Software Technology and Applications Competence Center

WHO World Health Organization

1 Introduction

Stroke is one of the leading causes of death worldwide and can be categorized as hemorrhagic and ischemic [1]. Hemorrhagic stroke originates from a blood vessel burst that causes bleeding in the brain. Ischemic stroke is caused by a blockage in a artery supplying the brain with blood resulting in brain cell damage or death. This Thesis is motivated by the increasing use of deep learning methods for genomic, proteomic and metabolomic data integration in precision medicine (personalized medicine) [2, 3]. Disease risk model is a method used in precision medicine to estimate the individuals absolute risk for the disease with the assessment of risk factors. These disease risk models can be a valuable resource for better public health policy while giving individual a personalized feedback about the disease risk [4]. There are many known risk factors for stroke, such as age and genetic predisposition [5]. These factors have been used to propose multiple stroke prediction models [6]. Some prediction models have been developed for patients with preexisting cardiovascular conditions while also being accurate for patients without the condition [7]. Using EHR data for stroke prediction by DNN in Estonian population is a novel venue. There are several methods for evaluating disease risks and finding disease risk factors. The prediction methods used in this Thesis are Machine Learning (ML) based binary classification models. Random Forest is chosen as the benchmark model to compare against the DNN models. The usage of DNN for tabular data is rather uncommon, as traditional ML methods have shown to perform equally well on this type of data, but in recent years advancements in the field have been made. Models like TabNet [8] and FastAI tabular learner [9] are designed for tabular data type. The aim of the Thesis is to test these DNN methods on real life Estonia's EHR data. However, EHR data poses challenges for DNN methods due to factors such as low signal to noise ratio, analytical variance and complex data integration requirements [2]. **The aim of this Thesis is to evaluate the available DNN architectures for predicting ischemic stroke using EHR data on the Estonian population.** The usage of DNN based models has many potential benefits.

Firstly, one can reasonably hope an increase in classification performance. Neural network based models have been shown to have excellent performance for metabolomic prediction models and outperform the other ML methods [10, 11]. Some Logical Observation Identifier Names and Codes (LOINC) codes mined from EHR that are used in this Thesis represent metabolite measurements while the others describe various biological processes.

Secondly, the architecture of DNNs will potentially allow to efficiently encode multiple types of data [8]. For the EHR data this could be used to add medical images and genetic information to the model. For this Thesis the LOINC codes are combined with International Classification of Diseases version 10 (ICD-10) diagnoses. An embedding layer is used on ICD-10 to capture the semantics of the codes.

Thirdly, the DNN models could alleviate the need for feature selection and engineer-

ing. In a multi-omics analysis the internal abstractions developed by the deep learning on each individual Omics data set increases the signal-to-noise ratio and eliminate the need for manual normalization of each individual Omics data type [2, 12, 13, 14]. As the EHR data contains large selection of different features this can decrease the model development time.

The FastAI Python library is used in the Thesis as it enables domain experts like medical researches to start using the pre-existing neural network architectures without the need of detailed implementation [15]. The library is not popular among metabolomics researchers as it is not even mentioned in the long list of publicly available resources for creating a DNN in a metabolomics article describing deep learning tools in year 2020 [3].

The research questions of this Thesis are the following:

Q1. Can the occurrence of ischemic stroke be predicted using pre-existing EHR data using 3 ML methods: Random Forest, FastAI tabular learner and TabNet?
Q2. Will the DNN models (FastAI, TabNet) outperform Random Forest for our tabular data?
Q3. What are the most important features for the predictive models?

Addressing the Q1 is essential for generation of disease risk model for ischemic stroke. The usage of FastAI learner and TabNet has given great results on many data sets and Q2 is focused on finding out weather this is true for the EHR data used in this Thesis. A systematic comparison of machine learning methods is used to find answers to Q1 and Q2. Random Forest is used to compare the DNN-based model for the binary classification task. The Q3 requires the analysis of feature importance which is done based on available methods for the implementation library used for ML modeling. The most important features for the models are then compared. This information in valuable for further research (not in scope of the Thesis) to analyse if the predictions are based on the known biological mechanism or something else.

The Thesis has the following structure. First, the ischemic stroke and its risk factors are described. Next, brief overview of the basis of EHR data and ML methods used is given in the Background section. In Methodology, the data set and methods used in this Thesis are explained in detail and the Results section presents the results of the analysis. The Discussion section discusses the results and compares the models' performances to the other similar models published; the limitations and future work regarding this Thesis are then discussed. Finally, the Thesis is summarized in the Conclusion.

2 Background

The Estonia's *Digilugu* is a collection of EHR which in addition to helping medical personnel can be a valuable resource for clinical research in an anonymised form. The background gives insight into what the data from *Digilugu* contains and how it can be used for ischemic stroke prediction. Firstly, the mechanisms and risk factors behind ischemic stroke are characterized. Next, the section provides an overview of the data present in EHR and problems related to EHR data usage. The ethical aspects of EHR data usage are discussed. Thirdly, the basic principles of ML methods used for model creation in the Thesis are presented and briefly explained. Finally, the focus is put on available DNN models for tabular data analysis. Methods used by the DNN in this Thesis such as batch normalization and embedding are explained. The reasons for choosing the FastAI tabular learner and TabNet are explained.

2.1 Ischemic stroke

The stroke is one of leading causes of death responsible for 11% of global total deaths in 2019 reported by the World Health Organization (WHO) [1]. The stroke can be categorized as **hemorrhagic and ischemic stroke**. The latter being responsible for the majority of strokes [16].

Ischemic stroke (I63) also known as brain ischemia or cerebral ischemia is caused by the blockage in a artery supplying the brain with blood. This blockage reduces the blood flow to the brain causing decreased oxygen levels resulting in brain cell damage or death. The blockage in the artery is caused by a blood clot or fatty build-up (plaque). The hemorrhagic stroke is caused by blood vessel rupture which results in blood leakage interfering with the brain functions. This either happens in the brain or between the brain and the skull. The hemorrhagic stroke is accountable for about 20 % of stroke cases [16]. As majority of stroke cases are ischemic the prediction of ischemic stroke is chosen as the objective of this Thesis.

The INTERSTROKE study identified 10 modifiable risk factors accounting for 91.5 % of the population-attributable risk for stroke [17]. These risk factors include: history of hypertension or a blood pressure of greater than 160/90 mmHg (carries the strongest risk), low levels of regular physical activity, a high apolipoprotein B (ApoB)-to-ApoA1 ratio, diet, high waist-to-hip ratio, psychosocial stress and depression, smoking, cardiac causes (such as atrial fibrillation and previous myocardial infarction), high alcohol consumption, diabetes mellitus. Age, gender and genetic factors are included to be risk factors of ischemic stroke [5]. The ischemic stroke has been associated with multiple metabolic biomarkers related to excitotoxicity (nerve cell damage or death caused by increased levels of neurotransmitters), oxidative stress and inflammation [18].

2.2 Electronic health records

The original purpose of the EHR was to organize and archive patients' records. Later the scope was broadened to include billing and service quality improvement as important features of the system [19]. Nowadays, medical researches also use these records for scientific studies. The primary stakeholders responsible for the development and use of EHR can be divided as follows:

1. Doctors and other medical staff. The EHR serves the purpose of accurate and available source of patient's records across multiple healthcare providers.

2. Administrative staff and health insurance funds. The EHR are used for legal documentation and auditing purpose.

3. Medical researchers. The data collected is used for research and clinical studies.

The different priorities and wishes of the stakeholders can cause challenges and limitations for the usage of EHR data in medical research further discussed in section 2.2.3. However, using the EHR data can be valuable for clinical research. The risk prediction models can be built not just making use of the cohort based model but all patients who have been in touch with the medical system [20].

2.2.1 Estonian EHR central database Digilugu

Digilugu is an Estonian nationwide EHR system owned and controlled by Estonian Ministry of Social Affairs. The central system is built by integrating data from different healthcare providers and databases in a common format so that every healthcare providers and the patients can access it online. The electronic patient record can include test results, medical images and doctors notes written in a free text format. The Health Level 7 version 3 format is used for all *Digilugu* documents [21]. Some parts of information in EHR is structured strictly (patient and clinical data, medications, diagnoses) while the rest is not (laboratory measurements, complaints). Many classifications standards are used to structure the data and improve clarity in *Digilugu* and EHR in general. Classifications standards that were used for this Thesis are described at the section 2.2.2. The extraction of structured data out of *Digilugu* requires data mining work not in the scope of this Thesis. This is further discussed in section 3.1.

The reporting of case summaries to the portal is mandatory to physicians and the collection of data began in 2009 [22]. There is a singular state provided health insurance broker in Estonia - Health Insurance Fund (*Haigekassa*) that covers 94% of the population [23]. The insurance bills (100% of billing is done digitally) are handled electronically motivating the clinicians to report in timely and well-formatted manner [22, 24].

2.2.2 Medical classification standards

ICD-10 (International Statistical Classification of Diseases and Related Health Problems 10) is developed and monitored by the WHO. The system uses alphanumeric codes that are used by medical professionals to interpret the diagnosis same way. Similar diseases are grouped together, for example I00-I99 are all diseases of the circulatory system. This grouping is based on the organ system and does not necessarily take into account the biological mechanisms behind the disease. **Ischemic stroke** studied in this Thesis is described by the code **I63**.

LOINC (Logical Observation Identifier Names and Codes) is an international standard used for health measurements, observations, and documents. The LOINC system is used in Estonia's EHR since 2016 [25]. The LOINC codes are used for metabolic measurements such as blood glucose, cholesterol and many others. The LOINC codes used in this Thesis, however, contain more than just metabolomic codes. The codes corresponding to cell counts and different biomarkers are used as well.

Anatomical Therapeutic Chemical (ATC) is a drug classification standard. It assigns a unique code for every medication in accordance with the organ or system it works on [26]. The code contains information from 5 levels: anatomical main group (e.g., alimentary tract and metabolism - A), therapeutic subgroup (e.g drugs used in diabetes - A10), pharmacological subgroup (e.g., blood glucose lowering drugs, excl. insulins - A10B), chemical subgroup (e.g., biguanides - A10BA) and chemical substance (e.g metformin - A10BA02). In the Estonian EHR data this code is available for every prescription drug that a person buys or is administered by a healthcare provider. The digital prescription system *e-Prescription* is widely used and 99% of all perception in Estonia are digital [27].

Both, the LOINC and ICD-10 codes are used as features in the predictive models of ischemic stroke developed in this Thesis. ICD-10 and ATC codes are used for selection of case and control groups further described in the section 3.1.

2.2.3 Challenges of using EHR data for research

The EHR data has many quality issues, such as high dimensionality (over 70 000 ICD-10 codes and over 70 000 different LOINC codes), heterogeneity (rarely 2 people have exactly the same medical history), sparseness (the amount of medical data available differs for subjects), random errors (data collection process is complex and prone to human mistakes) and systemic biases (some tests are only done if a doctor has a strong doubt of certain diagnosis) [28]. These problems combined make the reconstruction of the true patient state from EHR a challenging task [29].

Table 1. Common challenges of EHR data

Classification standard	Common problems identified
ICD-10	The reliability may be influenced on financial compensations, associated paperwork and coder bias.
LOINC	The values have no measure of sample quality. The methods and reagents used vary between laboratories and across time.
ATC	The time stamp describes the point of order not necessarily the time of administration.

The common problems identified for the EHR data are described in Table 1.

2.2.4 Ethical considerations of EHR usage

The restrictions for using EHR data in medical research are not only of technical nature. The ethical use of data is critical and the best practices are followed in all the experiments done in this Thesis. Two principles are highlighted as they are central for ethical data usage. Firstly, **informed consent** is required for all participants. This is achieved via broad consent form signed by the gene donors of Estonian Biobank whose data is collected and used. Secondly, the **privacy and the duty of confidentiality** is respected. This means that the EHR data used had been previously anonymized and contained only the necessary information. The confidentiality of the participant must not be compromised under any circumstances.

2.3 Machine Learning

Machine learning (ML) is a subfield of AI research focused on computer algorithms that improve automatically through experience. The following more formal definition is proposed by Tom M. Mitchell [30], "A computer program is said to learn from experience E with respect to some class of task T and performance measure P, if its performance at task in T, as measured by P, improves with experience E" In this thesis the experience E is data from EHR. The task T is the prediction if a person will be diagnosed with ischemic stroke or not. This is known as the binary classification problem. The performance measure P, will help to define the accuracy of predictions.

The Thesis focuses on ML models know as DNNs. Random Forest is used as an bench-marking model. These models are chosen as they are established methods for

binary classification problems and have the ability to handle both the numerical and categorical variables.

