FAKULTA MECHATRONIKY,
INFORMATIKY A MEZIOBOROVÝCH
STUDIÍ TUL

Bakalářská práce

Alzheimer’s dementia recognition from spontaneous speech using deep neural networks

Studijní program:	B0613A140005 Informační technologie
Studijní obor:	Inteligentní systémy
Autor práce:	**Mariia Buntovskikh**
Vedoucí práce:	Ing. František Kynych Ústav informačních technologií a elektroniky
Konzultant práce:	doc. Ing. Petr Červa, Ph.D. Ústav informačních technologií a elektroniky

Liberec 2023

FAKULTA MECHATRONIKY,
INFORMATIKY A MEZIOBOROVÝCH
STUDIÍ TUL

Zadání bakalářské práce

Alzheimer’s dementia recognition from spontaneous speech using deep neural networks

Jméno a příjmení:	**Mariia Buntovskikh**
Osobní číslo:	M19000052
Studijní program:	B0613A140005 Informační technologie
Specializace:	Inteligentní systémy
Zadávající katedra:	Ústav informačních technologií a elektroniky
Akademický rok:	2022/2023

Zásady pro vypracování:

1. Familiarize yourself with the issue of speech processing using deep neural networks.
2. Familiarize yourself with the ADReSS 2020 Challenge.
3. Train a model using the ADReSS dataset to detect Alzheimer’s dementia.
4. Try different approaches to solving the task.
5. Compare the quality of the recognizer with published approaches from the ADReSS 2020 Challenge.

Rozsah grafických prací:	dle potřeby dokumentace
Rozsah pracovní zprávy:	30-40 pages
Forma zpracování práce:	tištěná/elektronická
Jazyk práce:	Angličtina

Seznam odborné literatury:

[1] LUZ, Saturnino, et al. Alzheimer's dementia recognition through spontaneous speech: the ADReSS challenge. Interspeech 2020, s. 2172–2176.

[2] PAPPAGARI, Raghavendra, et al. Using State of the Art Speaker Recognition and Natural Language Processing Technologies to Detect Alzheimer's Disease and Assess its Severity. In: INTERSPEECH. 2020. p. 2177-2181.

[3] STEVENS, Eli, Luca ANTIGA and Thomas VIEHMANN. Deep Learning with PyTorch: Build, train, and tune neural networks using Python tools. New York: Manning Publications Co. ISBN 1617295264.

Vedoucí práce:	Ing. František Kynych Ústav informačních technologií a elektroniky
Konzultant práce:	doc. Ing. Petr Červa, Ph.D. Ústav informačních technologií a elektroniky
Datum zadání práce:	24. října 2022
Předpokládaný termín odevzdání:	22. května 2023

L.S.

prof. Ing. Zdeněk Plíva, Ph.D. děkan	prof. Ing. Ondřej Novák, CSc. vedoucí ústavu

V Liberci dne 24. října 2022

Prohlášení

Prohlašuji, že svou bakalářskou práci jsem vypracovala samostatně jako původní dílo s použitím uvedené literatury a na základě konzultací s vedoucím mé bakalářské práce a konzultantem.

Jsem si vědoma toho, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu Technické univerzity v Liberci.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědoma povinnosti informovat o této skutečnosti Technickou univerzitu v Liberci; v tomto případě má Technická univerzita v Liberci právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Současně čestně prohlašuji, že text elektronické podoby práce vložený do IS/STAG se shoduje s textem tištěné podoby práce.

Beru na vědomí, že má bakalářská práce bude zveřejněna Technickou univerzitou v Liberci v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů.

Jsem si vědoma následků, které podle zákona o vysokých školách mohou vyplývat z porušení tohoto prohlášení.

2. května 2023 Mariia Buntovskikh

Rozpoznávání Alzheimerovy demence ze spontánní řeči pomocí hlubokých neuronových sítí

Abstrakt

Tato práce je zaměřena na výzvu ADReSS (Alzheimer's Dementia Recognition through Spontaneous Speech) z konference INTERSPEECH 2020. K řešení této výzvy byly použity různé přístupy k dosažení základních výsledků pro klasifikační a regresní úlohy.

