Vrije Universiteit Amsterdam

Universiteit van Amsterdam

Master Thesis

Application and Assessment of Prediction Models using Deep Learning on Dutch Unstructured Clinical Text

Author: Gyan de Haan (ghn231)

1st supervisor:	Peter R. Rijnbeek
Academic supervisor:	Adam S.Z. Belloum
daily supervisor:	Tom M. Seinen
2nd reader:	Iacer C. A. C. Calixto

A thesis submitted in fulfillment of the requirements for the joint UvA-VU Master of Science degree in Computer Science

July 11, 2022

Abstract

Prediction models trained on electronic health records are gaining popularity, they enable the development of personalized treatments and risk factors for clinical decision making. The objective of this study is to assess the performance of deep neural network models for multiple patient level prediction problems on Dutch electronic health records, using both structured data and unstructured text data. A convolutional neural network, recurrent neural network, and transformer model are trained on these prediction problems. Multiple text preprocessing methods are evaluated and used. Evaluation of the deep learning models is performed over three prediction problems. The deep neural networks are compared with traditional machine learning models, and with deep neural networks that are trained on structured data. The results show that a convolutional neural network, trained on unstructured text has the best performance over all prediction problems.

Keywords: *clinical prediction model, deep neural networks, natural language processing, machine learning, electronic health records*

Acknowledgements

I would like to thank my committee for continuous supervising and support throughout my thesis. Tom has been a great daily supervisor who did not shy away from theoretical discussion and setting harsh deadlines when needed. I would like to thank Peter for providing the opportunity to do this research at his department and using the department resources. The weekly meetings with Adam were highly appreciated for his guidance and advice. Furthermore, I would like to acknowledge the Medical Informatics department of the ErasmusMC as a whole for making my thesis an exciting, wonderful experience and supporting me along the steps of research. Specifically I would like to thank Henrik John for providing the dementia cohort inclusion and exclusion criteria. Finally, I would like to acknowledge my housemates and family that supported me throughout this thesis.

Contents

List of Figures

List of Tables

1

Introduction

Usage of observational data from EHRs to build patient-level prediction models

Observational healthcare data in electronic health records (EHRs) are gaining popularity as data sources for studying disease progression, quality improvement and creating prediction models (1, 2). Prediction models on a patient level enable the development of personalizing treatments and risk factors for clinical decision making (1). A prediction task can be defined as using a labelled dataset, consisting of a set of predictive variables, to learn the mapping of these variables to the correct labels. This can be done by defining an at-risk target group of patients, that will get a positive label if they experience a specific outcome during a specified time window, also called the outcome population (1). Figure 1.1 shows this process. The benefits of using EHR data for patient-level prediction are the availability to use data from a substantial number of patients, at multiple time points, which have a distribution being reflective of the real-world (as opposed to traditional research cohorts) and the ability to study a wide range of clinical outcomes (2).

Structured data and unstructured data

EHRs consist of both structured data: such as coded clinical conditions, demographic information, and measurements, as well as unstructured text data: general practitioners notes, discharge letters and specialist communications(3). This unstructured text data is abundant in most EHR systems, due to its more narrative, engaging nature (3). Text contains detailed patient narratives and allows for expression of the physician, while the clinical codes can be too limited and unable to convey the necessary nuances (3). For example, additional information is needed to convey indecisive diagnoses, which need qualification or that rely upon uncommon symptoms (3).

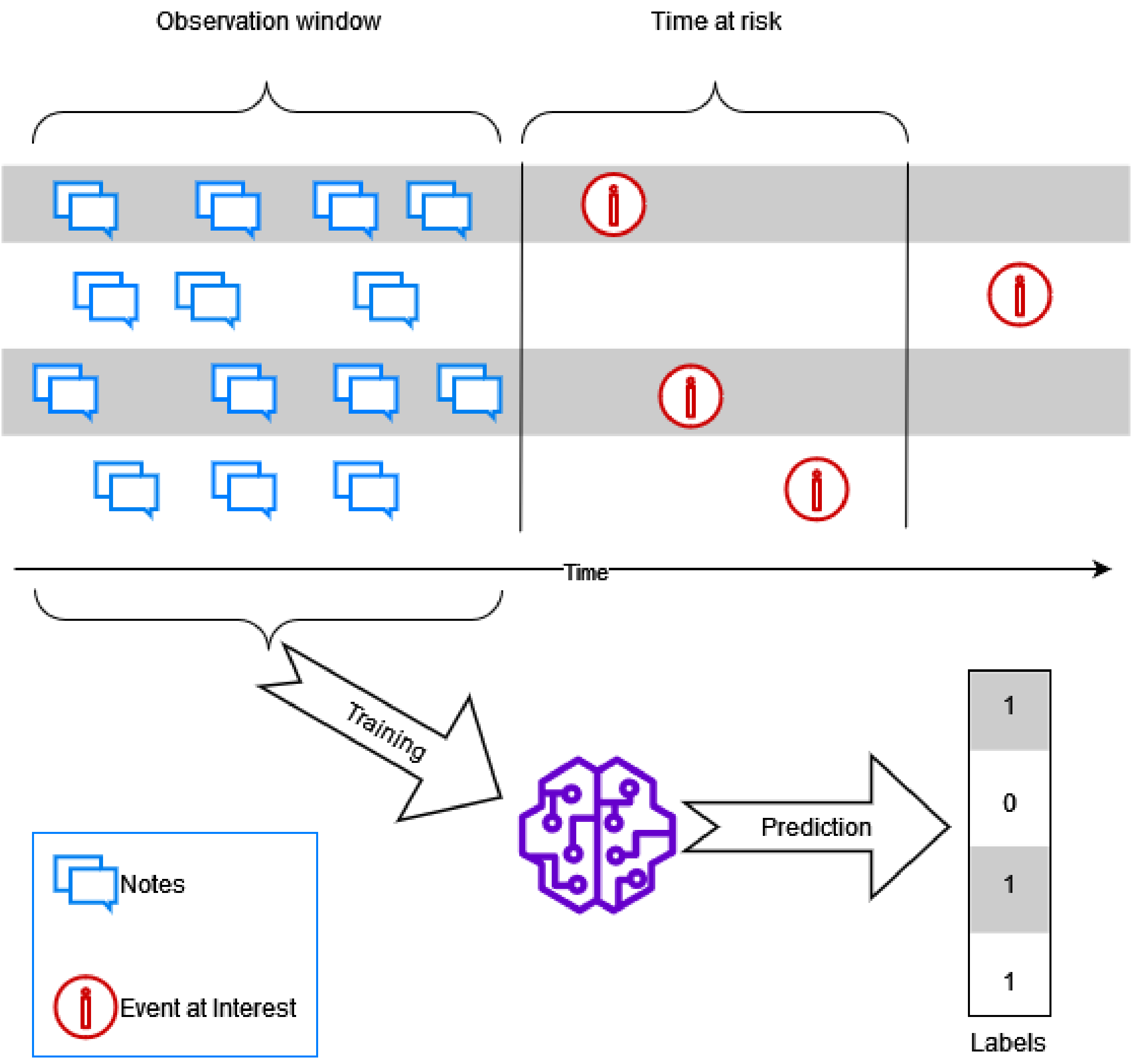

Figure 1.1: Patient level prediction problem where text input is given to a model. The trained model predicts the class labels. The true labels are derived from the occurrence of the event at interest at the time of risk window for every subject.

Use of NLP to incorporate the unstructured data in a Patient-level Prediction model

Enabling the use of unstructured text by computers is the focus of the field Natural Language Processing (NLP). This field was founded over 50 years ago (4), but many challenges still exist. The usage of NLP algorithms on clinical text is arduous due to the data being privacy sensitive. The confidentiality issues create difficulty in creating suitable data to develop and evaluate algorithms. The nuances of medical text are supplementary challenges. Unfortunately, even though federated learning solved many confidentiality issues for this research (5), the privacy issues are evident in the whole medical field, which still results in a lack of publicly available data for the NLP algorithms to be evaluated on. For NLP tools and algorithms English is the most common language used, especially in the medical field (6). Even though there has been some NLP research on Dutch clinical text (6, 7), the usage of machine-based NLP and deep learning methods for prediction on Dutch clinical text is not well substantiated, with some studies translating Dutch to English to bypass this problem (8).

Deep learning and traditional machine learning methods

Artificial Intelligence (AI) is a field that focuses on enabling computers, intelligent agents, to achieve intelligence (9). One of the major subsets in AI is Machine Learning (ML). ML uses a learning process to train a model. A subset of ML is Deep Learning (DL). DL is deep, due to using multiple layered neural networks to extract (high level) features from an input. DL has many applications including, image processing, reinforcement learning and natural language processing. Deep learning techniques have been used for NLP in the clinical domain, due to the efficient processing and ability to gain state-of-the-art results (10). Deep learning architectures that are most used in this paradigm are recurrent neural networks (RNNs), convolutional neural networks (CNNs), attention models and adversarial learning models. Attention models gained popularity, most notably due to BERT (10). Commonly used embeddings as input for deep neural networks (DNN) are word2vec (11), Glove (12) and Fasttext (13), some models such as BERT (14) train their own word embeddings.

Thesis Objective

Observational Health Data Science (OHDSI) is a multi-stakeholder, interdisciplinary collaborative to bring out the value of health data through large-scale data-analytics. They provide open-source software packages to perform research on data in the OMOP-CDM format [1]. Peter Rijnbeek is co-leading the Patient-Level Prediction working group within OHDSI together with Jenna Reps that built a framework on top of the OMOP-CDM for large-scale development and validation of prediction models across the world. Tom Seinen has been developing an extension to extract text features from clinical notes in the OMOP-CDM to be used in the Patient-Level Prediction framework. The objective of this thesis is to assess if deep learning methods for text analysis are beneficial to add to the OHDSI analytical framework.

Research Questions

There are three challenges that were addressed in the research. First, there has been little research performed on using deep learning on Dutch clinical data. Second, the EHR text data has a relative high occurrence of spelling errors, abbreviations, and incomplete sentences. Third, it is unclear if the unstructured data contains equal or more information than the structured data. In order to reach this goal, the following research questions will be addressed.

1. How do different DNN architectures, such as RNNs, CNNs and attention models, compare in performance to traditional Machine Learning models, using unstructured clinical text data?

   This is the main research question of this thesis. In answering this research question the most suitable DNN architectures for unstructured Dutch clinical data will be presented, while analysing if they have a competitive performance.

2. What is the effect of different pre-processing methods, such as embeddings, spelling corrections, abbreviations, on the performance of the artificial intelligence models?

   Answering this sub-question will help determine the effect of different pre-processing methods on the model performance. The pre-processing methods influence the amount and quality of features that are used to train the models.

3. How does the use of text data compare to the use of structured data ?

   This can be considered a sub-question of the main research question, where the performance of using unstructured to structured data is investigated.

[1] https://ohdsi.org/

Background

This chapter summarizes and describes the relevant concepts in deep learning and NLP methods that have been researched. First, the NLP pre-processing methods are introduced, followed by the explanation of spelling correction, abbreviation correction algorithms and several text representation methods. Then the relevant deep learning methods that are introduced, including recurrent neural networks, convolutional neural networks, and transformer models.

EHR

Risk prediction algorithms in traditional clinical studies have been developed from large cohort studies (2). These research cohort data are well standardized. In contrast, EHR data often contains much noise: missing data, unstructured data, non-normalized data. EHR text data includes spelling mistakes, abbreviations and offers a challenge to be represented in a numerical way.