2.3.1 Random Forest

Random Forest is an ensemble learning method used for classification or regression tasks. Random Forest utilises multiple **decision trees** and outputs the joint prediction of singular trees [31]. A decision tree is a flowchart-like structure in which each internal node represents a test on a feature and each leaf node represents a class label. The branches of the decision tree represent conjunctions of features that lead to those class labels.

The main hyper-parameters used when tuning the fit of Random Forest models are: the number of individual trees (*n_estimators*), the maximum number of levels in a individual tree (*max_depth*), number of features considered for splitting a node (*max_features*).

2.3.2 Deep neural network

Inspiration for the DNN is drawn from the brain where connected neurons transmit signals to one another. The first mathematical model of the neuron was proposed by Warren McCullough and Walter Pitts in 1943 [32]. All logical operations could be implemented using the McCullough-Pitts neuron, with the exception of XOR that would require more than one neuron. However, the McCullough-Pitts neuron lacked a learning algorithm and the network had to be built manually. This model was developed further in 1958 by Frank Rosenblatt to be more generalized computational model called the perceptron. The major improvements of the perceptron where the use of a learning algorithm and non-integer connection weights [33]. Since the perceptron is considered the first generation of neural networks where the network is composed of only a single layer of neurons, the underlying mechanisms behind it are pictured in Figure 1. The artificial neuron has one or multiple inputs ($x_1 ... x_n$) and each input has an associated weight($w_{1j} ... w_n$). Each input is multiplied by the corresponding weight value and is then summed together by the transfer function. Activation function is used to evaluate the value of the transfer function and if the threshold is exceeded an output of 1 is returned.

A DNN is an artificial neural network (ANN) with multiple layers between the input and the output layer. Fully-connected layers is the term used for this kind of structure when neurons of the consecutive layers are connected in an all-to-all manner [34]. This can be described mathematically as calculating the activation vector a described by the Equation 1. Given an input of a row-vector i the layer multiplies this input with the matrix composed of weight values W and adds a bias vector b. Both the weight matrix W and bias vector b are learned by training the network.

$$a = i \cdot W + b \tag{1}$$

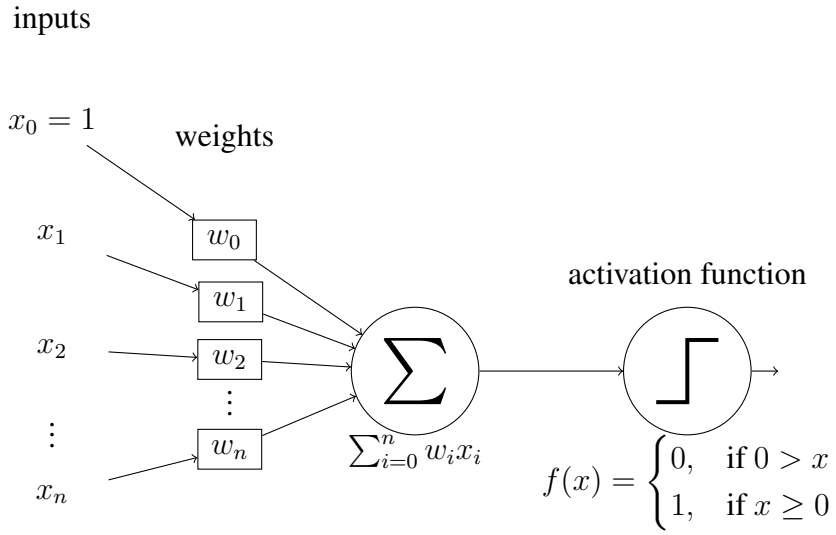


Figure 1. The perceptron. The $x_1 ... x_n$ indicate inputs and $w_{1j} ... w_n$ the corresponding weights. The Σ is the transfer function that is calculated as the inputs and corresponding weights are multiplied and then summed together. Activation function $f(x)$ evaluates the value of the transfer function and returns an output of 0 or 1.

Figure 2 depicts how layers are organized in a feed-forward neural network. This model can represent non-linear functions thanks to the use of multiple layers of thresholded neurons [34]. However, the learning algorithm of the perceptron did not allow stacking of multiple layers. The use of interconnecting layers was made possible by Rummelhart, Hinton and Williams with the efficient way of back-propagating error gradient thorough multiple layers [35]. To understand the importance of **back-propagation** the learning algorithm of the fully-connected neural networks should be explained first. The binary cross entropy loss (calculated as in the equation 2) is used as a loss function in this Thesis.

$$L = \sum_{i}^{N} [-t_i \cdot log(p_i) - (1 - t_i) \cdot log(1 - p_i)] \tag{2}$$

The true value $t_i = 1$ when the i-th data points correct answer was positive. A negative correct answer of i-th data point would result in $t_i = 0$. The prediction p_i is the probability of i-th sample belonging to the positive class according to the model. The L is calculated given all N samples, the predictions p_i and true values t_i according to Equation 2. The aim of the training process is to find such values for the weight parameters that minimal number of mistakes are made. The learning from examples begins by finding the gradient of the loss with respect to each parameter in each of the weight matrices and the bias vectors used [34, 35]. The idea of gradient decent is to change each parameter value by a

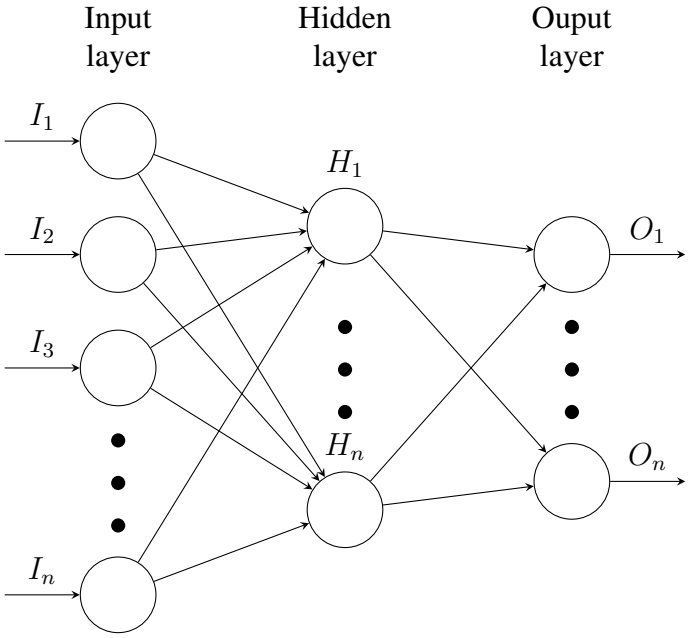


Figure 2. Layers of deep neural network. The input layer contains inputs $I_1, ..., I_n$ and output layer outputs $O1, ..., O_n$. Between input and output layer there are hidden layer(s).

small step in opposite to the respective gradient with the aim to decrease the error by a little [35]. This process is made more stable and computationally efficient by computing average gradient over a mini-batch (smaller subset of samples) and performing the learning step based on that average. This method of randomly assigning the data points into batches is called Stochastic Gradient Descent optimization algorithm [36]. The size of each update is calculated as the averaged gradient times a learning rate. An extension of stochastic gradient descent algorithm called Adam is used in this Thesis [37]. The name Adam is derived from "adaptive moment estimation". Adam was chosen as it is an optimization algorithm that can handle sparse gradients on noisy problems, and it is relatively easy to configure (default parameters do well on most problems). Adam has been shown to efficiently solve practical deep learning problems [37]. The **back-propagation** method is used to compute all the gradients mentioned previously. The computation of gradients begins by finding the derivatives in the last layer and then moving layer by layer back toward the inputs while calculating all gradients needed for the learning. The gradient descent is guaranteed to only converge toward some local minimum, and not necessarily the global minimum error (indicating that there is no guarantee of finding the p_i that would result with the lowest loss via this process). Despite this, the back-propagation has yielded excellent results in many real-world applications [30].

2.4 DNN for tabular data

Tabular data (also sometimes referred as relational data[1] or structured data) is something that is held in a table (such as SQL database, Excel sheet or a Pandas data frame). The data is composed of rows and columns. The aim of the model would be to predict the values of one column based on the other columns for every row in the table. Popular approaches for classification of tabular data are ensemble methods like Random Forest and XGBoost[38].

EHR data may not be inherently in a tabular format (*Digilugu* uses XML data structure) but is often converted to tables for research purposes as a result of data mining and processing. This is the case for EHR data used for this Thesis as well. For a more detailed snapshot of biological processes many layers of Omics data are added together [39]. This approach would be possible for many other Omics data types (such as metabolomics, microbiomics, transcriptomics) collected by the Estonian Biobank [40]. For the scope of this Thesis only the data available in EHR are used (disease diagnosis, some metabolomic and cellular measures).

The usage of DNN models for EHR and Omics tabular data is motivated by the following:

- Classification performance improvement (particularly on large data sets) [10, 11, 41].
- Potential to efficiently encode multiple types of data (e.g., medical images) along with tabular data [8].
- Alleviate the need for feature engineering [8].
- Potential for data-efficient domain adaptation [34].

The two DNNs implemented in this Thesis are the **FastAI tabular** and **TabNet**. More details about these methods is given in sections 2.4.3 and 2.4.4. Both of these methods use **embedding** and **batch normalization layers** that are further explained below.

2.4.1 Batch normalization

Batch normalization is a widely adopted technique of adaptive reparameterizations which enables faster and more stable training process of DNNs [42]. The exact reasons for batch normalization effectiveness are not fully understood [43]. The Batch normalization transform algorithm proposed in [42] is shown in Algorithm 1 where the input layer is normalized by re-centring and re-scaling.

[1]The author of this Thesis is aware of the definition of relational model by Edgar F. Codd. As the practical use of these data structures does not differ in the context of the Thesis relational and tabular are used interchangeably.

Algorithm 1: Batch Normalizing Transform

Input: Values of x over a mini batch: $B = \{x_1, ...x_m\}$;
Parameters to be learned: γ, β
Result: $\{y_i = BN_{\gamma,\beta}(x_i)\ \}$

1 $\mu_B \longleftarrow \frac{1}{m}\sum_{i=1}^{m} x_i$
2 $\sigma_B^2 \longleftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_B)^2$
3 $\hat{x}_i \longleftarrow \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$
4 $y_i \longleftarrow \gamma\hat{x}_i + \beta \equiv BN_{\gamma,\beta}(x_i)$

The first step of algorithm 1 is finding the mean for the mini-batch. Next, on the line 2 the mini-batch variance is calculated. After that the values are normalized using the mean and variance found on previous steps. Finally, on line 4 scale and shift are applied.

2.4.2 Embedding

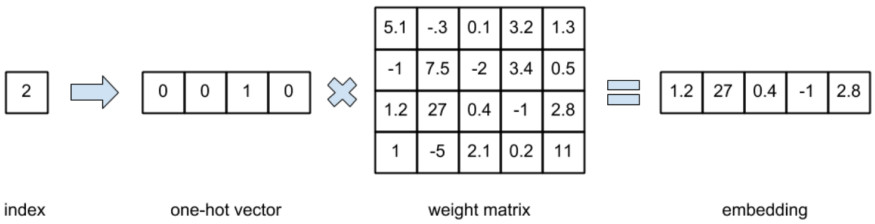


Figure 3. Example of embedding process. A index is transformed to one-hot encoded vector which is multiplied by the weight matrix. As a result embedding is created. Image orginates from Tambet Matiisen blog post [44].

Embedding improves ML on large inputs like sparse vectors representing words. This is achieved via the relatively low-dimensional embedding space (which is continuous) into which high-dimensional vectors (one-hot encoded) are translated. The aim of this process is to capture some semantics of the input and place semantically similar inputs closer together in the embedding space. [45] Figure 3 demonstrates how a value (index) is converted to one-hot vector that is multiplied by an weight matrix resulting in emebdding. The embedding layer is trained with the network or an already pretrained embedding is used. For this Thesis the embedding is used for both DNNs in the hope that it will increase the model's performance via capturing the semantics of categorical variables such as ICD-10 codes.

2.4.3 FastAI tabular

FastAI is a Python library built on PyTorch by Jeremy Howard [9]. The version 1 was released in 2018. The motivation behind FastAI is to enable building state-of-the-art deep learning models simply and quickly. This enables domain experts like medical researches to start using the pre-existing neural network architectures without the need for detailed implementation [15]. Nevertheless FastAI is not widely known or used among metabolomics researchers. An article released in 2020 listing publicly available resources and modules for deep learning in metabolomics research did not even mention the FastAI [3]. Regardless, FastAI libary is still a powerful tool utilising many state-of-the art practices for tabular data. FastAI tabular uses **embedding layer** (inspired by word embedding) for categorical features. This means that categorical variables are mapped into Euclidean space in a functional approximation problem. This mapping is learned by the neural network during the supervised training process. Intrinsic properties of the categorical variables may be revealed by the mapping. Other benefits of entity embedding are a reduction of memory usage and an increased speed of the neural network compared to one-hot encoding. It has been demonstrated that for sparse data the embedding helps the neural network to generalize better. As EHR data have high cardinality features this helps to avoid the overfitting. [45]

FastAI implements the **cyclical learning rates** for training. The learning rate cyclically varies between reasonable boundary values which improves the classification accuracy [46].