V rámci předzpracování dat bylo nutné provést extrakci příznaků pro akustická a lingvistická data. Byly použity předtrénované modely: příznaky ze zvukového záznamu byly extrahovány modelem SpeechBrain pro verifikaci mluvčích založeným na Time-Delay Neural Network (TDNN) a příznaky z přepisů byly extrahovány modelem Bidirectional Encoder Representations from Transformers (BERT).

První část této práce se zaměřuje na vývoj klasifikačního modelu pro rozpoznávání Alzheimerovy choroby (AD). Výsledky ukazují, že model neuronové sítě dosahuje nejvyšší klasifikační přesnosti 85 % na dané testovací množině s použitím transkripcí a překonává základní model o 10 % pro lingvistická data. Model K-Nearest Neighbour (KNN) dosáhl přesnosti 71 % pro akustická data, což je o 14 % více než základní výsledek.

Druhá část studie se zaměřuje na vývoj regresního modelu pro odhad skóre Mini-Mental State Examination (MMSE). Modely jsou hodnoceny pomocí statistických ukazatelů, jako je střední kvadratická chyba (RMSE) a hodnoty R-squared (r2). Výsledky ukazují, že model ElasticNet dosahuje nejnižší hodnoty RMSE 4,35 a překonává základní model o 0,85 bodu.

U obou úloh dosažené výsledky překonaly nejlepší známé výsledky pro úlohu ADReSS.

Závěrem lze říci, že tato práce prokazuje účinnost modelů strojového učení pro klasifikaci AD a predikci skóre MMSE. Výsledky ukazují potenciál těchto modelů pomáhat při včasné detekci a sledování AD a poskytují poznatky o kvalitě datového setu.

Klíčová slova: Alzheimerova choroba, Umělá inteligence, Strojové učení, Zpracování přirozeného jazyka, Zpracování řeči

Alzheimer's dementia recognition from spontaneous speech using deep neural networks

Abstract

This thesis is focused on ADReSS (Alzheimer's Dementia Recognition through Spontaneous Speech) challenge at INTERSPEECH 2020. To solve this challenge different approaches were used to achieve baseline results for classification and regression tasks.

As a part of data preprocessing, feature extraction was needed for acoustic and linguistic data. Pretrained models were used: features from audio recording were extracted by SpeechBrain speaker verification model based on Time-Delay Neural Network (TDNN) and features from transcriptions were extracted by Bidirectional Encoder Representations from Transformers (BERT) model.

The first part of this work focuses on developing a classification model to recognise Alzheimer's disease (AD). The results show that the Neural Network model achieves the highest classification accuracy of 85% on the given testing set using transcriptions, outperforming the baseline model by 10% for transcriptions. For speech recording, K-Nearest Neighbour (KNN) has achieved test accuracy of 71%, which is higher than the baseline result by 14%.

The second part of the study focuses on developing a regression model for predicting Mini-Mental State Examination (MMSE) scores. The models are evaluated using performance metrics, such as root mean squared error (RMSE) and R-squared (r2) values. The results show that the ElasticNet model achieves the lowest RMSE of 4.35, outperforming the baseline model by 0.85.

For both tasks, achieved results have outperformed the best-known results for the ADReSS challenge.

In conclusion, this thesis demonstrates the effectiveness of machine learning models for the classification of AD and the prediction of MMSE scores. The results highlight the potential for these models to assist in the early detection and monitoring of AD, and provide insights about dataset quality.