Spelling faults

Spelling errors can be classified in six types. Words are spelled with missing characters such as *liason* instead of *liaison*, additional characters such as *publically* instead of *publicly*, incorrect characters such as *supercede* instead of *supersede*, swapped characters such as *recieve* instead of *receive*, replacement with similar words such as *forward* instead of *foreword* or spacing mistakes such as *deter gent* instead of *detergent*. There are several methods to correct spelling mistakes. Cammel, et al. (8) uses a combination of the Norvig algorithm (15), with a Dutch dictionary such as (16) and domain specific vocabulary and knowledge (SNOMEDCT) (17). The Norvig's algorithm does not only identify misspelled words, but it also corrects them to the *correct* word, the word with the highest probability as determined in the Norvig's algorithm for being correct. Although Norvig's algorithm is frequently mentioned and used, it has several drawbacks: It assumes that any word within a true Damerau-Levenshtein distance of 1, where a single transposition of two adjacent characters, insertion, deletion, substitution are accepted edits on substrings, is infinitely more likely to be the correct spelling than a word with a true Damerau-Levenshtein distance of 2. This can result in a false correction for example when *juse* is corrected to *just* but should have been corrected to *juice*. The algorithm has an expensive compute step when it calculates all possible candidate terms within the maximum allowed true

Damerau-Levenshtein distance, which makes the algorithm slow. Due to the above mentioned computationally expensive step Norvig's algorithm has a maximum edit distance of 2 in order to be practically used. Symspell (18) addresses the time scaling problem of Norvig's algorithm. Using a delete only method to calculate the restricted Damerau-Levenshtein distance, e.g., substrings cannot be modified twice, Symspell can achieve a one million times speed up compared to Norvigs algorithm. Another method, as explained in Karthikeyan, et al. (19) is, filtering out domain specific entities using a knowledge base, then using spellcheckers such as pyspellchecker [1] and hunspell [2] to identify misspelled words. These incorrect words are corrected based on a transformer model, such as RoBERTa (20). The transformer models can use suggestions to pick from or be open in predicting the correct spelled word. Since Dutch clinical text data is not available in large well structured, noise-free corpora, training a transformer model might prove difficult, this if further explored in the Deep learning Architectures section. Other ideas using word embeddings such as word2vec trained on the data, including the incorrect words, and making use of the arithmetic nature of embeddings to resolve misspelled words as described in multiple AI blogs [3] are proposed. This approach works on the underlying assumption that the embedding space of the spelling mistakes is a "constant" displacement from the embedding space of the correct spelling.

Abbreviations

Spelling correction methods have difficulties in correcting abbreviations, therefore abbreviations are regarded as a separate problem from misspelled words. Abbreviations are also domain dependent, used frequently and consistently in clinical texts, and are occasionally be part of knowledge bases such as SNOMEDCT (17), . Since abbreviations are not part of a language dictionary, similar to misspellings, a method of identifying and correcting abbreviations is needed. Linguistic and domain heuristics can be used to identify the abbreviations. An abbreviations is most of the time a permuted sub string of the abbreviated word, therefore short and using letters from the original word. One way of dealing with abbreviations is by adding them to the vocabulary as is done in Cammel, et al. (8), one can also use heuristics, such as exact consonant matching, to find the corresponding words,

[1] https://pypi.org/project/pyspellchecker/

[2] http://hunspell.github.io/

[3] https://forums.fast.ai/t/nlp-any-libraries-dictionaries-out-there-for-fixing-common-spelling-errors/16411/38 and https://edrushton.medium.com/a-simple-spell-checker-built-from-word-vectors-9f28452b6f26

such as dgn, for dagen (days in Dutch). Heuristics can also be used to assess which abbreviations need corrections, as certain abbreviations are used so frequently it can also be beneficial to keep the abbreviation as is. A trade-off must be made between reducing data noise or keeping detailed data.

Text representation

NLP is known to not have one solution for all problems. Therefore, multiple features and combinations of covariates (features) need to be explored. Common text features are bag-of-words, topic models, word embeddings, contextual embeddings, N-gram analysis and POS tags (2, 3, 21). Additional data sources could be necessary to create features, for example SNOMEDCT can be used for the extraction of mentioned concepts or used in training word embeddings. Challenges in the NLP domain for non-English languages also show that the use of NLP algorithms will impact the downstream tasks (22). One example is the language specific tokenizer instead of a general tokenizer. Tokenizing the text is a major step in the pipeline, but there are a variety of choices: Domain specific heuristic made tokenizer, sub-word tokenizers or general language tokenizers. The tokenizing technique influences the features dimensionality and content representation. For example, stemming reduces words to their word stem, reducing the number of features but also the amount of detail. The choice between complex or simple text representations is not straightforward, it has been shown that simpler text representations such as bag-of-words and tf-idf can have better results than more complex representations, such as embeddings, on binary classification tasks in the clinical domain (21).

RNN and CNN in the clinical NLP domain

Convolutional Neural Networks (CNNs) train on data through convolutions and pooling functions. CNNs have been commonly used to model signals, such as image, and have proven to be quite successful in this regard. The convolutions can also be applied on word or document embeddings. These operations made convolutional neural networks popular among researchers using deep learning for text classification (10, 23). One model type that has been more popular among clinical researchers is the recurrent neural network (RNN)(10, 23). This model does not use convolutions and pooling layers, instead it uses LSTM cells or GRU cells to model time series and sequences. The RNN can be configured in a bi-directional setup, traversing the input sequence in both directions, which has shown to improve the performance of the RNN model.

Transformer models

Since the introduction of transformer models by Viswani, et al.(24) there has been an increase in NLP language models . BERT (14) has been one of the most influential language models in the NLP domain, that has inspired many other models, such as RoBERTa, which uses alternative parameters and training methods compared to BERT (25). BERT and RoBERTa are both models that learned general language patterns while pre-training. Transformer models can be fine-tuned on domain- or problem-specific data for specific NLP tasks. In the Dutch medical domain, there are several pre-trained models that are of interest for this study. These models are either Dutch language models or medical domain language models or a combination thereof. BERTje (26), a Dutch monolingual version of BERT, and RoBERT (27), a Dutch monolingual version of RoBERTa, are both trained on only Dutch corpora. BioBERT (28) and SciBERT (29) are examples of BERT models trained on biomedical and scientific data, respectively. Where clinical data is significantly different from average Dutch documents, due to shorter sentences, spelling mistakes, abbreviations, keywords without verbs and a high percentage of multilingual terms, BERT models trained on biomedical data can be relevant. These models have been shown to increase performance on downstream NLP tasks in the biomedical domain (28). Recently, a model based on the RoBERTa architecture, trained on Dutch clinical data, has been published, called MedRoBERTa.nl (30). The model was evaluated on a Named Entity Recognition problem, therefore at this moment there is no indication for the performance of MedRoBERTa.nl on text classification tasks. MedroBERTa.nl is nevertheless promising due to being the only transformer model to date that has been trained on Dutch clinical data.

Deep learning Architectures challenges

RNNs, CNNS and attention models are all used in the clinical prediction field (10). Although these models differ in architecture, in the input features they need, in their popularity, and effectiveness over the past years (10), they share challenges. Similar to the NLP problem, there is not one solution, pipeline, architecture or one set of hyper parameters that performs well on all tasks. The optimization of hyper parameters and finding hyper parameters will be adding to the success or failure of these algorithms. As an objective of this research is to enable researchers, through OHDSI, to use deep learning techniques for their own studies and data, generalisation will be key. Finding the correct parameters and fine tuning these will therefore be of major importance. Deep learning models are known to require large scale data sets. The question is whether the text data is of high enough quality and quantity for these models to perform well, e.g., better than Machine Learning models and models for structured data. One way to tackle this problem of quality and quantity is by using a form of transfer learning. Transfer learning enables fine-tuning an existing model on the data or problem of interest. In the case of transformers this is possible by training only the last layers of a complex architecture. This reduces the needed quantity and quality of training data. This might also be a solution when using attention models, such as transformer models, on Dutch text data. Research has not shown yet if transfer learning will help performance in the clinical domain, especially the pre-training on Biomedical text is contested, but this idea is still promising (31, 32, 33, 34). There is a high variety in how transfer learning or pre-training is adopted and used, but further research is needed. Specific BERT and RoBERTa models for clinical data have been trained (35), also using Dutch clinical text(27).

Previous Results

Deep learning models for clinical prediction problems, presented in research, have been primarily trained on English (36). Mohamaddi, et al. (37) report an area under the receiver operating curve (AUROC) of 0.894 for a feed forward model and a AUROC of 0.735 for a BERT transformer model, on predicting hospital readmission in 30 days. A BERT model has also been trained to predict in-patient admission by Tahayori, et al. (38), which had an AUROC of 0.88. A CNN trained by Si, et al. (39) had a AUROC of 0.93 on predicting mortality within 30 days of hospital admission. In Krishnan, et al. (40) the best model was a feed forward model with an AUROC of 0.98 on predicting ICU mortality prediction from unstructured ECG text reports. In Grnarova, et al. (41) a CNN had the best AUROC performance of 0.858 on predicting ICU mortality prediction within 30 days. In Obeid, et al. (42), a CNN model, with a reported AUROC of 0.882, outperformed a RNN model on predicting intentional self-harm event, within 3 months. This literature shows that there has not been one deep learning architecture, trained on text, which performs best over all clinical prediction problems and the performance of one model can vary depending on the prediction problem and setting.

Menger, et al. (43) compares a RNN and a CNN with traditional Machine Learning models on predicting inpatient violence, based on Dutch clinical text. An improvement in performance for the deep learning models over the traditional Machine Learning models was observed. The RNN outperformed the CNN on this prediction problem.

2

Methods

In this chapter the design of the experiments is discussed. Starting with the data that was used in this thesis, followed by the definitions of population cohorts that were used in the experiments. Furthermore, the text pre-processing methods that were evaluated and the resulting text representations are described. Lastly, the models that were trained and the evaluation metrics that are used for the training and testing are presented.

Dataset and setting

In this research the Integrated Primary Care Information (IPCI) database was used. The IPCI database is an observational database containing data from computer-based patient records of selected general practitioners throughout the Netherlands, since 1992 [1]. The IPCI database consists of 350 GP practices, which are mainly located in the central part of the country, the Randstad. The number of active patients is 1.4 million, which comprises 8.1% of the Dutch population. The characteristics of the IPCI database can be found in Table 2.1 The database is mapped to the OMOP Common Data Model (CDM) format, enabling research within the Observational Health Data Science (OHDSI) community.

Population cohorts and prediction problems

The prediction problems are defined using patient cohorts. These cohorts are selected using the ATLAS application [2]. The cohorts are then processed to get label data and transformed to the right data format, e.g., dates to a date-time format, text to a string format. The experiments are run on three prediction problems: hospital readmission within 32 days, dementia within 5 years for subjects above 50 years old, end of life conversations within a

[1] https://www.ipci.nl/index.php/about

[2] https://atlas-demo.ohdsi.org/

Table 2.1: IPCI data characteristics

IPCI Database	Categories	Value
Demographics	Number of persons	2529355
	Sex	48.8% male, 51.2% female

year for subjects above 50 years old. Following literature (23, 35), subjects that passed away during the time at risk are excluded from the data cohorts. The population characteristics for these cohorts can be found in Tables 2.2, 2.3 and 2.4. The unstructured text data for the patient cohorts is selected using the ATLAS application. In this tool the cohorts can be defined by inclusion and exclusion criteria. These criteria are then used to filter the data on the OMOP CDM codes. The data is made available to the deep learning models in python by the package pyscopg2 [1]. The corresponding structured data for the patient cohorts is also selected using the ATLAS application, with the covariates being extracted through the use of the FeatureExtractionPackage [2] and made available to the models in python by the pyscopg2 package. These processes are visualised in Figure 2.1. The hospital readmission within 32 days prediction problem is chosen, due to being a regular occurrence in literature. The second prediction problem, on predicting dementia within 5 years, was selected due to being a more specific condition. The general practitioners at the ErasmusMC asked the medical informatics department to make a model for the end of life conversations within one year prediction problem. The three prediction problems are chosen on basis of occurrence in literature and variation in models to get generalize results.

Pre-processing methods

Spelling

To reduce the number of features, e.g., the number of unique tokens, a spelling correction was applied. To select the appropriate spelling correction algorithm, one would normally use a golden standard to evaluate or even train an algorithm. Due to the lack of such a dataset for this project, one common method was chosen, Symspell.

[1] https://www.psycopg.org/

[2] https://raw.githubusercontent.com/OHDSI/FeatureExtraction/main/extras/FeatureExtraction.pdf

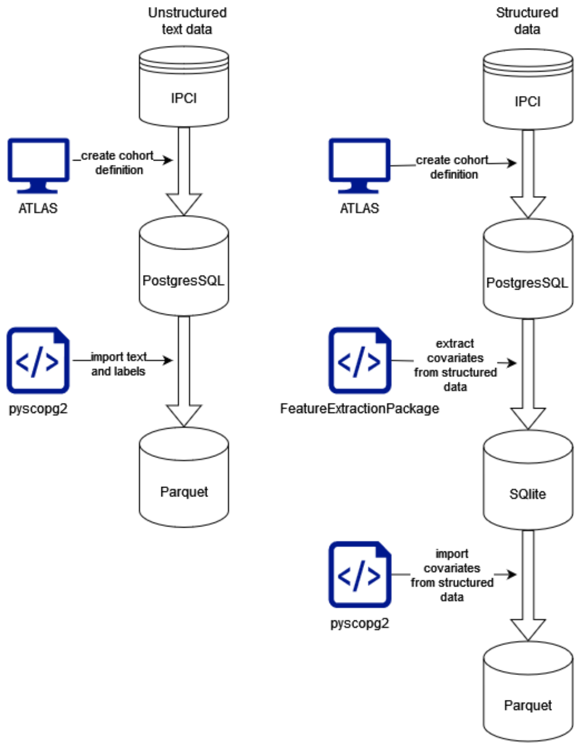

Figure 2.1: On the left the data cohort process for the unstructured data is shown. On the right the corresponding process for the structured data is shown.