In summary, FastAI tabular was chosen as it is relatively easy to use and is designed for domain experts while having a built-in functionality for tabular data.

2.4.4 TabNet

TabNet is a high-performance interpretable canonical deep tabular learning architecture developed by Sercan Ö. Arik and Tomas Pfister at Google [8]. **Sequential attention** is used by TabNet to choose features at each decision step to reason from. This results in model being interpretable by differentiating the important features from others. The authors of TabNet have demonstrated that TabNet outperforms other methods such as Decision Trees, Random Forest and many others on a wide range of publicly available data sets[8]. The TabNet adaption for FastAI (fast-tabnet 0.2.0 [47]) was used in this Thesis.

TabNet was chosen as it requires minimal data pre-processing while offering improved interpretability compared to classical DNN used by FastAI tabular learner.

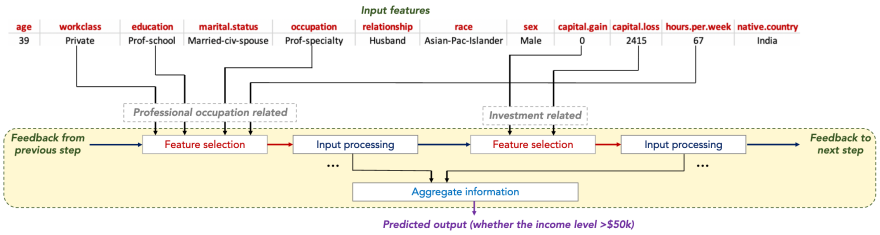


Figure 4. Figure taken from S. Ö. Arik and T. Pfister TabNet article [8]. Example of sparse feature selection of TabNet on Adult Census Income prediction[48]. Multiple decision blocks are used that focus on processing a subset of input features for reasoning. Figure depicts 2 feature blocks that process features related to professional occupation and investments. These features are then used to predict the income level [8].

3 Methodology

The code for data pre-processing was written in R (version 4.0.3) while all implementations of the models (Random Forest, FastAI tabular and TabNet) were written in Python (version 3.7.10). The FastAI version 2.3.1 was used to build the DNNs. Data analysis and plotting was done either in R using ggplot2 [49] or in Python. The code written for this Thesis is available in the following GitHub repository: `https://github.com/skurvits/Ischemic-stroke-models-using-DNN`.

The next subsections will describe the data and the analytical methods that are used in this Thesis.

3.1 Data preprocessing

The EHR data used in this Thesis was processed and imputed from epicrises documents by Software Technology and Applications Competence Center (STACC). The details of the process are described in the technical report [50]. The LOINC codes with values were mined from EHR data fields as they are not always held in a tabular format. The data set was produced using the data from February 2004 to March 2020 and contained 2.7 million entries.

After gaining access to the data mined by STACC all measurements after the first stroke were removed. The data gained was cleaned further and some LOINC values imported from other columns (not all LOINC codes were actually in the same column, but were mixed between few other columns). In some cases a LOINC measure was available with more than one measurement unit in the data. No unit conversion was done and each unit was considered as a different features (in the Results section these variables are referred to using additional number such as B.Hct_1 and B.Hct_2 for measures of hematocrit using 2 different measurement units). Rare analyses (columns with less than

45 occurrences) were removed same is true for the individuals with less than 10 different LOINC measurements available. The measurements that were taken more than 1000 days before the stroke were also removed. As the result of the cleaning process, out of the 783 LOINC codes available from the original data only 145 were used in the models.

The age during the time of measurement is calculated for each person based on the birth year. As the exact date of birth is unknown due to the ethical considerations (explained in the section 2.2.4) the age value may not be exact but cannot differ more than 1 year of the real value during the measurement.

No manual feature selection was used for any of the 3 models implemented, this is to validate the claim of TabNet's feature selection ability. Additional features were created for counting the number of LOINC codes available per person. Next, all LOINC measurements made on the same date were aggregated by the person.

3.1.1 Missing data and imputations

The EHR has a value for LOINC code only if the analysis has been done for that person. As there are many different LOINC codes, for every person some of them are missing for every person in the data set. The last observation carried forward approach is used so that the last available LOINC and ICD-10 values are used, if the last measurement date had no such value available then the value is imputed with the previous value when available. If after the last observation carried forward approach still a value is missing it is imputed with zero. This imputation mimics better the "non-measured" characteristic of the feature. Imputation with the mean value was also tested but yielded worse results.

3.1.2 Categorical data

For Random Forest the one-hot encoding was used for categorical data. The one-hot encoding encodes each of the n features in a vector of n elements, with all elements set to 0 but one, at a different index for each feature. Both the FastAI tabular and TabNet DNN use embedding for the categorical features, thus a prior one-hot encoding prior is not necessary.

3.1.3 Selection of cases and controls

From the data case and control groups were formed using the PopSel software [51]. For every case suitable controls (no ischemic stroke diagnosis) are found with the corresponding sex and the most similar birth year. The last known measurement date must not be more than 1000 days before the diagnosis of stroke for the cases. This resulted in 749 cases and 2033 controls. The similarities of case and control groups are described in Figure 5. The Venn diagram in the figure shows that 62 out of the total of 473 ICD-10 codes in the data were only present for the cases group while 229 only

occurred in the control group. The 182 ICD-10 values have occurred in both groups. In a later section, a sensitivity analysis is performed (Section 4.4) to confirm that the machine learning models do not learn to rely too heavily on this information.

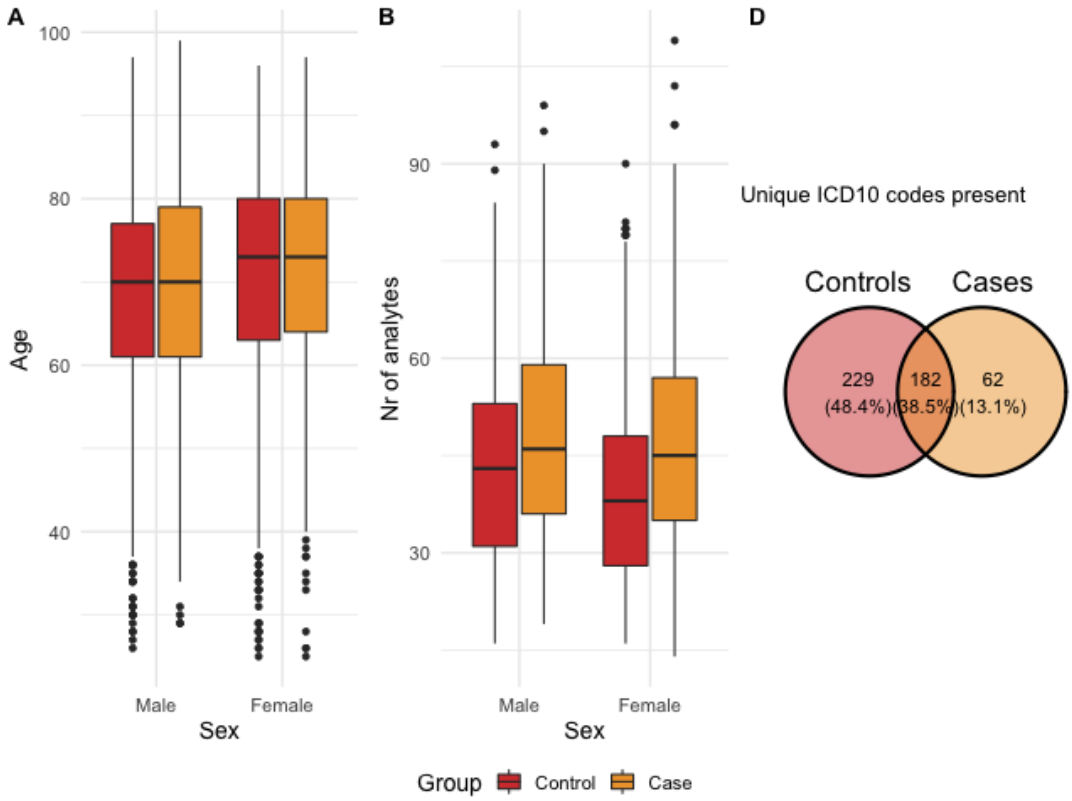


Figure 5. Case and control groups by birth year, sex and measurements data. Boxplot A compares the case and control group by sex and age. Boxplot B compares the case and control groups by sex and number of analytes available. The Venn diagram D shows how many ICD-10 codes are shared and how many are unique to the case and control groups.

In addition, the control group was selected such that no individuals with the following ICD-10 codes or drugs codes were included. ICD-10 codes selected against are the following cerebrovascular diseases: I60, I61, I62, I64. This selection is used to avoid individuals with hemorrhagic stroke or similar symptoms and was selected after consultation with medical doctor. ATC codes selected against: B01 (antithrombotic agents), M01A (anti-inflammatory and anti-rheumatic products, non-steroids) , M01BA03 (acetylsalicylic acid and corticosteroids), N02BA (salicylic acid and derivatives). These medications directly influence the biological mechanism of stroke and so may directly influence other biological signals indicating stroke.

This type of case-control matching introduces the underlying problem of imbalanced classes for binary classification task. There is a stronger incentive for the model to predict a more popular class when in doubt. This issue is addressed by selecting a non-biased evaluation metric as described in the section 3.2.2 and using methods for imbalanced class training as described in section 3.3.

3.2 Evaluation framework

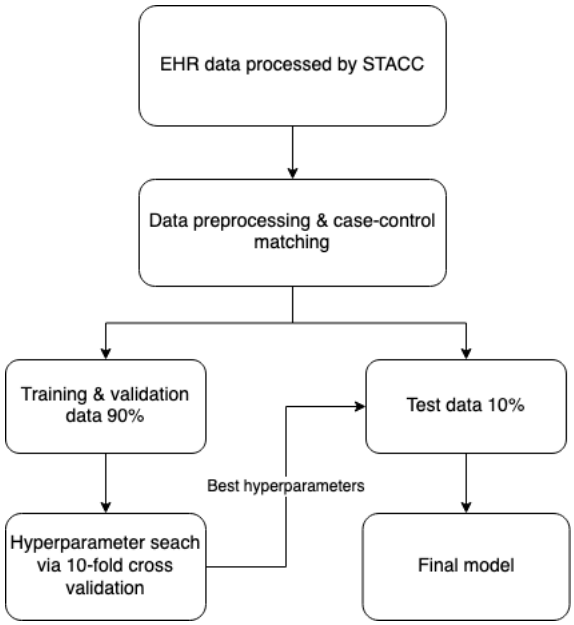


Figure 6. Workflow of model creation and testing

Section 2.3 described how ML algorithms use training data as examples to learn the rules for the classification problem. One problem of finding such rules is the **over-fitting** of the model. This happens when the model performs well on the training data but fails to generalize and thus the performance drops when used on new data. These models have less practical value as usually the purpose of a model is to use it on new and unencountered data. To evaluate the over-fitting the available data are split into training and test data sets. Use of test samples is the most fair estimate of a model's ability to generalize[52]. The test data set is used only for the final evaluation of the model and is not used in the training or hyper-parameter selection process. As selecting the right hyper-parameters for the data is crucial for good performance of the model this is solved with using k-fold cross validation described next.

3.2.1 Stratified k-fold cross validation

Cross validation is a method from Monte Carlo family used to sort data into training and validation sets. The stratification means that each set created contains the same proportion of samples of each class as the whole data set. A 10-fold cross validation is illustrated on the Figure 7.