Keywords: Alzheimer's disease, Artificial Intelligence, Machine Learning, Natural Language Processing, Speech Processing

Contents

List of Tables

List of abbreviations

AD	Alzheimer's Disease
ADReSS	Alzheimer's Dementia Recognition through Spontaneous Speech
AI	Artificial Intelligence
BERT	Bidirectional Encoder Representations from Transformers
CV	Cross-Validation
ECAPA-TDNN	Emphasized Channel Attention, Propagation and Aggregation in TDNN
GP	Gaussian Process
KNN	K-Nearest Neighbour
LOSO	Leave One Subject Out
MCI	Mild Cognitive Impairment
ML	Machine Learning
MLP	Multi-layer Perceptron
MMSE	Mini-Mental State Examination
NLP	Natural Language Processing
NN	Neural Network
PCA	Principal Component Analysis
RMSE	Root Mean Squared Error
SGD	Stochastic Gradient Descent
SVM	Support Vector Machine
TDNN	Time-Delay Neural Network

1 Introduction

Alzheimer's disease (AD) is a neurodegenerative disorder that affects millions of people worldwide. It is characterized by progressive memory loss, cognitive impairment, and behaviour changes. There is no cure for AD, but the current state of medicine can slow the progression and improve everyday life for patients and their families. Early detection of AD is critical for early intervention, which can help to have a comfortable life for as long as possible. Current methods for diagnosing AD usually are costly, time-consuming, and often only accurate when the disease has already progressed.

Artificial intelligence (AI) and natural language processing (NLP) show promising results in identifying AD signs from spontaneous speech, including early stages. Spontaneous speech changes with every AD stage, which can be used to indicate some cognitive impairment. Spontaneous speech analysis with AI is a non-invasive and cheap tool to detect AD.

This thesis is focused on Alzheimer's Dementia Recognition through Spontaneous Speech (ADReSS) challenge: the main purpose is to develop a machine learning (ML) model to classify AD and non-AD individuals from spontaneous speech.

The results of this challenge can significantly change current approaches to the detection and management of AD. An accurate speech-based diagnostic tool could help identify individuals at risk of developing AD before symptoms become severe, enabling better management of the disease, reducing healthcare costs and improving patient outcomes.

1.1 Main goals of this thesis

The main goals of this thesis are to gain a comprehensive understanding of the application of neural networks in speech processing, to explore the ADReSS 2020 Challenge, to develop a model that can identify Alzheimer's dementia utilising the ADReSS dataset, to experiment with different methods for classification and regression tasks, and to compare achieved results against the existing published techniques for the ADReSS 2020 Challenge.

1.2 Speech processing in Machine Learning

Speech processing is one of the most important directions in the Machine Learning (ML) field thanks to the wide scope of usage and great possibilities it provides. Speech processing is a process, when computers can analyse a speech and get needed information from it.

Currently, Artificial intelligence in this field is capable to detect and recognise specific humans, emotions of speech, the main meaning of a given sentence or the whole long speech, to make a summary and to translate in real time. State of the art models usually are extremely robust to any noise.

The main purpose of speech processing is to achieve and overcome the human level of performance on speech tasks.

Speech processing is able to improve a lot in our daily life. It can speed up medical analysis, it can help us to communicate better in any language in real time, and it can help us to improve our speech itself (for example by detecting filler words or non-native accents). It can be used as a transcription or assistance tool in healthcare to help doctors and nurses with their job.

1.3 What is Alzheimer's disease?

Alzheimer's disease (AD) is a brain disorder that with time dangerously affects everyday life. The main known risk is age. AD usually occurs in individuals above age 65 [1].

There are a few ways to diagnose AD:

- **Brain scans** such as magnetic resonance imaging (MRI), computerised tomography (CT), and positron emission tomography (PET). Brain imagining can help, however, sometimes it's difficult to know what is normal age-related brain changes and what are abnormal changes.

- **Laboratory tests**: blood, urine, measure the levels of proteins associated with AD by collecting cerebrospinal fluid.

- **Memory and cognitive tests**. There are several screening tests to detect cognitive impairment. The most popular is Mini-Mental State Examination (MMSE); the maximum score is 30. The score is considered normal if it's higher than 25. A score of 24 or below can indicate possible memory and cognitive problems.

  One of MMSE's disadvantages is that scores can be affected by years of education, age and personal background [2].