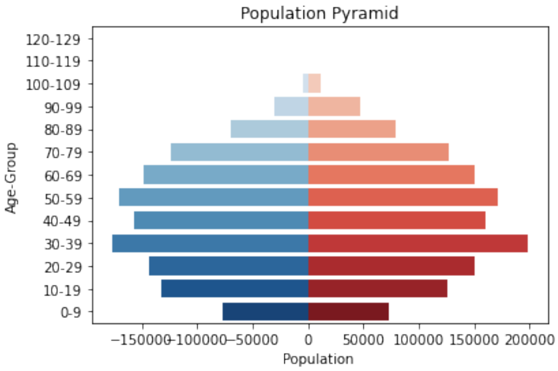

Figure 2.2: Population Pyramid for the IPCI database

Table 2.2: Hospital readmission characteristics

Hospital readmission	Categories	Value
Cohort definition	Problem statement	Predicting hospital readmission within 32 days
	Index event	In patient visit
	Time at risk	32 days
	Observation window	31 days
Demographics	Number of persons	113072
	Sex	46.1% male , 53.9% female
Text	Cases without notes	13273 from 188608, 387 outcome cohort, 12886 target cohort

Table 2.3: Dementia characteristics

Dementia	Categories	Value
Cohort definition	Problem statement	Predicting dementia in 5 years for subjects that are over 50 years old
	Index event	last GP visit in 2012-2014 for subject between 50 - 84 years old
	Time at risk	5 years
	Observation window	1 year
Demographics	Number of persons	160468
	Sex	46.5% male, 53.5% female
Text	Cases without notes	212 from 160468, 0 outcome cohort, 212 target cohort

Table 2.4: End of Life conversations characteristics

End of Life Conversations	Categories	Value
Cohort definition	Problem statement	Predicting the need of an end of life conversation within one year for subjects that are at least 50 years old
	Index event	last GP visit in 2019 for subject between 50 - 94 years old
	Time at risk	1 year
	Observation window	1 year
Demographics	Number of persons	413215
	Sex	46.2% male, 53.8% female
Text	Cases without notes	14811 from 413215, 25 outcome cohort, 14786 outcome cohort

Abbreviations

Adding the SNOMEDCT abbreviations to the vocabulary of correct spelled tokens was used to manage the abbreviations in the text. A spelling correction algorithm was applied, as adding abbreviations to the vocabulary only will not reduce the number of features. This will reduce the number of features seeing as some abbreviations were corrected to known abbreviations. For example, let us take unknown abbreviations for *paracetamol*, *pcm* and *pct*. After *pcm* is added to the vocabulary the spellings algorithms will correct *pct* to *pcm*, thereby reducing the number of features.

Token deletion

To reduce features, some tokens were deleted, after applying the spellings and abbreviation correction algorithms. The anonymized tokens in the text were deleted in all experiments. These tokens are names, places or other privacy sensitive information, which have been replaced with #NAME# or another appropriate token within #'s. Dates and other numbers are also deleted, for privacy reasons and to reduce feature dimensions.

Single character words in Dutch are extremely rare with u (you) and o, a, e (as an expressions of surprise) being the only words. Since these have no semantic meaning the single character words were deleted, these are either spelling mistakes, accidental keystrokes, or abbreviations (that have been expanded in the previous step). Tokens that appear less than one hundred times in the text data set are also deleted. These tokens can be considered as either spelling mistakes, or unknown and uncommon abbreviations, or uncommon words. In all cases the deletion of these tokens will reduce the number of features, without losing to much semantic value in the text. In a similar fashion tokens that occur in more than 60% of all the data cases, with an exception to age and sex, are also deleted. These tokens are so common in the text data that they can be considered either stop words or words with little semantic value to the prediction problem.

Text representations

The text representations that are used in this research are a bag-of-words representation on unigrams for the Machine Learning models and the pretrained MedRoberta.nl tokenizer, a byte-level Byte Pair Encoding (BPE) tokenizer, for the deep learning models.

Missing entries

There are data instances that do not have any associated text. Age and sex of the subject are added as text to the text field in all instances.

Structured data features

The structured data for the prediction models were extracted using the OHDSI Feature-Extraction R package [1]. The covariates that are extracted are a combination of all drugs or medicine that were prescribed, drug classes, condition, condition classes, procedures, observations, as well as demographic data such as age, sex. Covariates that occur in less than 0.1% and in 100% of the subject instances are removed. The observation window was chosen to be the same as reported in tables 2.2, 2.3, 2.3.

Models

Unstructured text data

The deep learning models that were trained and evaluated on the unstructured data were a 2-layer CNN model, a 2-layer BiLSTM model and MedRoberta.nl, all with 2 fully connected layers on top. These were trained and evaluated on the three prediction problems with several pre-processing and text representation settings. The hyper parameters for these models were picked and chosen by the author and can be found in the Appendix 6.

The traditional Machine Learning models that were evaluated on the unstructured data are a linear model trained with L1 prior as regularizer and a gradient boosting classifier. The hyper parameters for these models were picked and chosen by the author and can be found in the Appendix 6. The same evaluation metrics are used for these models as for the deep learning models. Therefore, the results of these experiments were part of the same table.

Structured data

The deep learning model that is trained on the structured data is a three layer fully connected network. The hyper parameters were picked and chosen by the author and can be found in the Appendix 6. The evaluation metrics that are used for the unstructured text data models were used to evaluate the performance of the structured data model as

[1] https://github.com/OHDSI/FeatureExtraction

well. The results are presented in a separated table with the best performing models on the unstructured data as repeated entries to ease comparisons.

The deep learning models were coded by making use of the pytorch and pytorch-lightning library [1]. The traditional Machine Learning models were coded by making use of the scikit-learn library [2].

A Titan X GPU, a Titan Xp GPU, both with 12 Gb of Ram and 56 CPUs were available for training the models. Due to sharing resources, all of this hardware was not available at all times, therefore the number of GPUs and CPUs were added as a parameter when running the experiments.

Evaluation

In all the experiments where models were trained the following approach is taken to get the most general representative evaluations: The experiments are run with at least 3-fold cross validation, and with at least three different seeds. This ensures that the chance of getting a "good" random seed is reduced and that the results are more generalisable.

Each model is evaluated using the following metrics: area under the receiver operating curve (AUROC), area under the precision recall curve (AUPR) and F1 score and Brier score. The F1 score is calculated at the risk threshold that gives the maximum F1 score for the model. For all metrics, with an exception the Brier score, a higher score indicates a greater performance. In addition, the calibration curves will be plotted. These metrics are acquired and calculated with a 95% confidence interval, from the cross validated test over all three seeds. To compare the best performing models and metrics over the different configurations the results were put in the same table as the for the pre-processing methods.

Due to a large class imbalance in all the prediction problem data cohorts, two methods were chosen to help train the models. Either weights were added to the loss function, or a weighted random sampler is used in the data loader. Both methods aim to help reduce the effect of the class imbalance when training the models. The weighted random sampler undersamples the majority class, while oversampling the minority class. The weighted loss function does not change the occurrence of the classes, it enacts a higher penalty loss for the minority class. For the deep learning models the cross-entropy loss function was used when training the models. Which for these prediction problems was the same as the binary cross-entropy loss function.

[1] https://pytorch.org/ https://www.pytorchlightning.ai/

[2] https://scikit-learn.org/stable/

In one table the results will be presented that aim to answer the first and second research question: How do different DNN architectures, such as RNNs, CNNs and attention models, compare in performance to traditional Machine Learning models, using unstructured clinical text data? What is the effect of pre-processing methods, such as embeddings, spelling corrections, abbreviations, on the performance of the DNN models? The results used to answer the third research question, how does the use of text data compare to the use of structured data, was presented in a separate table, while using the same evaluation metrics.

3

Results

In this chapter the results of the experiments are presented. The results are clustered per prediction problem. The tables show the evaluation metric scores for every model and pre-processing method configuration. A M indicates that a spellings correction and abbreviation algorithm has been performed, a D indicates that the token deletion pre-processing has been performed, a $M + D$ indicates that both have been performed. A S indicates that the model has been trained on structured data. The *italic* scores highlight the best performing metrics for that model over all pre-processing methods. The **bold** scores highlight the best performing metrics over all models.

Hospital readmission

In table 3.1 the results of the experiments for the hospital readmission within 32 days prediction problem can be found. The corresponding curves for AUROC, AUPRC and the calibration curves can be found in Figures 3.1, 3.2, 3.3 respectively.

The CNN architectures, with the combination of the pre-processing methods, performs on all evaluation metrics, except for the Brier score, the best of all models. The traditional machine learning methods have the best Brier score of all the models. The CNN and BiLSTM models outperform the traditional machine learning methods on the AUROC, AUPRC and F1 score. The MedRoBERTa.nl transformer model performs worse on all metrics compared to the traditional machine learning models.

For the deep neural networks, the pre-processing methods caused a performance gain on the AUROC and F1 metric, with the token deletion pre-processing method outperforming the spellings correction algorithm. The combination of the two pre-processing methods performs the best. The MedRoBERTa.nl transformer model does not experience the same gain in performance over the pre-processing methods regarding the AUPRC as

Table 3.1: Hospital readmission results

Hospital Readmission	Models	pre-processing methods	AUROC	AUPRC	F1score	Brier score
Deep Neural Networks	CNN		0.74 +- 0.04	0.23 +- 0.01	0.30 +- 0.02	0.17
		M	0.78 +- 0.03	0.26 +- 0.01	0.33 +- 0.03	0.17
		D	0.80 +- 0.03	0.32 +- 0.01	0.38 +- 0.02	*0.16*
		M + D	***0.87 +- 0.03***	***0.43 +- 0.01***	***0.47 +- 0.02***	*0.16*
	BiLSTM		0.70 +- 0.03	0.17 +- 0.01	0.26 +- 0.03	0.19
		M	0.72 +- 0.03	0.19 +- 0.01	0.28 +- 0.02	*0.18*
		D	0.72 +- 0.02	0.20 +- 0.01	0.28 +- 0.02	0.21
		M + D	*0.75 +- 0.02*	*0.21 +- 0.01*	*0.30 +- 0.02*	0.19
	MedRoBERTa.nl		0.62 +- 0.04	*0.13 +- 0.01*	0.20 +- 0.04	0.24
		M	0.63 +- 0.03	*0.13 +- 0.01*	0.20 +- 0.03	0.24
		D	0.63 +- 0.04	*0.13 +- 0.01*	*0.21 +- 0.03*	*0.23*
		M + D	*0.64 +- 0.02*	*0.13 +- 0.01*	*0.21 +- 0.02*	*0.23*
Machine Learning Models	LASSO		*0.68 +- 0.02*	*0.18 +- 0.01*	*0.24 +- 0.02*	***0.08***
		M	*0.68 +- 0.01*	*0.18 +- 0.01*	*0.24 +- 0.02*	***0.08***
		D	*0.68 +- 0.03*	*0.18+- 0.01*	*0.24 +- 0.02*	***0.08***
		M + D	*0.68 +- 0.01*	*0.18 +- 0.01*	*0.24 +- 0.02*	***0.08***
	Gradient Boosting Classifier		*0.69 +- 0.03*	0.19 +- 0.01	*0.25 +- 0.03*	***0.08***
		M	*0.69 +- 0.02*	*0.20 +- 0.01*	*0.25 +- 0.03*	***0.08***
		D	*0.69 +- 0.02*	0.19 +- 0.01	*0.25 +- 0.03*	***0.08***
		M + D	*0.69 +- 0.02*	*0.20 +- 0.01*	*0.25 +- 0.03*	***0.08***

Table 3.2: Comparison of using structured and unstructured data on the hospital readmission prediction problem

Hospital readmissions	Models	pre-processing methods	AUROC	AUPRC	F1	Brier score
Unstructured text data models	CNN	M + D	**0.87**	**0.43**	**0.47**	**0.16**
	BiLSTM	M + D	0.75	0.21	0.30	0.19
	MedRoBERTa.nl	M + D	0.64	0.13	0.21	0.23
Structured data	Fully Connected Deep Network	S	0.74	0.21	0.30	0.20

the BiLSTM and CNN. The traditional machine learning models do not experience any clear difference over the pre-processing methods. The traditional machine learning methods have performances that are comparable to each other. The deep learning models have performances that are dissimilar. The MedRoBERTa.nl transformer model performs worse over all metrics regarding the BiLSTM and CNN model. The CNN model outperforms the BiLSTM model on all metrics except for the Brier score.