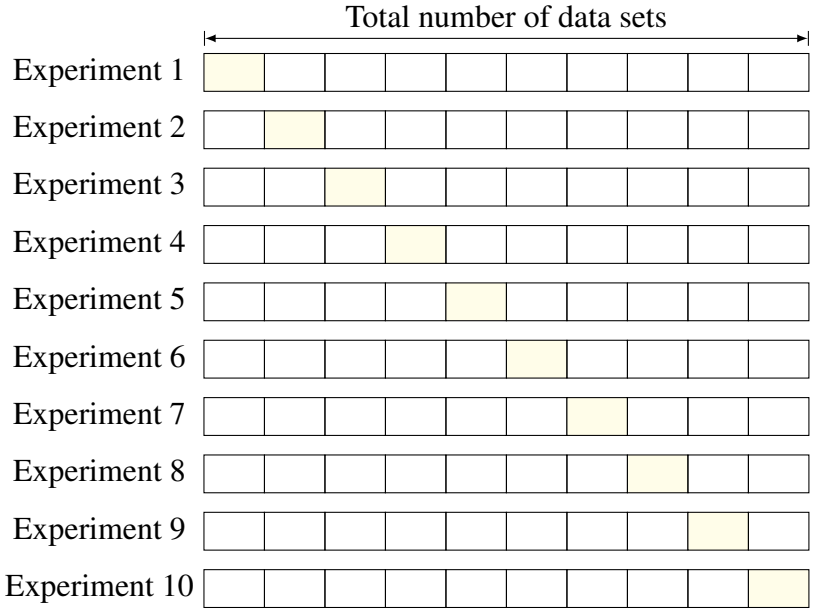
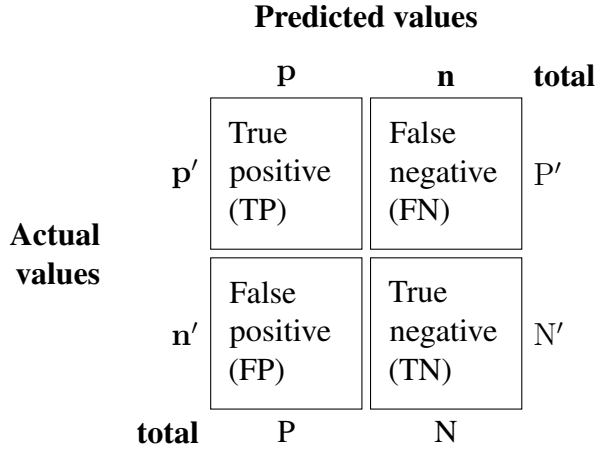


Figure 7. 10-fold Cross validation. Each yellow box is a 10% validation set that is used as a test set for the corresponding experiment.

Although, the k-fold cross validation gives insight into the performance of the models, the final comparison of 3 models is made using test data not previously used for the hyper-parameter tuning. This approach helps to better understand how well the model performs on novel data and gives assurance that the hyper-parameters were not tuned in accordance with the test data selected. To compare the results of the 3 models the stratified 10-fold cross validation implemented uses exactly the same data for every fold across all three models.

3.2.2 Evaluation metric

The **confusion matrix** is a table used for classification model evaluation on test data where the true values are known. The figure 8 illustrates how all predicted values are divided into 4 sets. Using the confusion matrix, the following performance metrics are commonly calculated: accuracy, precision, F1-score. All the performance metrics



$$False\ positive\ rate = \frac{FP}{N}$$

$$True\ positive\ rate\ (recall) = \frac{TP}{P}$$

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

$$Accuracy = \frac{TP+TN}{P+N}$$

$$F1 - score = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

Figure 8. Confusion matrix and common performance metrics

mentioned before should be calculated for both classes (predicted to have ischemic stroke and predicted to not have the stroke) due to the class imbalance present in data.

The **receiver operating characteristic curve (ROC)** is commonly used for medical decision making and ML model evaluations [53]. The ROC graph is a two-dimensional graph in which true positive rate is plotted on the Y-axis and false positive rate is plotted on the X-axis. The ROC graph illustrates the relative trade-offs between benefits (TP rate) and costs (FP rate). The diagonal line $y = x$ on ROC graph represents randomly guessing a class. For a classifier to be better than random it must appear in the upper left triangle of the ROC graph. In this case-control study the class distribution between cases and controls is skewed (number of controls is higher to increase power of the study). The ROC curves are however insensitive to changes in class distribution. The change in proportion of the positive to negative instances in test set will not change the ROC curve. This can be illustrated using the confusion matrix (Figure 8). The class distribution of positive to negative instances is the relationship of the left column to the right column. The performance metric using values from both columns are inherently sensitive to class skew. Accuracy, precision and F1-score use both of these columns from the confusion matrix. The change of class distribution results in a change in these values. ROC, however, is calculated using only true positive rate and false positive rate. Both of these metrics are strict columnar ratios and therefore do not depend on class distributions. Any performance metric that uses values from both columns will be inherently sensitive to class skew.

3.3 Model implementations

For all models stratified 10-fold cross validation was used to select the optimal hyper-parameters for the task. The evaluation metric used for each model was the average area under the curve (AUC) value calculated across all 10 folds. The results of the 10-fold cross validation are shown in detail in section 4.1.

3.3.1 Random Forest

The Python sklearn library's Random Forest algorithm was implemented [54]. The best hyper-parameters were found using grid search. The following ranges of hyper-parameters were used:

- $max\ features \in \{sqrt\ , 25, 50\}$
- $max\ depth \in \{10, 20, 30, 40, 50, 60\}$
- $min\ samples\ leaf \in \{2, 5, 8, 11\}$
- $number\ of\ estimators \in \{600, 1000, 1400, 1800\}$

The best parameters were selected based on AUC value of 10-fold cross validation described in section 4.1.1.

3.3.2 FastAI tabular

The FastAI tabular model was implemented using Python FastAI library [9]. Stratified 10-fold cross validation was used for hyper-parameter search. The best hyper-parameters were found using grid search. The following ranges of hyper-parameters were used:

- $learning\ rate \in \{0.005, 0.01, 0.02, 0.025, 0.05\}$
- $number\ of\ layers \in \{2, 3\}$
- $number\ of\ nodes \in \{100, 300, 500, 1000\}$

The best parameters were selected based on AUC value of 10-fold cross validation described in section 4.1.2.

3.3.3 TabNet

Grid search was used in combination with 10-fold cross validation to find the most suitable hyper-parameters. The tested hyper-parameters were chosen based on the recommendations in the TabNet article [8].

- $learning\ rate \in \{0.005, 0.01, 0.02, 0.025\}$
- $N_{steps} \in \{3, 4, 5, 6, 7, 8, 9, 10\}$
- $N_d = N_a \in \{8, 16, 32, 64, 128\}$
- $\gamma \in \{1.0, 1.2, 1.5, 2.0\}$
- $B \in \{256, 512, 1924, 2048\}$

The best parameters were selected based on AUC value of 10-fold cross validation described in section 4.1.3.

3.4 Interpreting the model by feature importance

A mathematical definition of interpretability does not exist. Nonetheless, the model's interpretability can be described as the degree to which a human can understand the cause of a decision [55]. Another definition proposed is that interpretability is the degree to which a human can consistently predict the model's result [56]. The inerpretability has critical value for clinical models as it builds trust and transparency.

This Thesis focuses on finding the most important features for every model tested to answer the research question Q3. The tools used for finding the most important features are either available by the implementation library (sklearn, TabNet) and/or commonly used for such model. The top 20 most important features were found for every model.

For Random Forest the feature importance was calculated using sklearn. The sklearn package uses Gini importance (mean decrease impurity) that is defined as the total decrease in node impurity averaged over all trees of the ensemble. As one feature can occur in multiple nodes the feature importance is calculated over the sum of all occurrences and the sum is divided by the total impurity reduction of all nodes [57].

The FastAI tabular model has no built-in feature importance functionality. A model-agnostic approach of finding the feature importance (model reliance) is used [58] instead. The model reliance is found by a permutation importance method that was implemented by the author of the Thesis. The permutation feature importance is defined as the decrease in a model score when the values of a single feature are permuted [59]. The relationship between the feature and the target is broken by this procedure, and the drop in the model score is indicative of how much the model depends on the feature. Firstly, the original error is estimated. Then for each feature a new feature matrix is built by permuting the

feature. The new permuted feature matrix is used to estimate the error. The permutation feature importance is found as the error of the permutation divided by the original error.

TabNet sparse feature selection enables the model to be interpretable via finding the most important features [8]. This built in functionality was used to find the top 20 important features.

4 Results

This Chapter describes the results of the analysis conducted in this Thesis. In Section 4.1 the results of 10-fold cross-validation experiments for finding the best hyper-parameters are described. These hyper-parameters were used for training the final model which is evaluated on a test set. The results of the final models are detailed in Section 4.2. In addition, simple ensembles models using the 3 trained models were evaluated on test set. Next, the most important features for every model were found and are reported in the Section 4.3. The ICD-10 was the most important feature for both DNN models used. To evaluate the impact of disparity of ICD-10 codes between cases and controls the sensitivity analysis on that feature was performed. Finally, the classification errors for all 3 models were analyzed further.

4.1 10-fold cross validation

All 10-folds used were identical across all models meaning that that the fold 1 contained the same individuals in training and test data for the random forest, FastAI and TabNet models.

4.1.1 Best hyper-parameters for Random Forest

The following hyper-parameters resulted with the highest average AUC value of 0.92 in the 10-fold cross validation.

$$
\begin{aligned}
n\_estimators &= 1800 \\
max\_features &= sqrt \\
max\_depth &= 60 \\
min\_samples\_leaf &= 5
\end{aligned}
$$

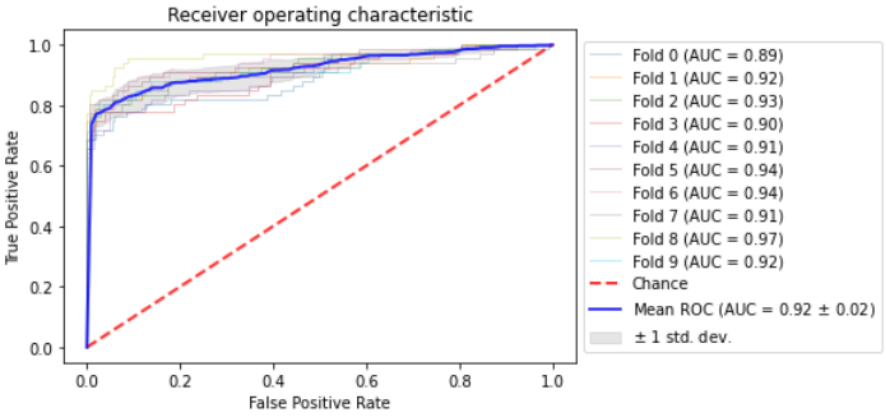


Figure 9. Random Forest ROC plots for 10-fold cross validation

The results of 10-fold cross validation on training data using Random Forest are shown in Figure 9. All 10-folds were similar with the lowest AUC value of 0.89 and highest of 0.97 resulting in standard deviation of 0.02 for the AUC values.

4.1.2 Best hyper-parameters for FastAI

The following hyper-parameters resulted with the highest AUC value in the 10-fold cross validation.

$$learning\_rate = 0.05$$
$$number\ of\ layers = 2$$
$$number\ of\ nodes\ layer\ 1 = 500$$
$$number\ of\ nodes\ layer\ 2 = 1000$$

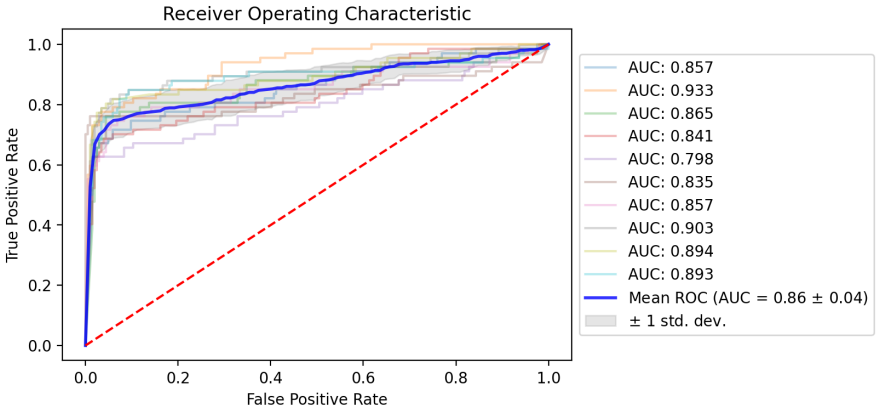


Figure 10. AUC measures of 10-fold cross validation of FastAI tabular model

The results of 10-fold cross validation on training data using FastAI tabular learner are shown in Figure 10. The average AUC value was 0.86 with the standard deviations of 0.04. The cross validation results indicate more variance between models tested on different fold. The lowest fold achieved the AUC value of only 0.798 while the highest value achieved was 0.933. The FastAI learner model was less stable and had a lower mean AUC score compared to Random Forest. The model was more sensitive to the data set used for training and testing indicated by higher standard deviation of AUC scores and lower overall score.

4.1.3 Best hyper-parameters for TabNet

The following hyper-parameters resulted with the highest mean AUC value in the 10-fold cross validation.

$$
\begin{gathered}
learning\_rate = 0.02 \\
N\_steps = 3 \\
N\_d = N\_a = 128 \\
\gamma = 1.0 \\
B = 2058
\end{gathered}
$$

The results of 10-fold cross validation on training data using TabNet are shown in Figure 11. The average AUC value was 0.93 with the standard deviations of 0.01. The highest AUC value of the 10-folds was 0.954 and the lowest 0.904.