1.3.1 Alzheimer's disease stages

There are 5 AD stages:

- **Asymptomatic stage**: an individual has biological changes in the brain before any cognitive symptoms.
  - This stage can take up to 20 years [1].
  - MMSE is normal (25-30 points).
- **Mild Cognitive Impairment (MCI) stage**: people in this stage can have some memory or cognitive function problems but those changes usually don't affect daily activities. Not every individual with MCI will develop Alzheimer's. MCI can be an early stage of AD if hallmark brain changes are present [3].
  - This stage can take up to 7 years.
  - MMSE score at this stage can be 24-30 points.
- **Mild (Early stage of AD)**: Common first signs are: difficulty remembering newly learned information (e.g. new language), word-finding, and taking longer to finish usual everyday tasks.
  - Lasts about 2 years.
  - MMSE at this stage is 21-26 points.
- **Moderate (Middle stage of AD)**: In this stage, individuals have problems with reasoning, conscious thoughts and correct detecting and recognising (e.g. sounds, words, time or place). It's almost impossible to learn new information and express thoughts correctly with brain damage caused by AD in this stage [1].
  - Lasts about 2-4 years [4].
  - MMSE at this stage is 15-20 points.
- **Severe (late stage of AD)**: In this stage, AD makes everyday life almost impossible: usually there is no body control (bladder control, eating and swallowing food independently, mouth breathing when needed). Speech is totally lost or limited up to 10 words, which makes communication impossible.
  - Lasts about 1.5 - 2.5 years.
  - MMSE at this stage is lower than 15 points.

1.4 ADReSS 2020 Challenge

AI is capable to help diagnosing AD in different ways:

- Help in studying and detalisation biomarkers (biological signs of disease, which can be found on brain scans or in blood tests).
- Detect new and old disease signs in the behaviour of individuals, which are not always easy to notice.

AI is not only capable to diagnose AD by the already known rules, but it can analyse a lot of information and find heretofore unknown patterns and signs to detect AD much earlier than it's possible now, which can help preventing some complications and maybe to find a breakthrough cure.

One of the ways to detect AD by AI is a speech analysis. There are changes in a speech in every stage of Alzheimer's disease (except for the asymptotic stage), as it was described in subsection 1.3.1 on page 13. It is possible to detect AD not only by word analysis but also emotions and voice level. As the disease progresses an individual has not only problems with words but with stable voice level and emotions as well (e.g. unexpected shouting).

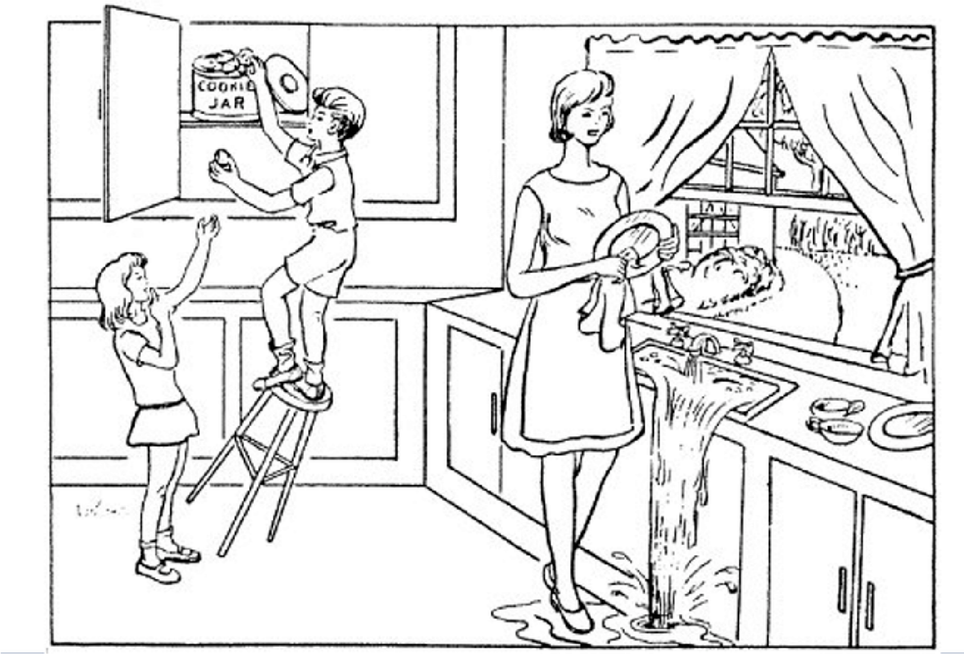

Figure 1.1: The Cookie Theft Image from the Boston Diagnostic Aphasia Examination [5].