In table 3.2 the results of the fully connected network trained on structured data can be found. The CNN outperforms the fully connected deep model. The performance of the BiLSTM is comparable to the fully connected model. The transformer model performs worse on all metrics.

Dementia

In table 3.3 the results of the experiments for the dementia within 5 years prediction problem can be found. The corresponding curves for AUROC, AUPRC and the calibration curves can be found in Figures 3.4, 3.5, 3.6 respectively.

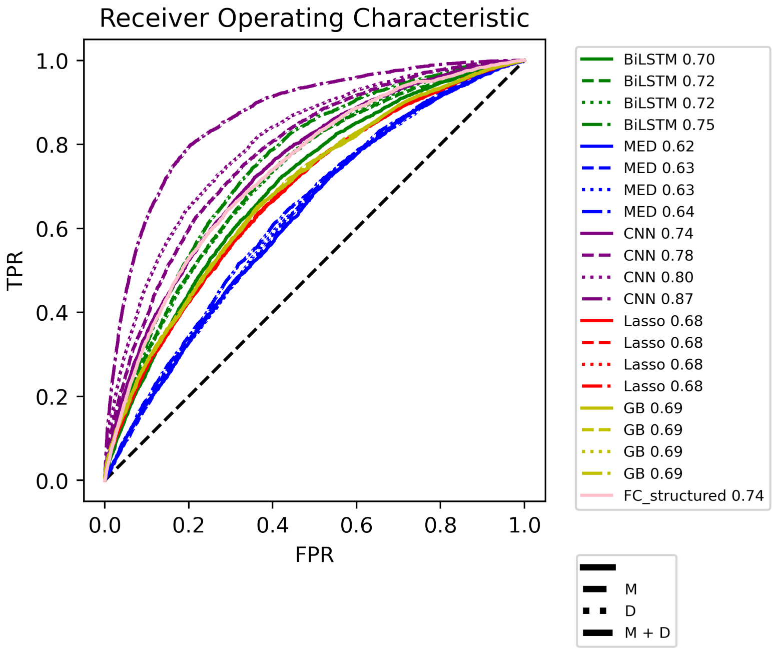

Figure 3.1: AUROC performance and curves for the hospital readmission prediction problem.

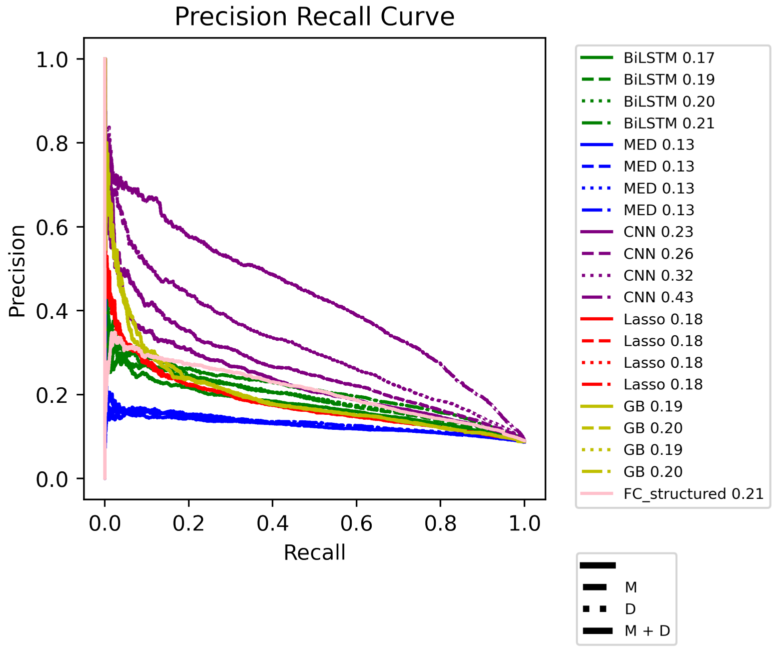

Figure 3.2: AUPRC performance and curves for the hospital readmission prediction problem.

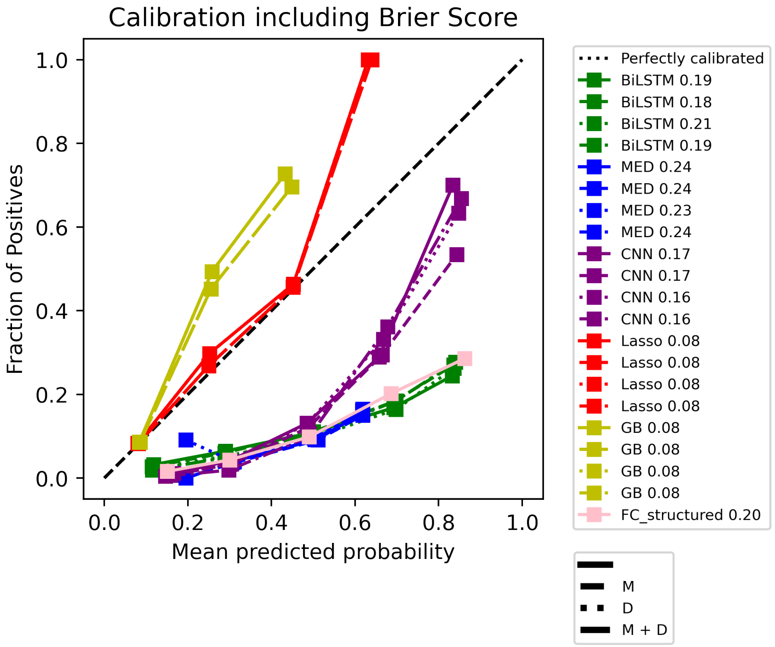

Figure 3.3: Brier score and calibration curves for the hospital readmission prediction problem.

The CNN architectures, with the combination of the pre-processing methods, performs on all evaluation metrics, except for the Brier score, the best of all models. The traditional machine learning methods have the best Brier score of all the models. The CNN and BiLSTM models outperform the traditional machine learning methods on the AUROC, AUPRC and F1 score. The MedRoBERTa.nl transformer model performs worse on all metrics compared to the traditional machine learning models.

For the deep neural networks, the pre-processing methods cause a performance gain on the AUROC and F1 metric, with the token deletion pre-processing method outperforming the spellings correction algorithm. The combination of the two pre-processing methods performs the best. The MedRoBERTa.nl transformer model does not experience the same gain in performance over the pre-processing methods regarding the AUROC and AUPRC as the BiLSTM and CNN. The traditional machine learning models do not experience any clear difference over the pre-processing methods. The traditional machine learning methods have performances that are comparable to each other. The deep learning models have performances that are dissimilar. The MedRoBERTa.nl transformer model performs worse over all metrics regarding the BiLSTM and CNN model. The CNN model outperforms the BiLSTM model on all metrics.

In table 3.4 the results of the fully connected network trained on structured data can

Table 3.3: Dementia results

Dementia	Models	pre-processing methods	AUROC	AUPRC	F1score	Brier score
Deep Neural Networks	CNN		0.93 +- 0.04	0.30 +- 0.01	0.38 +- 0.02	***0.02***
		M	***0.96 +- 0.03***	0.55 +- 0.57	0.57 +- 0.02	0.13
		D	***0.96 +- 0.03***	0.69 +- 0.01	0.70 +- 0.02	0.13
		M + D	***0.96 +- 0.03***	***0.83 +- 0.01***	***0.87 +- 0.01***	0.14
	BiLSTM		0.87 +- 0.03	0.11 +- 0.01	0.20 +- 0.07	0.22
		M	0.91 +- 0.03	0.19 +- 0.01	0.32 +- 0.07	0.15
		D	0.90 +- 0.02	0.17 +- 0.01	0.26 +- 0.02	0.07
		M + D	*0.93 +- 0.02*	*0.27 +- 0.01*	*0.44 +- 0.02*	*0.06*
	MedRoBERTa.nl		0.51 +- 0.04	*0.02 +- 0.01*	*0.04 +- 0.04*	*0.24*
		M	*0.54 +- 0.03*	*0.02 +- 0.01*	*0.04 +- 0.04*	*0.24*
		D	0.53 +- 0.04	*0.02 +- 0.01*	*0.04 +- 0.03*	0.25
		M + D	*0.54 +- 0.02*	*0.02 +- 0.01*	*0.04 +- 0.02*	0.25
Machine Learning Models	LASSO		*0.81 +- 0.02*	*0.06 +- 0.01*	*0.13 +- 0.02*	***0.02***
		M	*0.81 +- 0.01*	*0.06 +- 0.01*	*0.13 +- 0.01*	***0.02***
		D	*0.81 +- 0.03*	*0.06 +- 0.01*	*0.13 +- 0.03*	***0.02***
		M + D	*0.81 +- 0.01*	*0.06 +- 0.01*	*0.13 +- 0.01*	***0.02***
	Gradient Boosting Classifier		*0.81 +- 0.03*	*0.07 +- 0.01*	*0.13 +- 0.02*	***0.02***
		M	*0.81 +- 0.02*	*0.07 +- 0.01*	*0.13 +- 0.03*	***0.02***
		D	*0.81 +- 0.02*	*0.07 +- 0.01*	*0.13 +- 0.02*	***0.02***
		M + D	*0.81 +- 0.02*	*0.07 +- 0.01*	*0.13 +- 0.01*	***0.02***

Table 3.4: Comparison of using structured and unstructured data on the dementia within 5 years prediction problem

Dementia	Models	pre-processing methods	AUROC	AUPRC	F1	Brier score
Unstructured text data models	CNN	M + D	**0.96**	**0.83**	**0.87**	**0.14**
	BiLSTM	M + D	0.93	0.27	0.44	0.06
	MedRoBERTa.nl	M + D	0.54	0.02	0.04	0.25
Structured data	Fully Connected Deep Network	S	0.87	0.09	0.18	0.13

be found. The CNN and BiLSTM outperform the fully connected deep model. The performance of the MedRoBERTa.nl is worse than the fully connected model on all metrics.

End of Life Conversations

In table 3.5 the results of the experiments for the end of life conversations within 1 year prediction problem can be found. The corresponding curves for AUROC, AUPRC and the calibration curves can be found in Figures 3.7, 3.8, 3.9 respectively.

The CNN architectures, with the combination of the pre-processing methods, performs on all evaluation metrics the best of all models. The traditional machine learning methods share the best Brier score of all the models. The CNN and BiLSTM models outperform the traditional machine learning methods on the AUROC, AUPRC and F1 score. The MedRoBERTa.nl transformer model performs worse on all metrics compared to the traditional machine learning models.

For the deep neural networks, the pre-processing methods cause a performance gain on the AUROC, AUPRC and F1 metric, with the token deletion pre-processing method out-

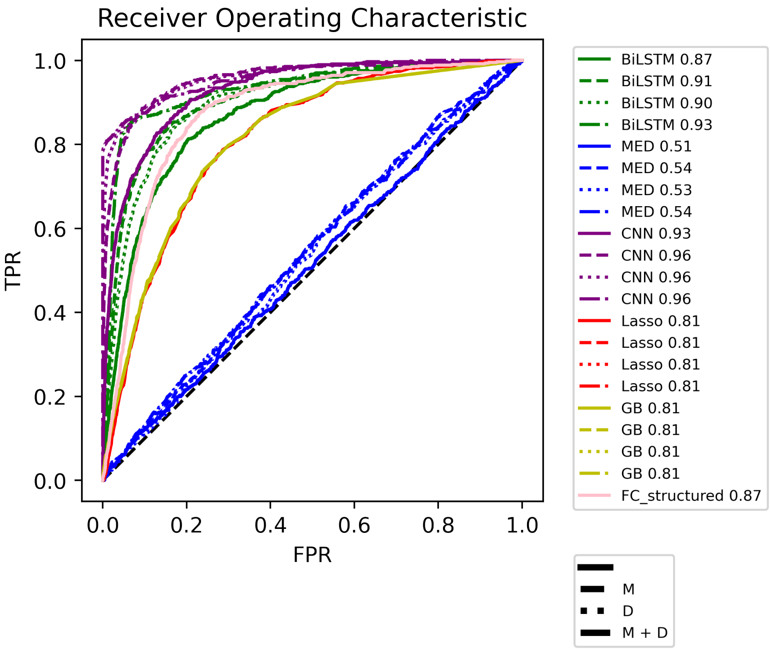

Figure 3.4: AUROC performance and curves for the dementia prediction problem.