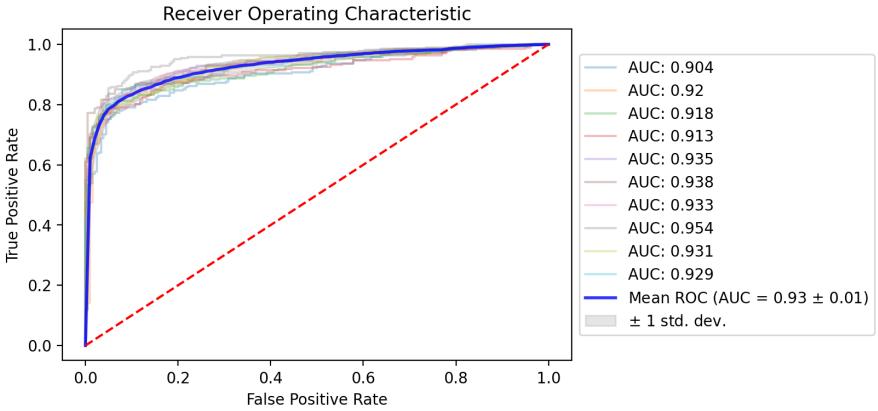


Figure 11. TabNet ROC plots for 10-fold cross validation

4.2 Model performance on test data

In this section we are interested how our models perform on novel data not used for the training and parameter tuning processes. The results of this section answer the research questions Q1 about weather ischemic stroke could be predicted using the pre-existing EHR data. This section is used as the final comparison of the 3 models used in the Thesis to answer the question Q2 weather the usage of DNN offered a performance increase compared to the Random Forest baseline model.

4.2.1 Random Forest performance

The hyper-parameters described in Section 4.1.1 were now used to train one model using the whole training set (90% of the available data) and the model was then evaluated on the training data (10% of data unused in the training process before). The results are shown in Figure 12.

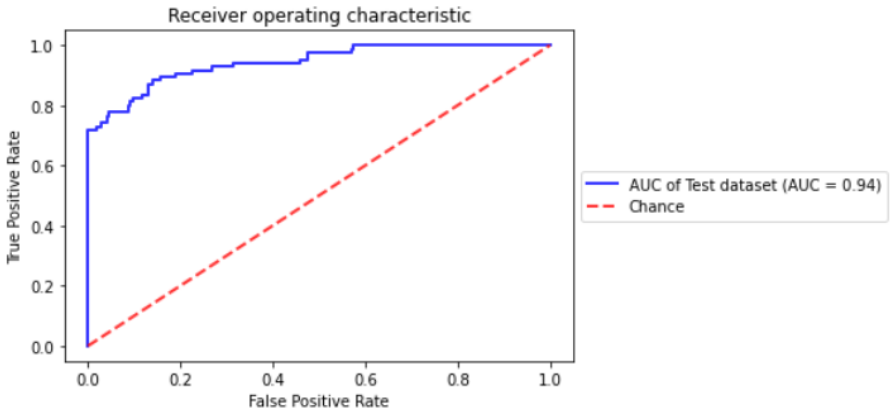


Figure 12. Random Forest ROC plot of the final model

The AUC value for test data was 0.94. A lower score was achieved as the average of 10-fold cross validation. This indicates that this model extrapolates well for new data.

4.2.2 FastAI performance

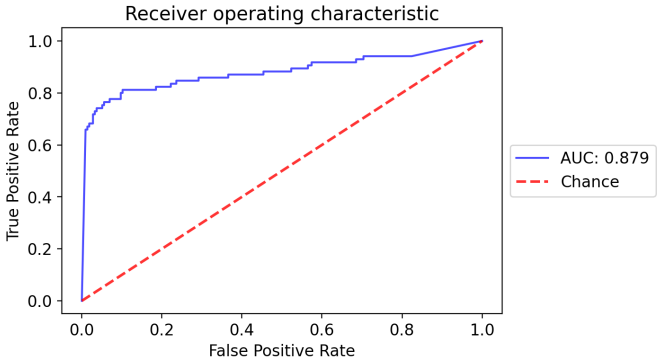


Figure 13. ROC of FastAI tabular final model

The ROC of the final model for FastAI tabular learner is shown on figure 13. The AUC value of 0.89 is in the expected range given the 10-fold cross validation results. Detailed architecture of the model implemented is given in the Appendix II.

4.2.3 TabNet performance

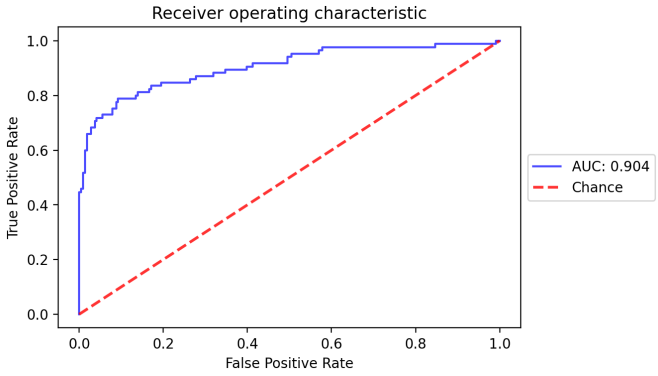


Figure 14. ROC of TabNet final model

The ROC of the final model for TabNet (Figure 14). The AUC value of 0.90 was in the lower end of the 10-fold cross validation results, as the lowest of all 10 experiments had the AUC of 0.90.

4.2.4 Comparison of models

Table 2. Results of 10-fold cross validation and test set

Evaluation framework	**Random Forest**	**FastAI tabular**	**TabNet**
10-fold cross validation	0.92 ± 0.02	0.86 ± 0.04	0.93 ± 0.01
Test set	0.94	0.88	0.90

As described by Table 2 Random Forest yielded the best results on test data followed closely by the TabNet. FastAI tabular did not perform as well, but was still able to generalize to test data. Detailed architecture of the models implemented is given in the Appendix section II.

4.2.5 Ensemble models

The ensemble models are created such that the average prediction values for all models in the ensemble are found. A few ensemble models are also calculated with using weights on the best singular models. For example in Table 3 3x Random Forest + FastAI + 2x TabNet means that the predictions of Random Forest model are multiplied by 3, the prediction value of FastAI is added with the prediction value of TabNet multiplied by 2.

The resulting value is then divided by 6 and based on this new prediction value a AUC is calculated.

Table 3. Calculated AUC values for different ensembles of 3 models for the test data.

Model	**AUC**
Random Forest	0.942
FastAI	0.879
TabNet	0.904
Random Forest + FastAI	0.943
Random Forest + TabNet	0.935
FastAI + TabNet	0.927
Random Forest + FastAI + TabNet	0.943
3 x Random Forest + FastAI + 2 x TabNet	0.945
3 x Random Forest + FastAI + TabNet	0.947
2 x Random Forest + FastAI	0.947
3 x Random Forest + FastAI	0.948
4 x Random Forest + FastAI	0.949
5 x Random Forest + FastAI	0.948

The best ensemble model composed of Random Forest and FastAI models achieved 0.95 AUC value compared to the highest single model of Random Forest AUC of 0.94. All simple ensembles created and tested (except for the Random Forest and TabNet pair) achieved a higher AUC than the single models used in the ensemble. This indicates that even if the DNN models alone may not achieve better results than Random Forest, the ensemble model using the DNN models and Random Forest may improves the results.

4.3 Feature importance

In this section, the results for feature importance analysis for research question Q3 are presented. Firstly, 20 most important features for each model are given. Next, the overlap of important features is found.

4.3.1 Most important features for Random Forest

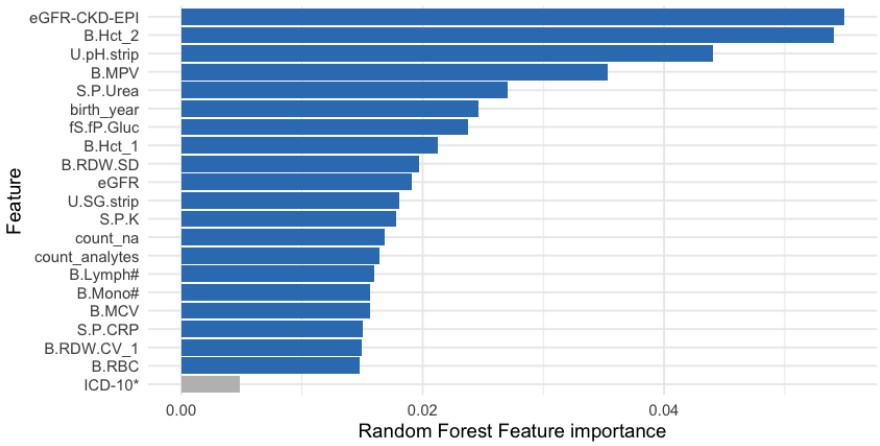


Figure 15. Random Forest model 20 most important features. The ICD-10* is the sum of all one-hot encoded ICD-10 code feature importance and is not actually one of the 20 most important features.

As for feature importance of Random Forest model the top 20 most important features are shown in Figure 15. All the features have rather small importance varying between 0.015 and 0.055. Feature with the highest importance is the LOINC code: 62238-1 which corresponds to **eGFR-CKD-EPI**. The eGFR-CKD-EPI is the estimated glomerular filtration rate using Chronic Kidney Disease Epidemiology Collaboration (CKD-EPI) formula. The eGFR-CKD-EPI value is calculated using age, sex, race[2] and serum creatinine levels. Next features following the eGFR-CDK-EPI are the LOINC code 4544-3 (B.Hct_2) measuring **hematocrit**, the proportion(percentage) of red blood cells in the blood and (LOINC code: 50560-2, U.pH.strip) the **pH of urine**. Out of non LOINC values the **year of birth** and the number of LOINC codes available are were the top 20 most important features for the model.

[2]Estonia's EHR does not use race in calculation formula.

4.3.2 Most important features for FastAI

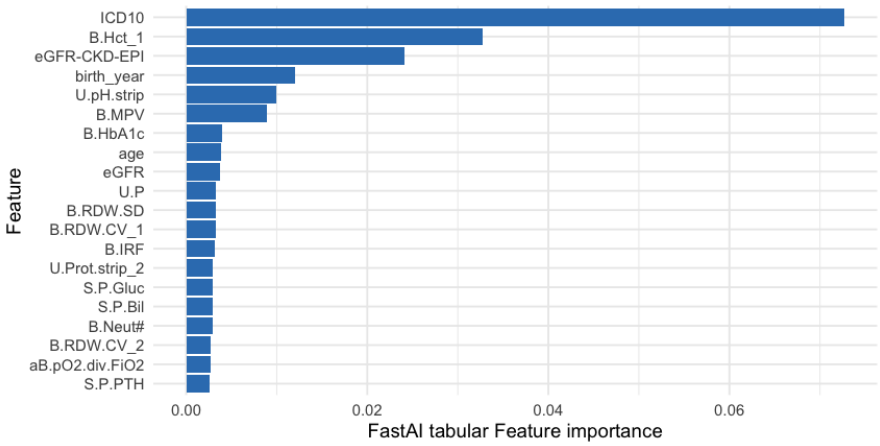


Figure 16. FastAI tabular model 20 most important features

As seen in Figure 16 the most important feature for the FastAI models is the **ICD-10 code** having over 10 times the importance of the least important feature pictured. Other top features are the **hematocrit** (LOINC code: 4544-3, B.Hct_2), **eGRF-CKD-EPI**, **year of birth** and the **pH of urine**. The FastAI feature importance plot differs from the Random Forest and TabNet as the feature importance drops significantly after the first feature while Random Forest and TabNet top features decrease in importance more gradually.