The ADReSS challenge [6] is focused on the detection of any cognitive impairment and Alzheimer's dementia by using spontaneous speech samples and their transcriptions.

Spontaneous speech samples were carefully collected and selected for this challenge to reduce common problems: unbalanced feature distributions and variations in audio quality.

One of the purposes of this challenge is to provide a new standardised dataset to make it easier to compare and test different approaches to detect AD.

Every audio recording contains a dialogue between a healthy or control subject and a doctor. The subject needs to describe what is happening in the Cookie Theft picture (see figure 1.1).

The challenge contains two tasks:

- Classification task, where it's required to create a model to predict label for each subject (AD or non-AD). The baseline is 62.5% for acoustic features and 75% for linguistic features.
- Regression task for MMSE score prediction (integer from 0 to 30). The baseline is an RMSE of 6.14 for acoustic features and 5.20 for linguistic features.

In each task, it's allowed to use speech and language data.

1.5 Thesis structure

This work is further structured as follows:

- The next chapter 2 describes how few other studies tried to solve ADReSS challenge tasks.
- It follows with methods and dataset descriptions (chapters 3 and 4) and evaluations of the chosen methods (chapter 5).
- Then in chapter 6 there is a comparison between achieved results in this work and results found on the internet.
- Finally, the last chapter 7 is the conclusion of this work.

2 Related works

The ADReSS challenge was introduced in the paper [6]. In this study, researchers used different feature extractors for audio recordings (e.g. Multi-resolution Cochleagram features (MRCG), and Minimal (basic statistics features: mean, standard deviation, median, minimum and maximum)). These feature sets contained a variety of features such as mel-frequency cepstral coefficients (MFCC), voice quality, fundamental frequency (F0), line spectral pairs (LSP), intensity features, and low-level descriptors (LLDs). A basic set of different language measures (e.g. duration, total utterances and others) was computed from transcriptions.

The study used five different methods for the classification task: Linear Discriminant Analysis (LDA), Decision Tree (DT), Nearest Neighbour (1NN), Random Forest (RF), and Support Vector Machine (SVM). The study also implemented a two-step classification process, segment-level (SL) classification and majority vote (MV) classification. The results show that the 1NN method had the best accuracy (57%) for acoustic features using the ComParE feature set for AD detection. LDA and RF on linguistic features provided the best accuracy for the classification task (75%) [6].

The baseline experiments for the regression task used five different methods: DT, linear regression, Gaussian process regression, least-squares boosting, and SVM. The results showed that DT provided the best RMSE scores: RMSE of 6.14 for MMSE prediction with MRCG features for acoustic data, and RMSE of 5.20 for linguistic data [6].

In another paper, the work of Haulcy and Glass tested several models, comparing their performance on the ADReSS dataset. The top-performing classification models were the SVM and RF classifiers trained on BERT embeddings - both achieved an accuracy of 85% on the test set. The best-performing regression model was the gradient boosting regression model trained on BERT embeddings and CLAN features, which had an RMSE of 4.56 on the test set [7].

A different study (done by Martinec and Senja) presented an accuracy of 77% for the classification task, using a Logistic regression model with different features. For regression, the best RMSE score of 4.44 on the test set was achieved by the SVM model [8].

The related works for the classification task of the ADReSS challenge report accuracies up to 85% for linguistic features only. The accuracy in most cases decreases to below 70% when only audio features are used. For the regression task, RMSE is around 4.5 for linguistic features only.