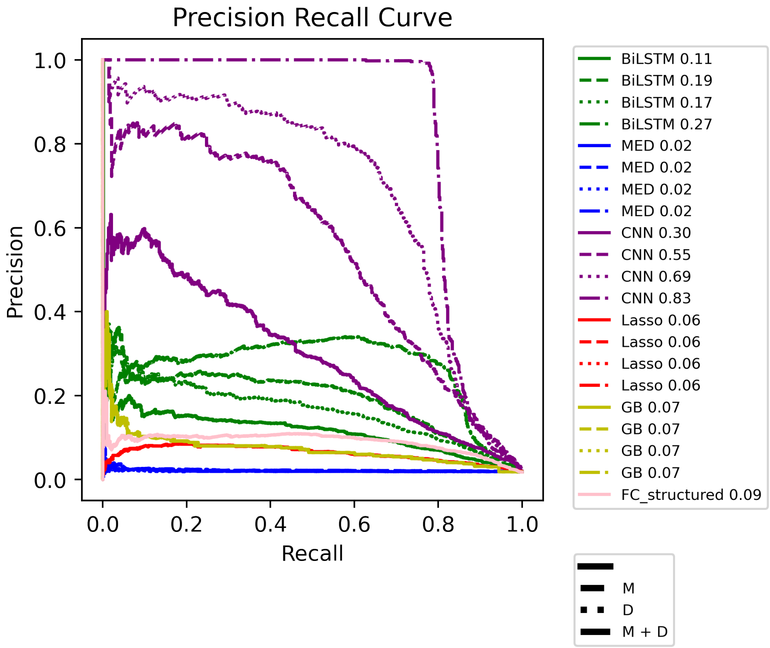

Figure 3.5: AUPRC performance and curves for the dementia prediction problem.

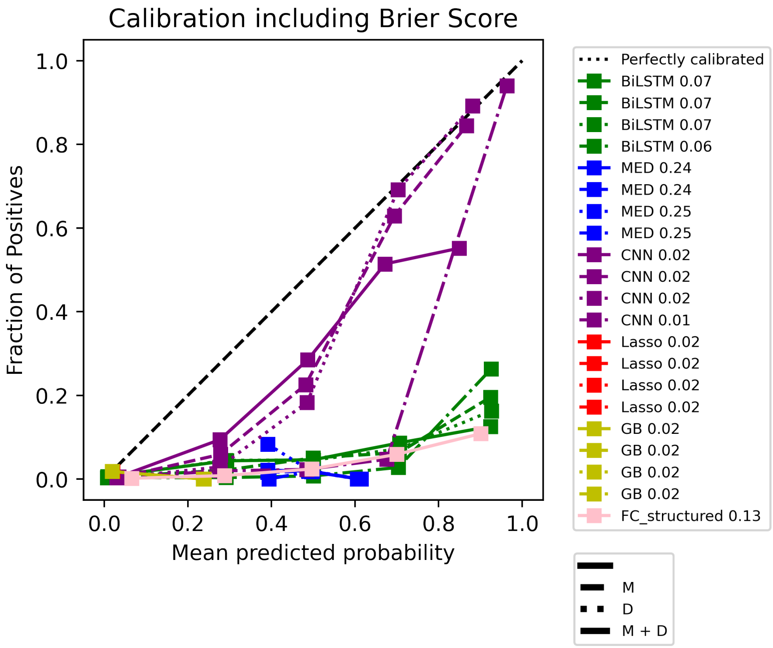

Figure 3.6: Brier score and calibration curves for the dementia problem.

performing the spellings correction algorithm. The combination of the two pre-processing methods performs the best. The traditional machine learning models do not experience any clear difference over the pre-processing methods. The traditional machine learning methods have performances that are comparable to each other. The deep learning models have performances that are dissimilar. The MedRoBERTa.nl transformer model performs worse over all metrics regarding the BiLSTM and CNN model. The CNN model outperforms the BiLSTM model on all metrics.

In table 3.6 the results of the fully connected network trained on structured data can be found. The CNN and BiLSTM outperform the fully connected deep model. The performance of the fully connected model is higher on all metrics compared to the transformer model.

Table 3.5: End of Life Conversations results

End of Life Conversations	Models	pre-processing methods	AUROC	AUPRC	F1score	Brier score
Deep Neural Networks	CNN		0.91 +- 0.02	0.15 +- 0.01	0.25 +- 0.01	***0.01***
		M	0.93 +- 0.01	0.24 +- 0.01	0.37 +- 0.02	***0.01***
		D	***0.95 +- 0.01***	0.60 +- 0.01	0.61 +- 0.01	***0.01***
		M + D	***0.95 +- 0.01***	***0.83 +- 0.01***	***0.88 +- 0.01***	***0.01***
	BiLSTM		0.81 +- 0.01	0.08 +- 0.01	0.17 +- 0.02	0.03
		M	0.85 +- 0.02	0.12 +- 0.01	0.24 +- 0.02	0.03
		D	0.88 +- 0.01	0.13 +- 0.01	0.24 +- 0.01	*0.04*
		M + D	*0.93 +- 0.01*	*0.25 +- 0.01*	*0.43 +- 0.02*	*0.04*
	MedRoBERTa.nl		0.65 +- 0.02	*0.02 +- 0.01*	0.04 +- 0.02	0.21
		M	0.65 +- 0.01	*0.02 +- 0.01*	0.04 +- 0.02	*0.22*
		D	*0.66 +- 0.02*	*0.02 +- 0.01*	*0.05 +- 0.01*	*0.22*
		M + D	*0.66 +- 0.01*	*0.02 +- 0.01*	*0.05 +- 0.01*	*0.22*
Machine Learning Models	LASSO		*0.80 +- 0.02*	*0.04 +- 0.01*	*0.08 +- 0.02*	***0.01***
		M	*0.80 +- 0.01*	*0.04 +- 0.01*	*0.08 +- 0.01*	***0.01***
		D	*0.80 +- 0.02*	*0.04 +- 0.01*	*0.08 +- 0.01*	***0.01***
		M + D	*0.80 +- 0.01*	*0.04 +- 0.01*	*0.08 +- 0.01*	***0.01***
	Gradient Boosting Classifier		*0.80 +- 0.02*	*0.04 +- 0.01*	*0.08 +- 0.01*	***0.01***
		M	*0.80 +- 0.01*	*0.04 +- 0.01*	*0.08 +- 0.01*	***0.01***
		D	*0.80 +- 0.02*	*0.04 +- 0.01*	*0.08 +- 0.01*	***0.01***
		M + D	*0.80 +- 0.01*	*0.04 +- 0.01*	*0.08 +- 0.01*	***0.01***

Table 3.6: Comparison of using structured and unstructured data on the end of life conversations prediction problem

End of Life Conversations	Models	pre-processing methods	AUROC	AUPRC	F1	Brier score
Unstructured text data models	CNN	M + D	**0.95**	**0.83**	**0.88**	**0.01**
	BiLSTM	M + D	0.93	0.25	0.43	0.04
	MedRoBERTa.nl	M + D	0.66	0.02	0.05	0.22
Structured data	Fully Connected Deep Network	S	0.87	0.05	0.13	0.13

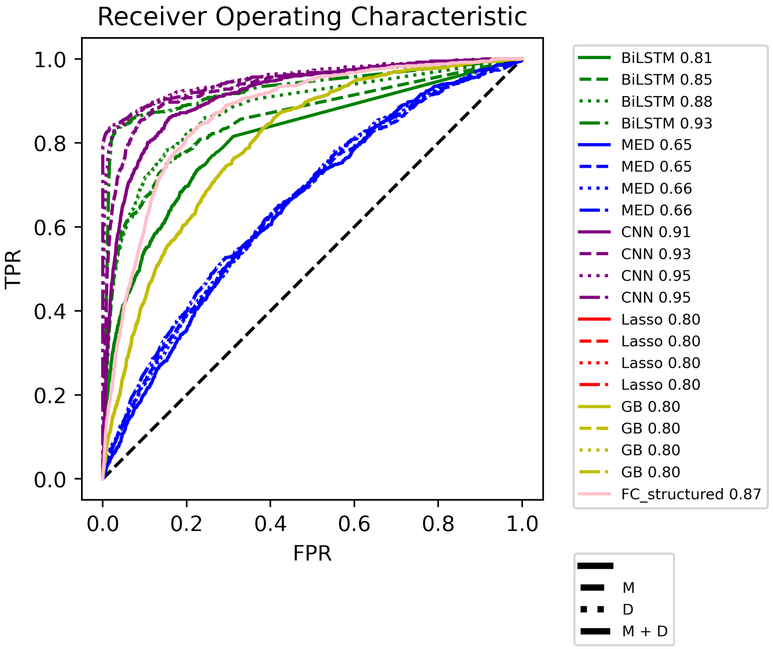

Figure 3.7: AUROC performance and curves for the end of life converstations prediction problem.

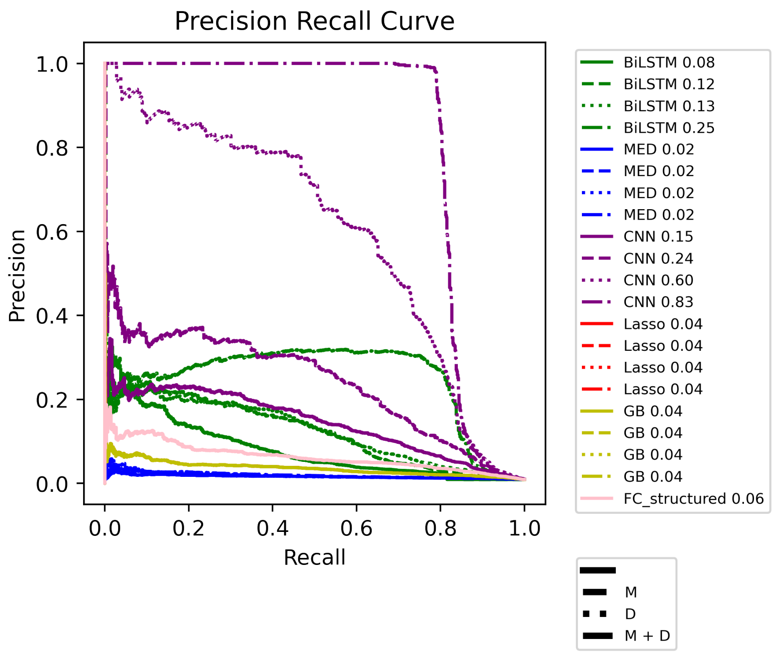

Figure 3.8: AUPRC performance and curves for the end of life converstations prediction problem.

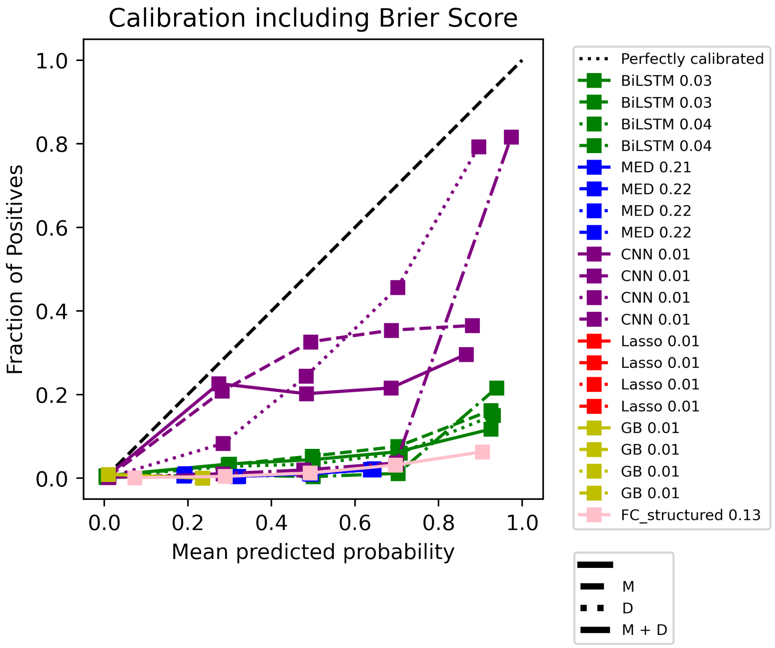

Figure 3.9: Brier score and calibration curves for the end of life conversations problem.