4.3.3 Most important features for TabNet

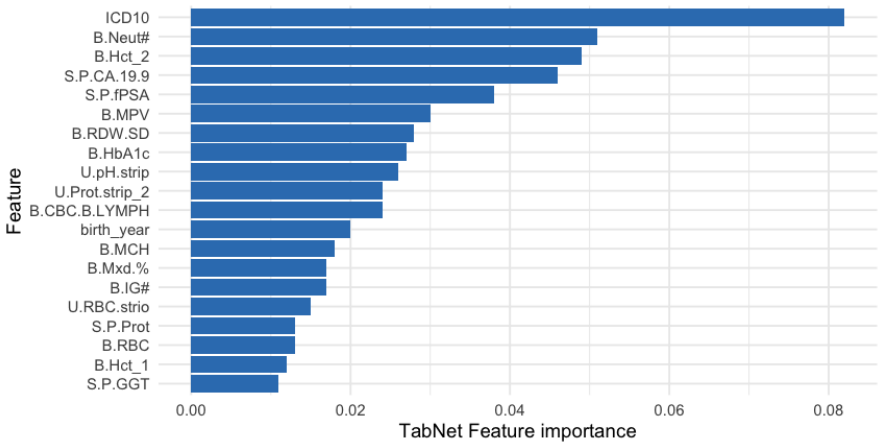


Figure 17. TabNet model 20 most important features

The **ICD-10 code** is the most important feature for the TabNet model as pictured in Figure 17. It is followed by the **number of neutrophils is blood** (LOINC code: 751-8 , B-Neut#), the **hematocrit** (LOINC code: 4544-3, B.Hct_2) and the **cancer Ag 19-9** (LOINC code: 24108-3, S.P.CA-19.9) which is a tumor marker.

4.3.4 Comparison of models

The overlap between the most important features found is displayed on the Venn diagram (Figure 18). For every model about half or more of the top 20 features found are unique to that model. The 5 features that are important to all 3 models are the **year of birth**, the **hematocrit** (B-Hct, LOINC code: 4544-3), **pH of urine** (U-pH-strip, LOINC code: 50560-2), **platelet mean volume** (B-MPV, LOINC code: 32623-1) and **erythrocyte distribution width** (B-RDW-SD, LOINC code: 21000-5). The hematocrit, platelet mean volume and erythrocyte distribution are measured as a subsets of complete blood count (hemogram) panel (LOINC code 58410-2), which is one of the most common analysis performed.

The 3 features shared only by FastAI and Random Forest are **glomerular filtration rate** (LOINC code:33914-3, eGFR), eGFR-CKD-EPI (LOINC code 62238-1), **erythrocyte distribution width** (LOINC code:788-0, B-RDW-CV). The erythrocyte distribution width found important here is not the same as the one mentioned previously to be important for all 3 models as it has a different LOINC code and measurement principle. The 2 features shared by only TabNet and Random Forest are the **count of erythrocytes in**

blood (LOINC code: 789-8 B-RBC) and **hematocrit** (B-Hct_1) measured in different units than the one mentioned previously.

FastAI and TabNet share 4 features not named before. Firstly, **ICD-10** the diagnosis given on that day which was the most important feature for both models. As this feature was one-hot encoded in the Random Forest model it is not properly accounted for in the Random Forest feature importance calculations. To get an estimate for comparison all ICD-10 one-hot encoded feature importances were summed together in the Random Forest feature importance analysis. The sum of all binary ICD-10 codes in Random forest achieved feature importance value of only 0.0048. Secondly, the following LOINC code of 32623-1 (B.-MPV) used for the **platelet mean volume**. Thirdly, the count of neutrophils in blood (LOINC code: 751-8, B-Neut). Finally, the **hemoglobin A1c/Hemoglobin total in blood** (LOINC code: 4548-4 , B.HbA1c), which measures the amount of hemoglobin with attached glucose and is a common test in diabetes diagnostics.

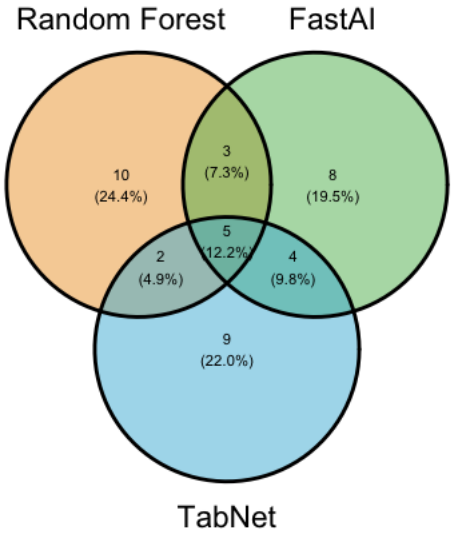


Figure 18. Venn diagram of the TOP 20 most important features for each model. The Random Forest is in upper left circle, FastAI tabular learner upper right circle and TabNet bottom circle.

4.4 Sensitivity analysis of ICD-10

The sensitivity analysis of the ICD-10 feature is done for all models implemented. The overall idea of the analysis performed is simple. If the model is very sensitive to certain

diagnosis, setting the diagnosis true for control group samples, should lead to them being misclassified as positive cases. This process is implemented as Algorithm 2.

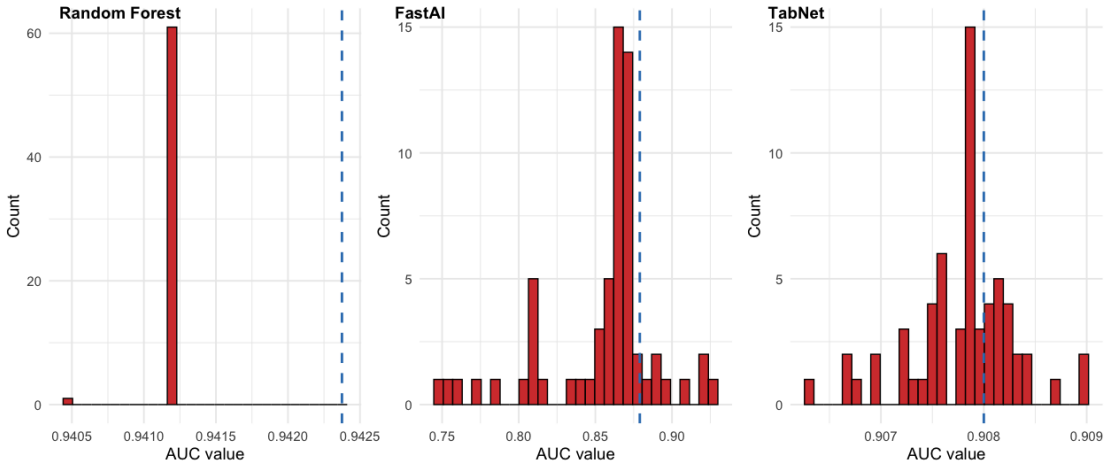


Figure 19. Distribution of the algorithm 2 results using 3 model: Random Forest, FastAI tabular learner and TabNet. The blue dashed line indicates the original test data AUC value for each model.

Figure 19 displays sensitivity analysis done for the Random Forest model. Overall all AUC values are a litter lower but the decrease in value is so small that it holds no practical importance. The imputation with the I69 resulted the lowest AUC value of 0.9405. These results are supported by the low feature importance of the ICD-10 codes as the sum of all one-hot encoded features was only 0.0048 not being even listed among the top 20 most important features for the model.

Figure 5 shows that 62 out of the total of 473 ICD-10 codes in the data used are only present for the cases group. The FastAI and TabNet models feature importance analysis ranked the ICD-10 as the most important feature (Figures 16 and 17) for both models. The aim of this sensitivity analysis is to detect to what extent the models rely on ICD-10 codes only available for cases groups as classification criteria. The experiment is explained by Algorithm 2. The general idea is to set the ICD-10 diagnosis of control group to a value only present in the case group and see if this leads to misclassification to a case. This is implemented for all ICD-10 codes present in only cases group and the misclassification is measured by the AUC value.

The FastAI sensitivity analysis results are displayed in Figure 19. The AUC value calculated on the original test data of 0.88 is higher than the average of AUC values. Overall, the AUC values seem to decrease (some are even as low as 0.75), indicating that the ICD-10 values are important for making decisions. The ICD-10 codes having the largest impact on the models predictive power (lowering the AUC below 0.8) are

Algorithm 2: ICD-10 sensitivity analysis

Input: S_{ICD-10}, the set of ICD-10 codes present only in the cases group.
Result: The AUC values for modified data sets.

1 **foreach** s *in* S_{ICD-10} **do**
2 Change all original values for the control group ICD-10 variable to s
3 Use the model on the modified data and calculate new ROC AUC value;
4 ;

following: O91, T22, A56, H47, L43. These codes represent different medical condition without a common cause such as breast infection, burn on the wrist and chlamydia infection.

The TabNet sensitivity analysis results are displayed in Figure 19 revealing that the change of ICD-10 codes has only a marginal effect on the models predictive power as the overall AUC values were all about 0.9. As the ICD-10 code was the most important features for the TabNet model having the overall highest value for the feature importance [3] across all three models this lack of change is unexpected. This could be explained by the overall low value of the ICD-10 feature importance of only 0.08.

4.5 Comparison of classification errors

All 3 models gave adequate results in predicting ischemic stroke, but none of the models were perfect. To analyse and compare the models further such value threshold for prediction scores is taken that for each model the number of true positive result for the test data is 61 and the number of false negative results are 24. This way all three models have made the same number of mistakes either by missing cases or wrongly classifying controls as cases. The Figure 20 shows that out of 61 true positive cases 52 are shared between three models. The Random Forest and TabNet share total of 6 cases that FastAI did not predict correctly. Likewise, the FastAI classified correctly 7 cases that both the Random Forest and TabNet missed. Both the Random Forest an TabNet had unique 2 cases that were predicted correctly using only that model.

A closer examination of the 24 false negative predictions of all models reveals that 14 cases are missed by all three approaches. These cases seem not to differ from the overall cases group of the study as described by the Table 4. Out of the 14 cases 5 were male and 9 females. The last measurement was taken in between of 6 to 996 days before stroke with the average of 369 days. The year of birth varies between 1920 to 1989 with the average of 1947. The number of LOINC codes available is between 21 to 67 with the

[3] The method for feature importance calculation differ for the FastAI and TabNet models.

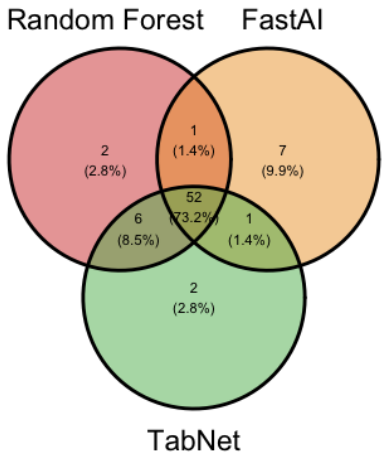


Figure 20. Overlap of true positive predictions of the 3 models (Random Forest upper left circle, FastAI tabular learner upper right circle, TabNet bottom circle) using the test data. Confidence thresholds for each model was select such that the number of true positive predictions would be 61 and false negatives 24.

Table 4. Comparison of the 14 false negatives cases for all models and all cases in the cohort

Feature	**FN avg ($\pm$ std)**	**All cases avg ($\pm$ std)**
Last measurement taken before stroke	369 ± 303 days	281 ± 268 days
Year of birth	1947 ± 12	1944 ± 13
Number of LOINC codes available	43 ± 15	47 ± 16
The proportion of ICD-10 codes available	0.57	0.76
The proportion of males	0.357	0.411

average of 43. The ICD-10 codes were available for 8 cases. All in all, the false negative cases seem not to stand out in any way from the cases that were correctly classified.

5 Discussion

This Chapter discusses the clinical and biomedical relevance of the results. Next, the limitations and future work related to this Thesis are described.

5.1 Comparing results

Risk factors (epidemiological or poly-genetic risk scores) with only modest discriminatory ability (AUC of 0.65) can provide sufficient stratification of risk to be useful for identifying high-risk individuals [60]. This indicates that all models created could be evaluated further for clinical utility as the lowest AUC on test data by the FastAI tabular learner was 0.88. This is answering Q1 that the occurrence of ischemic stroke can be predicted using the pre-existing EHR data using all 3 ML methods. Random Forest scored the highest AUC value using the test data. Although, the 10-fold cross validation results for TabNet looked promising the model's AUC value for test data was a bit lower. This indicates that usage of DNN did not give the performance increase hoped for. For the Q2 proposed the DNN methods used did not outperform the Random Forest. Similar AUC values for Random Forest and DNN models are not uncommon. In a study diagnosing Alzheimer's type dementia from blood samples the deep learning model produced AUC value of 0.85 while XGBoost resulted in AUC of 0.88 [61]. If the goal is to build a prediction model for ischemic stroke then the Random Forest method is preferred as it achieved the highest AUC value while also being easy to implement. Other ensemble based methods such as XGBoost, CatBoost and LightGBM could be implemented next as potential methods to increase the AUC value even further.

The results of the Thesis are promising as all models achieved very high AUC scores compared to other similar studies. The study using 3 US-based healthcare databases claimed AUC scores from 0.69 to 0.77 for predicting ischemic stroke in patients without atrial fibrillation in a 1 and 3 year time frames [7]. The model developed on Korean national health examination data for a 10-year stroke prediction claims to have AUC value of 0.83 for men and 0.82 for women [62].