3 Methodology

This chapter contains the theory that is needed to solve the ADReSS challenge: speech and text processing, data preprocessing, transfer learning for feature extraction and algorithms for classification and regression tasks.

3.1 Audio and text data for speech processing

In machine learning, audio and text data are two of the most used data types in speech processing.

Usually, audio data are preprocessed by trained feature extractors or spectrograms, which represent the sound signal in a suitable way for ML algorithms. Then these features can be used to train a model that can detect voice activity, and recognise words or emotions.

Sometimes it's good to transcript audio data to text format (e.g. to isolate speech meaning from voice level or noise). Text data can be preprocessed by a tokeniser to extract words and punctuation signs (tokens). Then those tokens can be encoded into numerical vectors (e.g. word embeddings). Numeric data can be used more effectively than raw text in ML algorithms. Then it can be used for natural language processing (NLP) tasks such as sentiment analysis, classification and translation.

3.1.1 Data preprocessing

Data preprocessing is the first step for speech processing in machine learning algorithms. The main goal is to prepare original data in a way to maximise the performance of the ML model and to minimise any errors. Speech preprocessing can vary depending on a specific task and dataset, but usually, speech preprocessing contains a few steps:

- Noise removing.
- Feature extraction.
- Normalisation.
- Augmentation to increase wanted variations on the dataset and improve ML model robustness and performance.

- Split data into training, validation and testing sets with the same label proportions to minimise imbalance between sets.

Principal Component Analysis

PCA can decrease the number of correlated features in a given dataset. The principal components (uncorrelated features) capture the maximum variance in the data. Choosing the appropriate number of principal components that retain enough information from features is important to minimise information loss [9].

By reducing the number of dimensions in the data, PCA can simplify the analysis, and improve the speed of ML algorithms.

PCA can be used for different tasks:

- Data compression.
- Feature extraction.
- Data visualisation.

PCA works by finding the eigenvectors and eigenvalues of the covariance matrix of the input data. The eigenvectors represent the directions of maximum variance in the data, and the eigenvalues represent the amount of variance explained by each eigenvector [9].

3.2 Transfer learning

Transfer learning is a process when a pre-trained model from another task can be reused to solve a different but related problem [10].

It is relatively rare to have a dataset of sufficient size. It is common to pre-train a model on a very large dataset and then to use it either as an initialisation or as a fixed feature extractor for the task.

In the case of the small dataset with data similar to the original dataset, transfer learning can be used, but it will be training only the last layers because the first layers will be used for general feature identification.

Transfer learning doesn't always work: for example, in the case of a dataset that is too small and very different to the original dataset, so there is a requirement to select more data.

There are method descriptions, which are used for feature extraction from speech recordings and transcriptions in the following sections: in section 3.2.1 is a description of the model which works with text inputs and in section 3.2.2 is the description of models which works with audio recordings.

3.2.1 Bidirectional Encoder Representations from Transformers

Bidirectional Encoder Representations from Transformers (BERT) is a pre-trained NLP model developed by Google [11]. It is based on the Transformer architecture and the output is word embeddings.

BERT uses a bidirectional self-attention approach to generate word embeddings where every token attends to context on both sides (left and right). This allows BERT to capture complex relationships between words and their context, leading to significant improvements in NLP tasks [11].

Since the release of BERT, several variations and extensions of the model have been developed, such as RoBERTa [12], ALBERT [13], and ELECTRA [14], which have achieved state-of-the-art performance on various NLP tasks.

BERT can handle different text inputs named ”sentence” where the maximum number of tokens is 512 (This limitation is made against low-quality output). It’s important to remember that the BERT sentence is not referring to a usual linguistic sentence, but to the input token sequence [11].

If the input sentence is too long for BERT, there are two techniques to use:

- Trimming the input text.
- Dividing the input text into segments of equal length and running each segment through BERT, then averaging the BERT embeddings.

It’s required in every input to use defined tokens: [SEP] (separator) and [CLS] (classification token).