4

Discussion

In this chapter the results and setup of this research is put into context. Implications of the results for current and further research are analysed. The discussion starts with the data that was used in this thesis, followed by the definitions and selection of the population cohorts. Furthermore, the text pre-processing methods that were tested and the resulting text representations are analysed. Lastly, the models that were trained and the evaluation metrics that were used for the training and testing are discussed.

IPCI Data

The IPCI dataset consists of a large part of the Dutch population. The size of this database reduces the chances of having non representative data for the whole Dutch population. Quality control steps (5) are conducted before new releases of the database. However, mapping errors can still persist when the data is mapped from the IPCI vocabulary (ICPC-1) to the OMOP-CDM vocabulary. The OMOP-CDM does enable other observational health care dataset to be used for external validation. Further research could replicate the experiments on EHR data sets from other sources to evaluate the findings of this research.

Population Cohorts

Excluding subjects that died during the time at risk, influences the observed risk in the dataset. As an example: the readmission prediction problem aims to predict if a subject will be readmitted within 32 days. Subjects that pass away during the 32 days can be subjects that died with a low readmitted chance or with a high readmitted chance. However, excluding the subjects that passed away was necessary to be compliant with previous research. The inclusion and exclusion criteria for the prediction problems presented in this research were not made by a practicing physician and no manual inspection of the recall and precision of these criteria was performed. In addition, the text data was selected after defining the population cohorts in ATLAS. When a data entry, such as a condition or diagnosis, was miscoded by a GP, incorrect labels could have been introduced in the training data. This is a risk that occurs when working with any type of observational database. The recommendation for current and future research is to include a practicing physician for evaluation on the inclusion and exclusion criteria, with an extra focus on the effect of subjects that passed away in the time at risk period.

Text pre-processing methods

As noted before the IPCI data contains many spelling mistakes, abbreviations, and unfinished sentences. There does not exist a cleaned golden standard (sub-)set of this dataset. This made assessment of the spelling correction algorithms and abbreviation handling methods not possible. Instead, a common approach was taken that has been done before (8). Research into other correction methods could be beneficial to the training speed and accuracy of the deep learning models. In addition, cleaning up the text data of the IPCI dataset could influence assessing if the GP coded correctly.

The embedding method for the traditional machine learning models was a unigram bag-of-words. This method has been used in previous literature and in the research department. For the deep learning models, the MedRoBERTa.nl tokenizer was used to tokenize the text. The benefit of this tokenizer is the ability to tokenize unseen or unknown words and the tokenizer has been trained on Dutch clinical text. Future research can assess if other embedding and tokenizing methods will improve the performance of the models.

In earlier stages of this research tokens that occurred in more than 80% or less than 1% of the data instances was deleted, instead of the current 60% and less than 100 occurrences in general. The results with the previous token deletion configuration were different than with the current configuration. The major increase in performance when using text pre-processing methods was not observed and the CNN and BiLSTM only slightly outperformed the machine learning methods. Therefore, more research into the pre-processing methods is suggested, with a focus on the deletion of tokens.

Models

Hyper parameters

In this research the models were manually evaluated and adjusted before the final experiments were run. An improvement that can be made is enabling more resources to do a more sophisticated hyper parameter configuration and architecture configuration.

Specially created models

Finding the best performing architecture for one prediction problem and discovering a new one for the next prediction problem was not the objective of this thesis. The chosen architectures are not necessarily the best performing architectures for one of the prediction problems. By choosing the same deep learning architecture for all three prediction problems there can be a discrepancy regarding performance with a highly specialised and trained deep learning model on a single prediction problem. It would be interesting to see the extent of this discrepancy between the "common" deep learning models in this research and the "specialised" models for one prediction problem.

Disparity in DNN performance

The performance between the deep learning models was not comparable to the unified performance between the traditional machine learning models. The chosen transformer model performed remarkably worse than the other two deep learning models.

The MedRoBERTa.nl model was pre-trained and is significantly larger and deeper than the other two models. Due to limited resources the choice was made to only train the classification layers on top of the transformer model, in contrast to the other two models where all layers were trained. Due to the large epoch training times, in comparison to the other models, the transformer model was also trained for less epochs than the other models. Transformer models are also known to be sensitive to hyper parameter settings, but due to limited resources a full hyper parameter search and optimization could not be performed. All these choices could have made a significant impact on the performance of the transformer model. It is therefore suggested that future research will dedicate more resources on training transformer models.

The high performance of the CNN architecture in comparison with the other two deep learning architectures could indicate that there was not enough data to train a BiLSTM and transformer model. Due to the high class imbalance, there were only a couple of thousand positive data instances available. The complexity of the chosen models could have been too high to properly train on these data cohorts.

Literature

The DNN performance presented in previous reported literature were trained and evaluated on different prediction problems, and primarily on non-Dutch text. The text preprocessing methods that boosted the DNN performance in this research were also not found in literature on clinical prediction models. Therefore, a direct comparison with the absolute performance of previous literature is difficult. However, several remarks and comparisons can be made. The transformer model did not reach the best performance of all models, which is different from previous reported results in literature. In Huang, et al. (35) the AUROC performance of ClincicalBERT transformer model outperformed the tested BiLSTM model and the tested BERT transformer model matched the performance of the BiLSTM. In this research the transformer model did not match the performance of the BiLSTM model in any of of the prediction problems. In Mohamaddi, et al. (37) the BERT transformer model did perform 0.10 AUROC points worse than the best deep learning model, which is comparable to performance of the MedRoBERTa.nl model in this research compared to the traditional machine learning models. In Zhang, et al. (23) the RNN matched the performance of a CNN on predicting hospital readmission with 30 days. Both architectures performed 0.1 AUROC point worse than the corresponding architectures in this research on the hospital readmission prediction problem. This difference can be caused by a difference in observation window, text representation or architecture and hyper parameter choice.

In this research a DNN outperformed the traditional machine learning models on all prediction problems. Menger, et al. (43) and Zhang, et al. (23) report a similar conclusion, albeit that the performance gap is smaller. More research on a variety of prediction problems is recommended to assess if DNN outperforming traditional machine learning is structural.

Overconfidence

In the Appendix 6 three graphs can be found of the AUROC for a CNN model trained on an earlier version of the dementia cohort. From left to right the epochs increase and interestingly the training and validation loss decreased. In the experiment an observation window from 1 month was chosen, instead of the eventually used 1 year. This resulted in some data instances not have any associated text notes, due to subjects not having any entries in that month. The training and validation loss indicated that the CNN model was training well, but when evaluating the model on several epochs the AUROC was decreasing.

The model was checked for overfitting, which did not occur. This experiment indicates a difference in loss function and evaluation metrics. The model was starting to mis-classify a lot of positive labels, hence the reduced AUROC over time, but it was getting more accurate regarding predicting a 0 risk for the negative label. Hence the model was getting overconfident that the predicted risk should be 0 at all times. This is likely a result of a high class imbalance and having no text to predict on. Future research could look at mitigating the problems that arise when only a small observation window, with a lot of missing entries, is chosen for prediction problems.

Evaluation metrics

The area under the receiver operating characteristic and the area under the precision recall curve are the most common evaluation metrics for prediction models in the clinical domain. Using these evaluation metrics in this research provided two challenges. Firstly, when using more than one evaluation metric, making statements such as this model is better than that model, is problematic. Secondly these evaluation metrics are not sufficient to make strong claims about the model performance from a practical perspective. The whole receiver operating characteristic curve is not relevant for a practical usage, a high false positive rate configuration will not be used. Adding calibration metrics, as done in this research, evaluates the resemblance of the predicted scores to the observed risk, adds another layer of complexity when evaluating the models.

An exploratory step to use only one evaluation metric has been made in this research, using the area under a F1-risk threshold curve. These curves can be found in the Appendix 6. This evaluation metric could not be fully tested in this research, but some interesting notes can be made. The highest point of the curve provides a method to configure the class labels on the prediction score. The location of the highest point gives information regarding the calibration of the model. Whereas the width of the curve around the highest point gives an indication about the discriminatory effect of the model. Research into evaluation metrics for clinical prediction models, with a strong link to a practical implementation of the model would be beneficial for the whole field.

Ensemble models

In this research, structured data and unstructured text data were compared, to assess the importance of the unstructured text data. The conclusion of the experiment shows that unstructured text data has useful information for prediction models. Next to comparing unstructured and structured data, Zhang, et al. (23) combined structured and unstructured data for deep learning models. Future research could build on this by experimenting with ensemble methods for deep learning networks.

Explainability

Explainable AI is a field that had an increased amount of interest in the past years (44, 45, 46, 47, 48, 49, 50). Exploring the usage of explainable AI for clinical prediction models is an interesting direction for the medical informatics field to consider. Research into explainability could help identify the cause for the high performance of the CNN regarding the other models in this research. It can also be used as a measure to evaluate the validity of training the model. For example, explainable AI can help identify if the models have a unwanted bias for certain ethnic groups. In addition, explainable AI should be considered in the context of using deep learning in practice. The ability to give insight into the prediction could help a patient and GP with using a deep learning model. It is therefore suggested that the role and possibilities for explainable AI in medical informatics is investigated.

5

Conclusion

In this section the results and other contributions of this research thesis are summarized. This section will start with answering the three research questions.

1. How do different DNN architectures, such as RNNs, CNNs and attention models, compare in performance to traditional Machine Learning models, using unstructured clinical text data?
2. What is the effect of different pre-processing methods, such as embeddings, spelling corrections, abbreviations, on the performance of the Artificial Intelligencemodels?
3. How does the use of text data compare to the use of structured data?

Subsequently the possible implications of this research for future research, especially for the OHDSI community are explored.

Research Question 1

In all three prediction problems there is a dissimilarity regarding the performance between the deep learning models. The transformer model performs worse on all metrics compared to the BiLSTM, CNN and traditional Machine Learning models. The BiLSTM and CNN both outperform the traditional models, but there is a disparity between the performance of these two deep learning models. The CNN performs the best of all models for each prediction problem, except when looking at calibration. Due to the high variance between the deep learning models performance, it is clear that the performance of the DNN is highly dependent on the architecture and other settings. In this research they did outperform the traditional Machine Learning models when trained on unstructured clinical data. The CNN model and BiLSTM do outperform the Machine Learning models on most evaluation metrics, but never on the Brier score. When comparing the calibration curves however it is clear that both deep learning models are better calibrated over the whole predicted probability spectrum than the traditional models. As a result this research has shown that deep neural networks, CNNs and RNNs, can outperform traditional Machine Learning models.

Research Question 2

The pre-processing methods have a mixed effect on the Artificial Intelligencemodels. The traditional Machine Learning models do not clearly benefit on any of the prediction problems from the pre-processing methods. The three deep neural networks do have an increased or matching performance on all evaluation metrics when the pre-processing methods were performed. The CNN model, trained on the combination of both pre-processing methods, experienced major performance gains regarding AUROC, AUPRC and F1 score for all prediction problems. The best performance was experienced when pre-processing methods were combined, the second best performance when the token deletion was performed and the third best when the spellings correction algorithm with the addition of abbreviations was used.

Text pre-processing methods have a positive effect on the performance of deep learning models, but do not influence the performance of traditional Machine Learning models, for patient level prediction methods.

Research Question 3

The fully connected model, trained on structured data, did not perform better than all the deep learning models that were trained on unstructured text. Instead, the CNN outperformed the fully connected model on all prediction problems and the BiLSTM had a comparable or better performance. This indicates that the unstructured text data has valuable information, which makes the data useful for prediction models.

Objective

The objective of the research thesis was to assess the potential benefit of using Deep Neural Networks for patient level prediction problems using text. The implementation of the data processing and experiments was made with the idea of reproducibility for the OHDSI community. The OMOP-CDM and already existing OHDSI analytics tools were used to define and generate the patient cohorts. Even though the models were made and trained in Python and not using the OHDSI R package, the code was written with the possibility to add a Python to R interface. The results of this thesis show there is a benefit of using DNN on text in the patient prediction models. More research is needed to determine which performance is needed for these models to be used in practise and what methods should be implemented to obtain these results. This can be evaluated and further researched by the OHDSI community, by expanding the current deep learning models in the patient level prediction package [1], to use the text data from EHRs.