The most important feature differed between DNN based models and Random Forest as the ICD-10 codes feature was most important for both the FastAI learner and TabNet and not used for Random Forest. Still out of the top 20 most important features the 5 features were shared by all 3 models. The research question Q3 was about most important features for the models and as we take a closer look at the 5 features important in all models we find links to previous studies of ischemic stroke risk factors. The hematocrit was found important for all 3 models. This is interesting as the association between hematocrit and the incidence of stroke has been a controversial topic and some studies have shown that higher levels of hematocrit are associated with a higher incidence of stroke [63]. The mean platelet volume and red blood cell distribution width which were again important in all models have been shown as risk factors for ischemic stroke [64, 65].

The explanation for why the pH of urine was a important feature for all models might not be as simple. On possibility is the usage of thiazide diuretics (ATC code C03 which was not considered in forming study cohort) which have shown to have some effect in alleviating (or protecting against) the ischemic stroke via blood pressure control. The usage of thiazide diuretics can change the pH of urine [66]. This should be researched further. One possibility is by removing the thiazide diuretics users from the study. Interestingly, while the estimate of age (varying between $\pm$ 1 year) was not important in all models but the year of birth was.

5.2 Limitations

The measurements taken from blood and urine samples used in this Thesis may not be the optimal parameters to measure for modelling the risk of ischemic stroke.

The selection of cases with stroke occurring up to 1000 days after the last measurement limits the capability of the model - the occurrence of stroke could be predicted no more than 1000 days after the last measurements. The impact on measurement times before occurrence of stroke has not been analyzed further.

Any biological interpretations based on the feature importance and overall models behaviours is not directly linked to biological relevance. The physicians order certain test based on their suspected diagnosis and the models may learn to predict the ischemic stroke based on the tests ordered. Even if the models don't provide insight into the underlying disease beyond physicians activity the model could still be beneficial for early discovery and disease risk evaluation. Moreover, the feature importance was evaluated differently for all models - when using a universal approach the most important features may change.

The amount of data used may not seem small but taking into account the high dimensionality, heterogeneity and sparseness of data more subject for the case and control groups would be beneficial for the DNN used. Despite this the models managed to generalize to test set.

The ensemble models AUC values were calculated on the test data and no k-fold cross validation process was involved in the evaluation of ensemble models. For selection of best ensemble model cross-validation could be used and then tested on another data set.

Finally, the ML algorithms compared to a human are more prone to the mistake of misclassification of adversarial examples [67, 68]. The semantics of the objects presented are not understood by the algorithm.

5.3 Future work

Firstly, the AUC value itself has no direct clinical interpretation. Other measures, such as the proportion of the population and the proportion of cases whose estimated risks

exceed clinically meaningful risk thresholds can be used to obtain more direct insight into the clinical utility of the models.

Secondly, the external validation could be used to prove the predictive power of the models. This means testing the models using sample of subject different from the original cohort.

Thirdly, and most importantly the models ability to influence patient outcomes could be measured [69]. This is affected by the models influence over clinicians and patients behavior.

Due to time limitations some popular ML methods for tabular data such as XGBoost, LightGBM and CatBoost were not implemented. In the future, these models could be explored with thorough hyper-parameter search and tuning. Additional sensitivity analysis could be done for other important features found in addition to the ICD-10 codes feature. As for the ensemble model created the model should be selected using k-fold cross validation and validated by a different test data.

Naturally, these methods used in this Thesis could be used to train models for other prevalent disease such as diabetes, Alzheimer's disease and even cancer. This Thesis focused on using the available EHR data as tabular type. The data could also be views as time series and other types of DNN could be tested such as long short-term memory architecture [70].

6 Conclusion

The aim of this Thesis was to **evaluate the available DNN architectures for predicting ischemic stroke using EHR data on the Estonian population**.

The three research questions that were stated at the beginning of the Thesis are the following:

Q1. Can the occurrence of ischemic stroke be predicted using pre-existing EHR data using 3 ML methods: Random Forest, FastAI tabular learner and TabNet?
Q2. Will the DNN models (FastAI, TabNet) outperform Random Forest for our tabular data?
Q3. What are the most important features for the predictive models?

The Thesis used EHR data from *Digilugu* which was pre-processed by STACC using data mining methods described in [50]. The **LOINC** and **ICD-10** values were extrapolated from the data and used to predict the ischemic stroke.

To tackle the **Q1** the stratified **10-fold cross validation** was first used to find the optimal hyper-parameters for each model. Next the model was trained using the best hyper-parameters found and validating using test data. All the models used resulted in high AUC values for the classification task. Best results were achieved by the Random Forest with the AUC value of 0.94 on the test data. This was followed by the AUC of 0.90 by the TabNet and FastAI tabular learner achieved the AUC value of 0.88 for the test data. Different ensemble models using the previous models were tested as well and a modest increase was achieved with final AUC value of 0.95. As all 3 models tested achieved high AUC values and generalized well on test data the occurrence of a ischemic stroke can be predicted using the pre-existing EHR data.

As the Random Forest scored the highest AUC value on the test data the answer to the **Q2** is that the DNN did not achieve a performance increase. Still, a closer look on classification errors by all 3 models reveals that Random Forest and TabNet shared more true positive predictions while FastAI tabular learner had some unique true positive predictions not classified by Random Forest nor the TabNet model. A combined model of FastAI and Random Forest achieved a AUC value of 0.95 indicating that even if the DNN-based models alone do not increase the performance of the Random Forest-based model an ensemble of multiple models can results in increased performance.

The top 20 most important features were found for every model to answer the **Q3**. These 5 features were important for all three models: **year of birth**, **hematocrit**, **pH of urine**, **platelet mean volume** and **erythrocyte distribution width**. Almost all of the features named have been associated with stroke by previous studies. The pH of urine has no direct link to stroke but could be associated with usage of thiazide diuretics, which can change the pH of urine and possible interact with occurrence of stroke. The effect of the thiazide diuretics usage should be further studied. For the DNN based models the ICD-10 code was the feature with the highest importance. As there are some ICD-10

codes present only in the cases (or controls) sensitivity analysis was performed for the importance of theICD-10 on the DNN models. This experiment demonstrated that the models predictive power was not solely dependent on the ICD-10 codes difference in 2 groups. The TabNet and Random Forest models relied less on ICD-10 codes missing in controls group and the FastAI model was influenced more by the ICD-10 code disparity between cases and controls.

These results highlight that EHR data can be a valuable resource for development of disease prediction models. In the future the clinical utility of the models for ischemic stroke prediction need to be evaluated.

References

[1] https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death, (visited: 01.04.2021).

[2] Grapov D., Fahrmann J., Wanichthanarak K., and Khoomrung S. Rise of deep learning for genomic, proteomic, and metabolomic data integration in precision medicine. *OMICS*, 22(10):630–636, oct 2018.

[3] Partho Sen, Santosh Lamichhane, Vivek B. Mathema, Aidan McGlinchey, Alex M. Dickens, Sakda Khoomrung, and Matej Orešič. Deep learning meets metabolomics: a methodological perspective. *Briefings in Bioinformatics*, 22(2):1531–1542, sep 2020.

[4] Chapman B.P., Lin F., Roy S., Benedict R.H.B., and Lyness J.M. Health risk prediction models incorporating personality data: Motivation, challenges, and illustration. *Personality Disorders*, 10(1):46–58, 2019.

[5] A. K. Boehme, C. Esenwa, and M. S. Elkind. Stroke risk factors, genetics, and prevention. *Circulation research*, 120(3):472–495, 2017.

[6] Jeena R. S. and Sukesh Kumar A. Stroke prediction models: A systematic review. *International Journal of Scientific and Engineering Research*, 9(4):1345–1348, 2018.

[7] Zhong Yuan, Erica A. Voss, Frank J. DeFalco, Guohua Pan, Patrick B. Ryan, Daniel Yannicelli, and Christopher Nessel. Risk prediction for ischemic stroke and transient ischemic attack in patients without atrial fibrillation: A retrospective cohort study. *Journal of Stroke and Cerebrovascular Diseases*, 26(8):1721–1731, 2017.

[8] Sercan O. Arık, Tomas Pfister, Google Cloud, AI, Sunnyvale, and CA. Tabnet: Attentive interpretable tabular learning. 1908.

[9] Jeremy Howard et al. fastai. `https://github.com/fastai/fastai`, 2018.

[10] Bahado-Singh R.O., Sonek J., McKenna D., Cool D., Aydas B., Turkoglu O., Bjorndahl T., Mandal R., Wishart D., Friedman P., Graham S.F., and Yilmaz A. Artificial intelligence and amniotic fluid multiomics: prediction of perinatal outcome in asymptomatic women with short cervix. *Ultrasound Obstet Gynecol*, 54(1):110–118, jul 2019.

[11] Taiga Asakura, Yasuhiro Date, and Jun Kikuchi. Application of ensemble deep neural network to metabolomics studies. *Analytica Chimica Acta*, 1037:230–236, 2018. Analytical Metabolomics.

[12] Wainberg M., Merico D., Delong A., and Frey B.J. Deep learning in biomedicine. *Nat Biotechnol*, 36:829–838, 2018.

[13] Angermueller C., Pärnamaa T., Parts L., and Stegle O. Deep learning for computational biology. *Mol Syst Biol*, 12:878, 2016.

[14] Min S., Lee B., and Yoon S. Deep learning in bioinformatics. *Brief Bioinform*, 18:851–869, 2017.

[15] Tiernan Ray. Fast.ai's software could radically democratize ai. *ZDNet*, oct 2016.

[16] Thomas Truelsen, Stephen Begg, and Colin Mathers. The global burden of cerebrovascular disease. *World Health Organization*, 01 2006.

[17] S. E. Kjeldsen, K. Narkiewicz, M. Burnier, and S. Oparil. The interstroke study: hypertension is by far the most important modifiable risk factor for stroke. *Blood pressure*, 26(3):131–132, 2017.

[18] Evgeny Sidorov, Dharambir K. Sanghera, and Jairam K. P. Vanamala. Biomarker for ischemic stroke using metabolome: A clinician perspective. *Journal of Stroke*, 21(1):31–41, jan 2019.

[19] J. Marshall, A. Chahin, and B. Rush. *Secondary Analysis of Electronic Health Records*. Springer Open, 2016.

[20] B. A. Goldstein, A. M. Navar, M. J. Pencina, and J. Ioannidis. Opportunities and challenges in developing risk prediction models with electronic health records data: a systematic review. *Journal of the American Medical Informatics Association*, 24(1):198–208, jan 2017.

[21] R. H. Dolin, L. Alschuler, C. Beebe, P. V. Biron, S. L. Boyer, D. Essin, E. Kimber, T. Lincoln, and J. E. Mattison. The hl7 clinical document architecture. *Journal of the American Medical Informatics Association*, 8(6):552–569, nov 2001.

[22] https://www.riigiteataja.ee/akt/121022017005?leiaKehtiv. Tervishoiuteenuste korraldamise seadus (21.04.2021).

[23] H. Liivlaid, N. Eigo, and S. Reisberg. Eriarstiabi haigestumusstatistika võrdlus tervise arengu instituudi ja eesti haigekassa andmetel. *Eesti Arst*, 98:17–26, 2019.

[24] e-Estonia briefing centre Electronic Health Records (e Health Records). https://e-estonia.com/solutions/healthcare/e-health-record/ (visited: 29.04.2021).

[25] https://www.elmy.ee/tooruhmad/loinc/dokumendid, (visited: 01.04.2021).

[26] WHO Collaborating Centre for Drug Statistics Methodology. Atc classification index with ddds, 2020.

[27] https://e-estonia.com/solutions/healthcare/e-prescription/, (visited: 21.04.2021).

[28] I. Landi, Glicksberg, B.S., H.C. Lee, S. Cherng, G. Landi, M. Danieletto, J.T. Dudley, C. Furlanello, and R. Miotto. Deep representation learning of electronic health records to unlock patient stratification at scale. *npj Digital Medicine*, 3(96), jul 2020.

[29] G. Hripcsak and D. J. Albers. Next-generation phenotyping of electronic health54 records. *Journal of the American Medical Informatics Association*, 20(1):117–121, 2012.

[30] Thomas M. Mitchell. *Machine learning*. McGraw-Hill, Inc., 1997.

[31] Peter Flach. *Machine learning. The Art and Science of Algorithms that Make Sense of Data.* Cambridge University Press, 2012.

[32] Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

[33] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, 1958.

[34] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[35] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning internal representations by error propagation. *Technical report, California Univ San Diego La Jolla Inst for Cognitive Science*, 1985.

[36] Léon Bottou. Large-scale machine learning with stochastic gradient descent. *In-Proceedings of COMPSTAT'2010*, pages 177–186, 2010.