How to use BERT:

- Prepare input sentences for tokenisation: correct linguistic errors, standardise the format along all input data and add special tokens, as required. The output of this step must be a normalised text with special tokens and up to 512 tokens long.
- Tokenise the input data. Tokenisation is the process of breaking down the input text into individual tokens or subwords and characters. In the case of subwords and characters, the token starts with ##. The output is an array of strings (words, subwords, characters) [11].
- Convert an array of strings into an array of BERT vocabulary indices. BERT vocabulary is not case-sensitive and has a size of around 30,000 tokens. Words that are not part of the vocabulary are represented as subwords and characters.
- Evaluate BERT on text to extract embeddings: the output contains the following hidden states: number of hidden layers (always 13 layers: the first layer is the input data and 12 BERT’s layers), how many words (tokens) in a given BERT sentence, feature array (length of 768).
- Create sentence embeddings from hidden states. There are a few ways to work with outputs, in this thesis two approaches were used to produce a single embedding of 768 length for each sentence:
  - Average the second to last hidden layer of each token.
  - Sum the second to last hidden layer of each token.

3.2.2 SpeechBrain models

SpeechBrain is an open-source toolkit built on PyTorch for speech processing that provides a wide range of pre-trained models, tools for data preprocessing, feature extraction and building custom models [15].

SpeechBrain provides some of the commonly used pre-trained models, which include:

- Automatic Speech Recognition (ASR) models to convert speech audio into text.
- Text-to-Speech (TTS models) to convert text into speech audio.
- Speaker Identification (SID) models to identify the speaker of an audio recording.
- Voice Activity Detection (VAD) models to detect whether an audio recording contains speech or silence.

In this work, two SpeechBrain models were tested: TDNN (Time delay neural network) and ECAPA-TDNN (Emphasized Channel Attention, Propagation and Aggregation in TDNN) to extract features from audio recordings.

TDNN

A Time Delay Neural Network (TDNN) is a type of feedforward NN that can process sequential data, such as speech signals or time series data [16].

The main difference between usual feedforward NN and TDNN is that TDNN processes each input independently with delay elements. Thanks to added delay elements to the input layer, TDNN model is capable to predict output for current input based on previous and future inputs. The output of the delay elements is fed to the series of hidden layers.

The SpeechBrain speaker verification (SP) model based on TDNN is designed to work with audio recordings of different lengths and provide a fixed-length output for each input recording. The model is trained on Voxceleb1 [17] and Voxceleb2 [18] training data. It can be used to extract speaker embeddings [15].

This SP system is made up of two components: a TDNN model and a statistical pooling method. During training, the system uses a loss function called Categorical Cross-Entropy Loss.

The SpeechBrain SP is trained with recordings sampled at 16kHz. SpeechBrain framework can automatically normalise input audio (resampling and single channel selection) if needed.

The SpeechBrain SP system uses a combination of input processing, time-delayed features, pooling, and normalisation techniques to handle audio recordings with different lengths and output fixed-length representations.

ECAPA-TDNN

The ECAPA-TDNN model is an extension of the TDNN model [19].

The main difference between TDNN and ECAPA-TDNN is that ECAPA-TDNN has a few modifications added to the original TDNN architecture to improve its performance. Specifically, these modifications include:

- Using emphasised channel attention to weight the input features based on their importance. This enables the model to focus on the most relevant features while ignoring noise or irrelevant information.
- Using propagation and aggregation to enhance the representation of the input features across different time frames. This is achieved by propagating and aggregating the feature maps across multiple TDNN layers, which allows capturing more dependencies in the input data.
- Adding a convolutional layer at the beginning of the network, which helps to extract high-level features from the input data before passing it through the TDNN layers.

The SpeechBrain SP system based on ECAPA-TDNN is trained on Voxceleb1 [17] and Voxceleb2 [18] training data.

The system can extract speaker embeddings thanks to attentive statistical pooling [15]. During training, the system uses a loss function called Additive Margin Softmax Loss [20]. To perform speaker verification, the system uses the cosine distance metric to compare the speaker embeddings.