[1] https://github.com/OHDSI/DeepPatientLevelPrediction

References

[1] Jenna Reps, Martijn Schuemie, Marc Suchard, Patrick Ryan, and Peter Rijnbeek. **Design and Implementation of a Standardized Framework to Generate and Evaluate Patient-Level Prediction Models Using Observational Healthcare Data**. **25**(8):969–975. 1

[2] Benjamin A. Goldstein, Ann Marie Navar, Michael J. Pencina, and John P.A. Ioannidis. **Opportunities and Challenges in Developing Risk Prediction Models with Electronic Health Records Data: A Systematic Review**. **24**(1):198–208. 1, 5, 7

[3] Elizabeth Ford, John A Carroll, Helen E Smith, Donia Scott, and Jackie A Cassell. **Extracting Information from the Text of Electronic Medical Records to Improve Case Detection: A Systematic Review**. **23**(5):1007–1015. 1, 7

[4] Christopher D. Manning and Hinrich. Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press. 3

[5] Maria A J de Ridder, Marcel de Wilde, Christina de Ben, Armando R Leyba, Bartholomeus M T Mosseveld, Katia M C Verhamme, Johan van der Lei, and Peter R Rijnbeek. **Data Resource Profile: The Integrated Primary Care Information (IPCI) Database, The Netherlands**. page dyac026. 3, 29

[6] J. Martijn Nobel, Sander Puts, Frans C. H Bakers, Simon G. F Robben, and Ane L. A. J Dekker. **Natural Language Processing in Dutch Free Text Radiology Reports: Challenges in a Small Language Area Staging Pulmonary Oncology**. **33**(4):1002–1008. 3

[7] Shreyasi Pathak, Jorit van Rossen, Onno Vijlbrief, Jeroen Geerdink, Christin Seifert, and Maurice van Keulen. **Automatic Structuring of Breast Cancer Radiology Reports for Quality Assurance**. In *ICDMW*, **2018-**, pages 732–739. IEEE. 3

[8] Simone A. Cammel, Marit S. De Vos, Daphne van Soest, Kristina M. Hettne, Fred Boer, Ewout W. Steyerberg, and Hileen Boosman. **How to Automatically Turn Patient Experience Free-Text Responses into Actionable Insights: A Natural Language Programming (NLP) Approach**. **20**(1):97. 3, 5, 6, 30

[9] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall Series in Artificial Intelligence. Prentice Hall/Pearson Education, 2nd ed edition. 3

[10] Stephen Wu, Kirk Roberts, Surabhi Datta, Jingcheng Du, Zongcheng Ji, Yuqi Si, Sarvesh Soni, Qiong Wang, Qiang Wei, Yang Xiang, Bo Zhao, and Hua Xu. **Deep Learning in Clinical Natural Language Processing: A Methodical Review**. **27**(3):457–470. 3, 7, 8

[11] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. **Efficient Estimation of Word Representations in Vector Space**. 3

[12] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. **GloVe: Global Vectors for Word Representation**. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543. 3

[13] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. **Enriching Word Vectors with Subword Information**. **5**:135–146. 3

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. **BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding**. 3, 8

[15] Peter Norvig. **How to Write a Spelling Correction**. 5

[16] J. Words! **Dictionaries.** 5

[17] K. A. Spackman, K. E. Campbell, and R. A. Côté. **SNOMED RT: A Reference Terminology for Health Care**. pages 640–644. 5, 6

[18] Wolf. **Symspell vs. Bk-tree: 100x Faster Fuzzy String Search &amp; Spell Checking**. 6

[19] Srinidhi Karthikeyan, Alba G. Seco de Herrera, Faiyaz Doctor, and Asim Mirza. **An OCR Post-correction Approach Using Deep Learning for Processing Medical Reports**. pages 1–1. 6

[20] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. **RoBERTa: A Robustly Optimized BERT Pretraining Approach**. **abs/1907.11692**. 6

[21] Willie Boag, Dustin Doss, Tristan Naumann, and Peter Szolovits. **What's in a Note? Unpacking Predictive Value in Clinical Note Representations**. **2017**:26–34. 7

[22] Aurélie Névéol, Hercules Dalianis, Sumithra Velupillai, Guergana Savova, and Pierre Zweigenbaum. **Clinical Natural Language Processing in Languages Other than English: Opportunities and Challenges**. **9**(1):12–12. 7

[23] Dongdong Zhang, Changchang Yin, Jucheng Zeng, Xiaohui Yuan, and Ping Zhang. **Combining Structured and Unstructured Data for Predictive Models: A Deep Learning Approach**. **20**(1). 7, 12, 32, 34

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. **Attention is All you Need**. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, **30**. Curran Associates, Inc., 2017. 8

[25] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. **RoBERTa: A Robustly Optimized BERT Pretraining Approach**. *CoRR*, **abs/1907.11692**, 2019. 8

[26] Wietse de Vries, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan van Noord, and Malvina Nissim. **BERTje: A Dutch BERT Model**. *CoRR*, **abs/1912.09582**, 2019. 8

[27] Pieter Delobelle, Thomas Winters, and Bettina Berendt. **RobBERT: A Dutch RoBERTa-based Language Model**. **abs/2001.06286**. 8, 9

[28] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. **BioBERT: A Pre-Trained Biomedical Language Representation Model for Biomedical Text Mining**. 8

[29] Iz Beltagy, Kyle Lo, and Arman Cohan. **SciBERT: A Pretrained Language Model for Scientific Text**. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620. Association for Computational Linguistics. 8

[30] Stella Verkijk and Piek Vossen. **MedRoBERTa.Nl: A Language Model for Dutch Electronic Health Records**. **11**:141–159. 8

[31] Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. **Domain-Specific Language Model Pretraining for Biomedical Natural Language Processing**. **3**(1):1–23. 9

[32] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. **Don't Stop Pretraining: Adapt Language Models to Domains and Tasks**. pages 8342–8360. Association for Computational Linguistics. 9

[33] Egoitz Laparra, Aurelie Mascio, Sumithra Velupillai, and Timothy Miller. **A Review of Recent Work in Transfer Learning and Domain Adaptation for Natural Language Processing of Electronic Health Records**. **30**(1):239–244. 9

[34] Vin Sachidananda, Jason Kessler, and Yi-An Lai. **Efficient Domain Adaptation of Language Models via Adaptive Tokenization**. pages 155–165. Association for Computational Linguistics. 9

[35] Kexin Huang, Jaan Altosaar, and Rajesh Ranganath. **ClinicalBERT: Modeling Clinical Notes and Predicting Hospital Readmission**. 9, 12, 32

[36] Tom M Seinen, Egill A Fridgeirsson, Solomon Ioannou, Daniel Jeannetot, Luis H John, Jan A Kors, Aniek F Markus, Victor Pera, Alexandros Rekkas, Ross D Williams, Cynthia Yang, Erik M van Mulligen, and Peter R Rijnbeek. **Use of Unstructured Text in Prognostic Clinical Prediction Models: A Systematic Review**. **29**(7):1292–1302. 9

[37] Ramin Mohammadi, Sarthak Jain, Amir T. Namin, Melissa Scholem Heller, Ramya Palacholla, Sagar Kamarthi, and Byron Wallace. **Predicting Unplanned Readmissions Following a Hip or Knee Arthroplasty: Retrospective Observational Study.** **8**(11):e19761. 9, 32

[38] Bahman Tahayori, Noushin Chini-Foroush, and Hamed Akhlaghi. **Advanced Natural Language Processing Technique to Predict Patient Disposition Based on Emergency Triage Notes**. **33**(3):480–484. 9

[39] Yuqi Si and Kirk Roberts. **Deep Patient Representation of Clinical Notes via Multi-Task Learning for Mortality Prediction**. **2019**:779–788. 9

[40] Gokul S. Krishnan and S. Sowmya Kamath. **A Supervised Learning Approach for ICU Mortality Prediction Based on Unstructured Electrocardiogram Text Reports**. In Max Silberztein, Faten Atigui, Elena Kornyshova, Elisabeth Métais, and Farid Meziane, editors, *Natural Language Processing and Information Systems*, pages 126–134. Springer International Publishing. 9

[41] Paulina Grnarova, Florian Schmidt, Stephanie Hyland, and Carsten Eickhoff. **Neural Document Embeddings for Intensive Care Patient Mortality Prediction**. 9

[42] Jihad S. Obeid, Jennifer Dahne, Sean Christensen, Samuel Howard, Tami Crawford, Lewis J. Frey, Tracy Stecker, and Brian E. Bunnell. **Identifying and Predicting Intentional Self-Harm in Electronic Health Record Clinical Notes: Deep Learning Approach.** **8**(7):e17784. 9

[43] Vincent Menger, Floor Scheepers, and Marco Spruit. **Comparing Deep Learning and Classical Machine Learning Approaches for Predicting Inpatient Violence Incidents from Clinical Text**. **8**(6). 9, 32

[44] Di Jin, Elena Sergeeva, Wei-Hung Weng, Geeticka Chauhan, and Peter Szolovits. **Explainable Deep Learning in Healthcare: A Methodological Survey from an Attribution View**. 34

[45] Lorenzo Malandri, Fabio Mercorio, Mario Mezzanzanica, Navid Nobani, and Andrea Seveso. **ContrXT: Generating Contrastive Explanations from Any Text Classifier**. **81**:103–115. 34

[46] Yong-Yeon Jo, Joon-myoung Kwon, Ki-Hyun Jeon, Yong-Hyeon Cho, Jae-Hyun Shin, Yoon-Ji Lee, Min-Seung Jung, Jang-Hyeon Ban, Kyung-Hee Kim, Soo Youn Lee, Jinsik Park, and Byung-Hee Oh. **Detection and Classification of Arrhythmia Using an Explainable Deep Learning Model**. **67**:124–132. 34

[47] Aniek F. Markus, Jan A. Kors, and Peter R. Rijnbeek. **The Role of Explainability in Creating Trustworthy Artificial Intelligence for Health Care: A Comprehensive Survey of the Terminology, Design Choices, and Evaluation Strategies**. **113**:103655. 34

[48] Erico Tjoa and Cuntai Guan. **A Survey on Explainable Artificial Intelligence (XAI): Toward Medical XAI**. **32**(11):4793–4813. 34

[49] Mateusz Szczepański, Marek Pawlicki, Rafał Kozik, and Michał Choraś. **New Explainability Method for BERT-based Model in Fake News Detection**. **11**(1):23705–23705. 34

[50] M. Beatrice Fazi. **Beyond Human: Deep Learning, Explainability and Representation**. **38**(7-8):55–77. 34

6

Appendix

F1-risk threshold curves

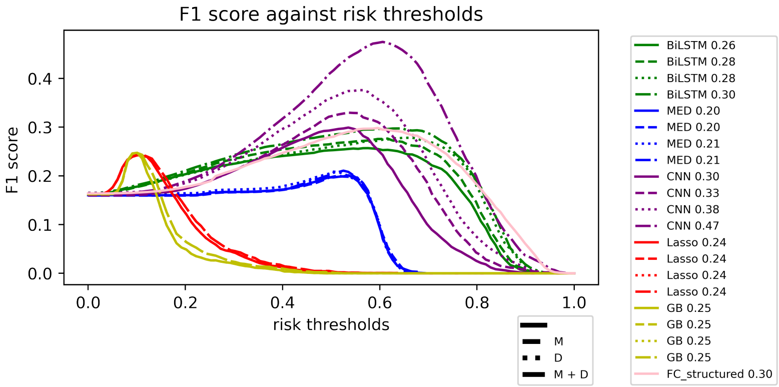

Figure 6.1: Maximum F1 performance and F1-riskthreshold curves for the hospital readmission prediction problem.