[37] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[38] https://www.kaggle.com/shivamb/data-science-trends-on kaggle. Historical data science trends on kaggle., 2019 (visited: 07.05.2021).

[39] Pastor Jullian Fabres, Cassandra Collins, Timothy R. Cavagnaro, and Carlos M. Rodríguez López. A concise review on multi-omics data integration for terroir analysis in vitis vinifera. *Frontiers in Plant Science*, 8:1065, 2017.

[40] Bram Peter Prins, Liis Leitsalu, Katri Pärna, Krista Fischer, Andres Metspalu, Toomas Haller, and Harold Snieder. Advances in genomic discovery and implications for personalized prevention and medicine:estonia as example. *Journal of Personalized Medicine*, 11(358), 2021.

[41] J. Hestness, S. Narang, N. Ardalani, G. F. Diamos, H. Jun, H. Kianinejad, M. M. A. Patwary, Y. Yang, and Y. Zhou. Deep learning scaling is predictable, empirically. 2017.

[42] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015.

[43] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization?, 2019.

[44] Tambet Matiisen. The use of embeddings in openai five, https://neuro.cs.ut.ee/the-use-of-embeddings-in-openai-five/, (visited: 21.04.2021).

[45] Cheng Guo and Felix Berkhahn. Entity embeddings of categorical variables. 2016.

[46] Leslie N. Smith. Cyclical learning rates for training neural networks. 2017.

[47] Grankin M. fast_tabnet (v0.2.0), 2020.

[48] Dua D. and Graff C. Uci machine learning repository, 2017.

[49] Hadley Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016.

[50] S. Laur, A. Ott, D. Särg, and J. Vilo. Analysis cleaning pipeline for estonian epicrisis documents. 2021. avaliable from authors.

[51] Toomas Haller. Popsel. `http://www.toomashaller.com/allele.html#POPSEL`, 2017.

[52] Kristjan Korjus, Martin N. Hebart, and Raul Vicente. An efficient data par-titioning to improve classification performance while keeping parameters interpretable. *PloS one*, 11(8):e0161788, 2016.

[53] Tom Fawcett. An introduction to roc analysis. *Pattern Recognition Letters*, 27:861–874, 2006.

[54] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[55] Tim Miller. Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*, 267:1–38, 2019.

[56] Been Kim, Rajiv Khanna, and Oluwasanmi O Koyejo. Examples are not enough, learn to criticize! criticism for interpretability. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

[57] Nembrini S., König I.R., and Wright M.N. The revival of the gini importance? *Bioinformatics*, 34(21):3711–3718, 2018.

[58] Aaron Fisher, Cynthia Rudin, and Francesca Dominici. All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously, 2019.

[59] Leo Breiman. Random forests. *Machine Learning*, 45(5):1–38, 2001.

[60] P. Maas, M. Barrdahl, A. Joshi, and et al. Auer, P. Breast cancer risk from modifiable and nonmodifiable risk factors among white women in the united states. *JAMA Oncology*, 2(10):1295–1302, 2016.

[61] Min Kim, Stuart Snowden, Tommi Suvitaival, Ashfaq Ali, David J. Merkler, and et al. Primary fatty amides in plasma associated with brain amyloid burden, hippocampal volume, and memory in the european medical information framework for alzheimer's disease biomarker discovery cohort. *Alzheimers Dement.*, 15(6):817–827, jun 2019.

[62] Jae woo Lee, Hyun sun Lim, Dong wook Kim, Soon ae Shin, Jinkwon Kim, Bora Yoo, and Kyung hee Cho. The development and implementation of stroke risk prediction model in national health insurance service's personal health record. *Computer Methods and Programs in Biomedicine*, 153:253–257, 2018.

[63] Yang R., Wang A., and Ma L. et al. Hematocrit and the incidence of stroke: a prospective, population-based cohort study. *Ther Clin Risk Manag.*, 14:2081–2088, oct 2018.

[64] F. Mayda-Domaç, H. Misirli, and M. Yilmaz. Prognostic role of mean platelet volume and platelet count in ischemic and hemorrhagic stroke. *Journal of stroke and cerebrovascular diseases : the official journal of National Stroke Association*, 19(1):66–72, oct 2010.

[65] Li B., Liu S., Liu X., Fang J., and Zhuang W. Association between red cell distribution width level and risk of stroke: A systematic review and meta-analysis of prospective studies. *Medicine (Baltimore)*, 99(16):66–72, 2020.

[66] Messerli F.H., Grossman E., and Lever A.F. Do thiazide diuretics confer specific protection against strokes? *Arch Intern Med.*, 163(21):2557–2560, nov 2003.

[67] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks, 2014.

[68] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, 2015.

[69] J. Usher-Smith, J. Emery, W. Hamilton, S. Griffin, and F. Walter. Risk prediction tools for cancer in primary care. *British Journal of Cancer*, 113(12):1645–1650, 2015.

[70] Wutong Wang, Hong Wang, Yongqiang Song, and Qian Wang. Mcpl-based ft-lstm: Medical representationlearning-based clinical prediction modelfor time series events. *Special Section on Data-Enabled Intelligence for Digital Health*, 7:70253–70264, jun 2019.

Appendix

I. Licence

Non-exclusive licence to reproduce thesis and make thesis public

I, **Siim Kurvits**,
*(*author's name)

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to

   reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

   Prediction Models of Ischemic Stroke Using Deep Neural Networks,
   *(*title of thesis)

   supervised by Toomas Haller and Ardi Tampuu.
   *(*supervisor's name)

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Siim Kurvits
14/05/2021

II. Neural network architectures in detail

Architecture of the FastAI tabular learner

```
TabularModel (Input shape: 2048 x torch.Size([2048, 149]))
============================================================================
Layer (type)         Output Shape         Param #    Trainable
============================================================================
                     2048 x 3
Embedding                                 9          True
----------------------------------------------------------------------------
                     2048 x 49
Embedding                                 21854      True
Dropout
BatchNorm1d                               298        True
BatchNorm1d                               402        True
----------------------------------------------------------------------------
                     2048 x 500
Linear                                    100500     True
ReLU
BatchNorm1d                               1000       True
----------------------------------------------------------------------------
                     2048 x 1000
Linear                                    500000     True
ReLU
----------------------------------------------------------------------------
                     2048 x 2
Linear                                    2002       True
----------------------------------------------------------------------------

Total params: 626,065
Total trainable params: 626,065
Total non-trainable params: 0

Optimizer used: <function Adam at 0x7f50a48c5d40>
Loss function: FlattenedLoss of CrossEntropyLoss()
```

Architecture of the TabNet

```
TabNetModel (Input shape: 2048 x torch.Size([2048, 149]))
==========================================================================

Layer (type)          Output Shape          Param #     Trainable
==========================================================================

                      2048 x 3
Embedding                                   9           True
--------------------------------------------------------------------------

                      2048 x 49
Embedding                                   21854       True
Dropout
BatchNorm1d                                 298         True
BatchNorm1d                                 402         True
--------------------------------------------------------------------------

                      2048 x 512
Linear                                      102912      True
Linear                                      131072      True
Linear                                      102912      True
BatchNorm1d                                 1024        True
Linear                                      131072      True
BatchNorm1d                                 1024        True
Linear                                      131072      True
BatchNorm1d                                 1024        True
Linear                                      131072      True
BatchNorm1d                                 1024        True
Linear                                      102912      True
Linear                                      131072      True
Linear                                      102912      True
BatchNorm1d                                 1024        True
Linear                                      131072      True
BatchNorm1d                                 1024        True
Linear                                      131072      True
BatchNorm1d                                 1024        True
Linear                                      131072      True
BatchNorm1d                                 1024        True
Linear                                      102912      True
Linear                                      131072      True
Linear                                      102912      True
BatchNorm1d                                 1024        True
Linear                                      131072      True
BatchNorm1d                                 1024        True
Linear                                      131072      True
BatchNorm1d                                 1024        True
Linear                                      131072      True
```

```
BatchNorm1d                              1024        True
Linear                                   102912      True
Linear                                   131072      True
Linear                                   102912      True
BatchNorm1d                              1024        True
Linear                                   131072      True
BatchNorm1d                              1024        True
Linear                                   131072      True
BatchNorm1d                              1024        True
Linear                                   131072      True
BatchNorm1d                              1024        True
----------------------------------------------------------------------------

                    2048 x 201
Linear                                   25728       True
BatchNorm1d                              402         True
Entmax15
----------------------------------------------------------------------------

                    2048 x 201
Linear                                   25728       True
BatchNorm1d                              402         True
Entmax15
----------------------------------------------------------------------------

                    2048 x 201
Linear                                   25728       True
BatchNorm1d                              402         True
Entmax15
----------------------------------------------------------------------------

                    2048 x 2
Linear                                   256         True
----------------------------------------------------------------------------


Total params: 3,038,041
Total trainable params: 3,038,041
Total non-trainable params: 0

Optimizer used: <function Adam at 0x7fb0dca39710>
Loss function: FlattenedLoss of CrossEntropyLoss()
```

III. Embedding of the ICD-10 value

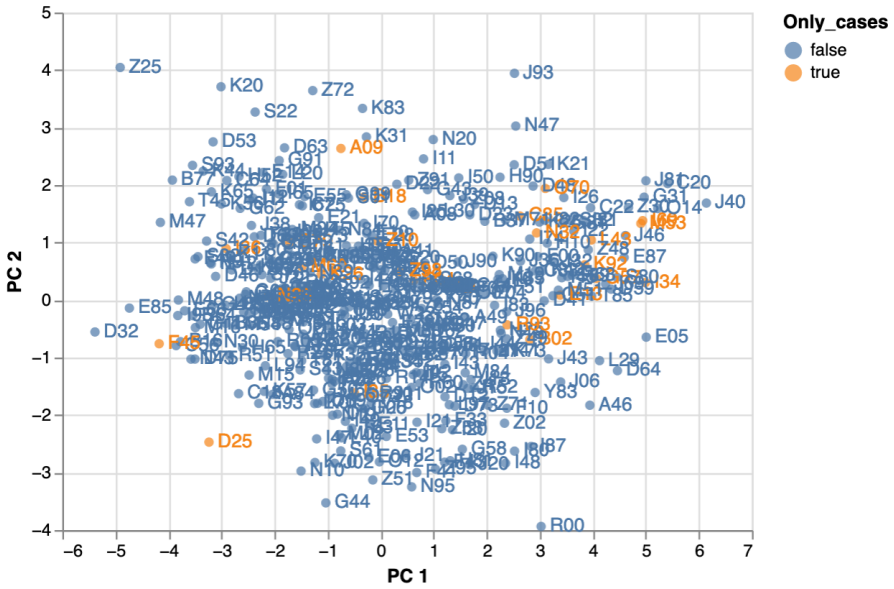


Figure 21. First 2 principal components calculated for the ICD-10 embedding created by the FastAI neural network. Orange dots represent ICD-10 values that were available only in the cases group.

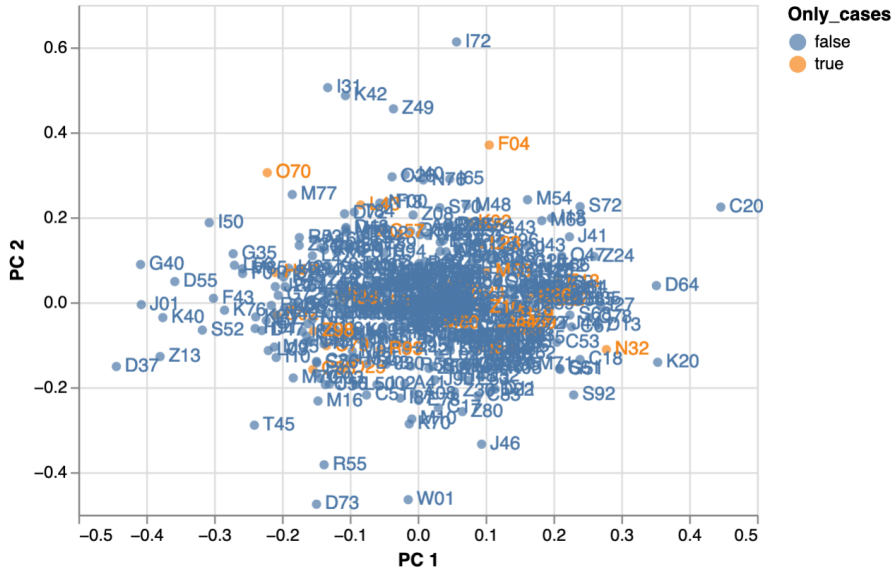


Figure 22. First 2 principal components calculated for the ICD-10 embedding created by the TabNet neural network. Orange dots represent ICD-10 values that were available only in the cases group.