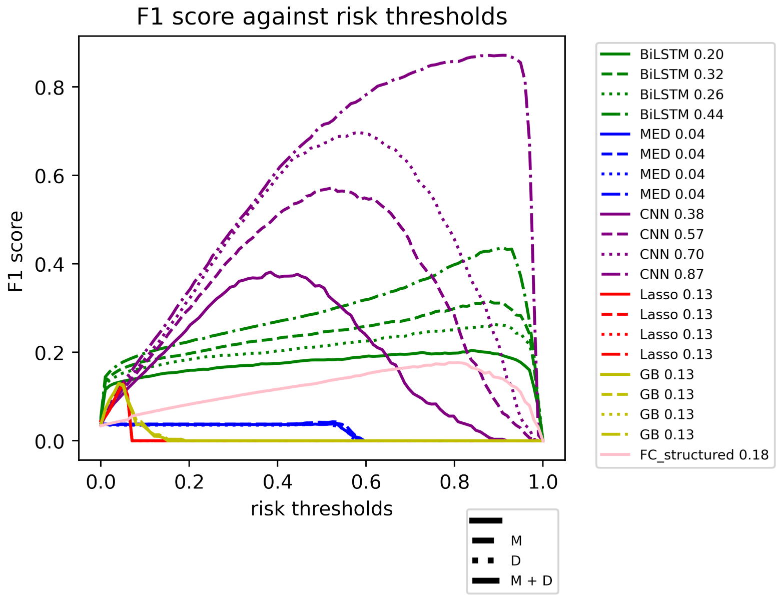

Figure 6.2: Maximum F1 performance and F1-riskthreshold curves for the dementia prediction problem.

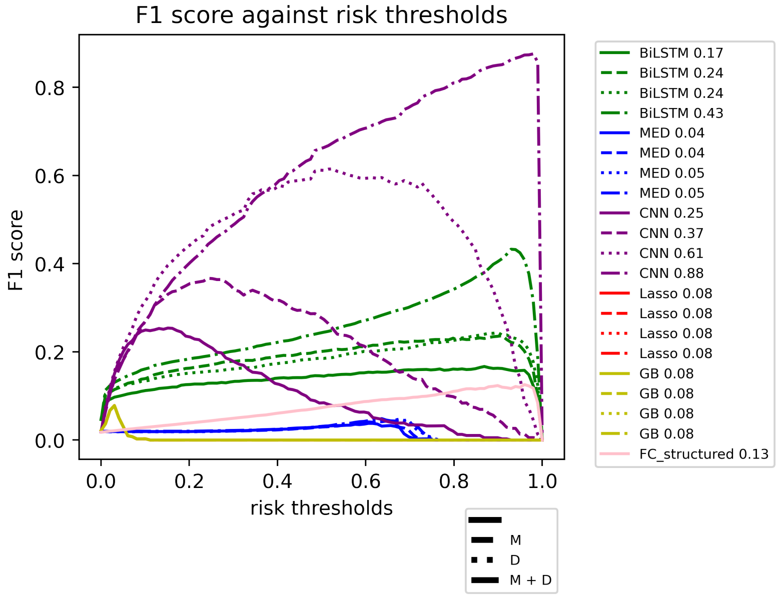

Figure 6.3: Maximum F1 performance and F1-riskthreshold curves for the end of life conversations prediction problem.

HyperParameters

BiLSTM

Hospital Readmission

```
adam_beta1: 0.9
adam_beta2: 0.999
dropout: 0.3
hidden_size: 128
input_size: 512
learning_rate: 1.0e-06
model_params:
  batch_size: 512
  dataloader_num_workers: 4
  max_seq_len: 512
  preprocessing_num_workers: 16
  use_sampler: true
num_layers: 2
optimizer_params:
  beta1: 0.9
  beta2: 0.999
  gpus: -1
  learning_rate: 1.0e-06
  lr_decay_factor: 0.999975
  lr_decay_step: 1
  lr_minimum: 0.0
  momentum: 0.0
  warmup: 2000
  weight_decay: 0.0
trainable: true
vocab_size: 52000
weights:
- 1
- 1
```

Dementia

```
adam_beta1: 0.9
adam_beta2: 0.999
dropout: 0.3
hidden_size: 128
input_size: 512
learning_rate: 1.0e-06
model_params:
  batch_size: 512
  dataloader_num_workers: 4
  max_seq_len: 512
  preprocessing_num_workers: 16
  use_sampler: true
num_layers: 2
optimizer_params:
  beta1: 0.9
  beta2: 0.999
  gpus: -1
  learning_rate: 1.0e-05
  lr_decay_factor: 0.999975
  lr_decay_step: 1
  lr_minimum: 0.0
  momentum: 0.0
  warmup: 2000
  weight_decay: 0.0
trainable: true
vocab_size: 52000
weights:
- 1
- 1
```

End Of Life Conversations

```
adam_beta1: 0.9
adam_beta2: 0.999
dropout: 0.3
hidden_size: 128
input_size: 512
```

```
learning_rate: 1.0e-05
model_params:
  batch_size: 512
  dataloader_num_workers: 4
  max_seq_len: 512
  preprocessing_num_workers: 16
  use_sampler: true
num_layers: 2
optimizer_params:
  beta1: 0.9
  beta2: 0.999
  gpus: -1
  learning_rate: 1.0e-06
  lr_decay_factor: 0.999975
  lr_decay_step: 1
  lr_minimum: 0.0
  momentum: 0.0
  warmup: 2000
  weight_decay: 0.0
trainable: true
vocab_size: 52000
weights:
- 1
- 1
```

CNN

Hospital Readmission

```
adam_beta1: 0.9
adam_beta2: 0.999
dropout: 0.3
hidden_size: 128
input_size: 512
learning_rate: 1.0e-06
model_params:
  batch_size: 512
  dataloader_num_workers: 4
  max_seq_len: 512
  preprocessing_num_workers: 16
  use_sampler: true
num_layers: 2
optimizer_params:
  beta1: 0.9
  beta2: 0.999
  gpus: -1
  learning_rate: 1.0e-06
  lr_decay_factor: 0.999975
  lr_decay_step: 1
  lr_minimum: 0.0
  momentum: 0.0
  warmup: 2000
  weight_decay: 0.0
trainable: true
vocab_size: 52000
weights:
- 1
- 1
```

Dementia

```
adam_beta1: 0.9
adam_beta2: 0.999
dropout: 0.3
hidden_size: 128
input_size: 512
learning_rate: 1.0e-06
model_params:
  batch_size: 512
  dataloader_num_workers: 4
  max_seq_len: 512
  preprocessing_num_workers: 16
  use_sampler: true
num_layers: 2
optimizer_params:
  beta1: 0.9
  beta2: 0.999
  gpus: -1
  learning_rate: 1.0e-05
  lr_decay_factor: 0.999975
  lr_decay_step: 1
  lr_minimum: 0.0
  momentum: 0.0
  warmup: 2000
  weight_decay: 0.0
trainable: true
vocab_size: 52000
weights:
- 1
- 1
```

End Of Life Conversations

```
adam_beta1: 0.9
adam_beta2: 0.999
dropout: 0.3
hidden_size: 128
input_size: 512
learning_rate: 1.0e-05
model_params:
  batch_size: 512
  dataloader_num_workers: 4
  max_seq_len: 512
  preprocessing_num_workers: 16
  use_sampler: true
num_layers: 2
optimizer_params:
  beta1: 0.9
  beta2: 0.999
  gpus: -1
  learning_rate: 1.0e-06
  lr_decay_factor: 0.999975
  lr_decay_step: 1
  lr_minimum: 0.0
  momentum: 0.0
  warmup: 2000
  weight_decay: 0.0
trainable: true
vocab_size: 52000
weights:
- 1
- 1
```

MedRoberta.nl

Hospital Readmission

```
adam_beta1: 0.9
adam_beta2: 0.999
dropout: 0.3
hidden_size: 128
input_size: 512
learning_rate: 1.0e-06
model_params:
  batch_size: 32
  dataloader_num_workers: 4
  max_seq_len: 512
  preprocessing_num_workers: 16
  use_sampler: true
num_layers: 2
optimizer_params:
  beta1: 0.9
  beta2: 0.999
  gpus: -1
  learning_rate: 1.0e-06
  lr_decay_factor: 0.999975
  lr_decay_step: 1
  lr_minimum: 0.0
  momentum: 0.0
  warmup: 2000
  weight_decay: 0.0
trainable: true
vocab_size: 52000
weights:
- 1
- 1
```

Dementia

```
adam_beta1: 0.9
adam_beta2: 0.999
dropout: 0.3
hidden_size: 128
input_size: 512
learning_rate: 1.0e-06
model_params:
  batch_size: 32
  dataloader_num_workers: 4
  max_seq_len: 512
  preprocessing_num_workers: 16
  use_sampler: true
num_layers: 2
optimizer_params:
  beta1: 0.9
  beta2: 0.999
  gpus: -1
  learning_rate: 1.0e-05
  lr_decay_factor: 0.999975
  lr_decay_step: 1
  lr_minimum: 0.0
  momentum: 0.0
  warmup: 2000
  weight_decay: 0.0
trainable: true
vocab_size: 52000
weights:
- 1
- 1
```

End Of Life Conversations

```
adam_beta1: 0.9
adam_beta2: 0.999
dropout: 0.3
hidden_size: 128
input_size: 512
learning_rate: 1.0e-05
model_params:
  batch_size: 32
  dataloader_num_workers: 4
  max_seq_len: 512
  preprocessing_num_workers: 16
  use_sampler: true
num_layers: 2
optimizer_params:
  beta1: 0.9
  beta2: 0.999
  gpus: -1
  learning_rate: 1.0e-06
  lr_decay_factor: 0.999975
  lr_decay_step: 1
  lr_minimum: 0.0
  momentum: 0.0
  warmup: 2000
  weight_decay: 0.0
trainable: true
vocab_size: 52000
weights:
- 1
- 1
```

Fully Connected Structured data model

Hospital Readmission

```
adam_beta1: 0.9
adam_beta2: 0.999
dropout: 0.2
hidden_size: 128
learning_rate:
↪  1.9054607179632475e-07
model_params:
  batch_size: 64
  dataloader_num_workers: 4
  preprocessing_num_workers: 16
  use_sampler: true
num_layers: 1
optimizer_params:
  beta1: 0.9
  beta2: 0.999
  gpus: 1
  lr_decay_factor: 0.999975
  lr_decay_step: 1
  lr_minimum: 0.0
  momentum: 0.0
  warmup: 2000
  weight_decay: 0.0
weights:
- 1
- 1
```

Dementia

```
adam_beta1: 0.9
adam_beta2: 0.999
dropout: 0.3
hidden_size: 128
input_size: 512
learning_rate: 1.0e-06
model_params:
  batch_size: 32
  dataloader_num_workers: 4
  max_seq_len: 512
  preprocessing_num_workers: 16
  use_sampler: true
num_layers: 2
optimizer_params:
  beta1: 0.9
  beta2: 0.999
  gpus: -1
  learning_rate: 1.0e-05
  lr_decay_factor: 0.999975
  lr_decay_step: 1
  lr_minimum: 0.0
  momentum: 0.0
  warmup: 2000
  weight_decay: 0.0
trainable: true
vocab_size: 52000
weights:
- 1
- 1
```

End Of Life Conversations

```
adam_beta1: 0.9
adam_beta2: 0.999
dropout: 0.3
hidden_size: 128
input_size: 512
learning_rate: 1.0e-05
model_params:
  batch_size: 32
  dataloader_num_workers: 4
  max_seq_len: 512
  preprocessing_num_workers: 16
  use_sampler: true
num_layers: 2
```

```
optimizer_params:
  beta1: 0.9
  beta2: 0.999
  gpus: -1
  learning_rate: 1.0e-06
  lr_decay_factor: 0.999975
  lr_decay_step: 1
  lr_minimum: 0.0
  momentum: 0.0
  warmup: 2000
  weight_decay: 0.0
trainable: true
vocab_size: 52000
weights:
- 1
- 1
```

Lasso model

Hospital Readmission

```
alpha=0.01
max_iter=10000
```

Dementia

```
alpha=0.01
max_iter=10000
```

End Of Life Conversations

```
alpha=0.01
max_iter=10000
```

GradientBoostingClassifier model

Hospital Readmission

```
'n_estimators' :  2000
'learning_rate': 0.001
'max_depth' : 4
```

Dementia

```
'n_estimators' :  2000
'learning_rate': 0.001
'max_depth' : 4
```

End Of Life Conversations

```
'n_estimators' :  2000
'learning_rate': 0.001
'max_depth' : 4
```

Overconfidence

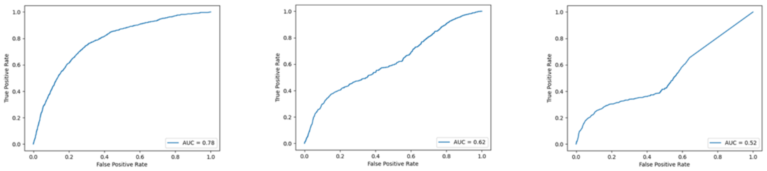

Figure 6.4: AUROC curves at increasing epochs for an CNN model

Code

The code made and used in this thesis is given to the ErasmusMC supervisors.