The training process and usage of this system are similar to the system based on the TDNN. Therefore, it can be used in the same manner as a TDNN-based system.

Difference between TDNN and ECAPA-TDNN

TDNN and ECAPA-TDNN are both NN architectures used in speech recognition tasks. Usually, ECAPA-TDNN can outperform TDNN models in speech recognition tasks.

However, there are few cases where ECAPA-TDNN performs worse than TDNN (e.g. specific task problems, implementation, hyperparameters).

3.3 Classification

Classification is a supervised learning method in ML that can work with any type of input data and any number of classes. Classification outputs are assigned as discrete class labels (for example, male or female).

The classification algorithm is trained on labelled data (each data point has a class label). The algorithm learns to identify relationships between features and each class to predict the class labels for new data points.

3.3.1 Neural Network

A Neural Network (NN) for classification tasks usually includes multiple layers of interconnected nodes, where each node receives an output from the previous layer as an input and performs a calculation by using an activation function (e.g. sigmoid or softmax) to transform given input and produce an output, that is passed on to the next layer nodes. The first layer receives the input features and the output layer produces the final predicted labels for each class [21].

There are different types of neural networks used for classification tasks, the most popular are:

- Feedforward NN.
- Convolutional NN (CNN).
- Recurrent NN (RNN).

An example of simple NN can be seen on figure 3.1.

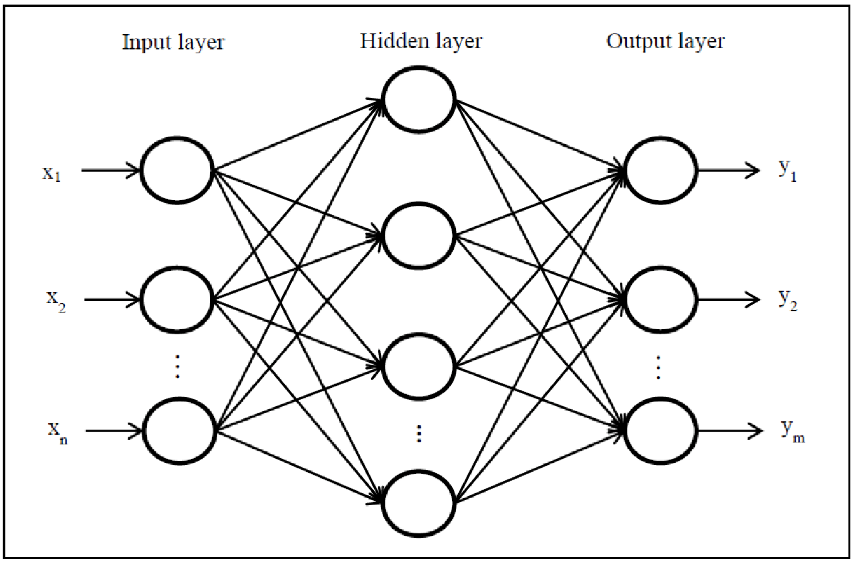

Figure 3.1: Example of NN structure [22].

Between the first and the last layers can be a different number of hidden layers, which allow the NN to capture complex patterns in the given data.

During the training phase, the NN weights and biases are adjusted using a backpropagation process, which tries to minimise the error between the predicted output and the actual label. There are different types of optimisation algorithms such as stochastic gradient descent, Adam and others [21].

3.3.2 Gaussian Process

The Gaussian process (GP) is a probabilistic non-parametric method, which can be used for binary classification tasks. It's based on Gaussian (normal) probability distribution. The normal distribution describes a symmetrical plot of data around its mean value [23].

3.3.3 Support Vector Machine

The Support Vector Machine (SVM) classifier is trained to identify the optimal hyperplane, which effectively divides the data into distinct classes.

If the data is not linearly separable, SVM can utilise a kernel function to map the input data to a higher-dimensional feature space where linear separability may be achieved. In this higher-dimensional space, SVM can still identify the best hyperplane and provide predictions in the original input space [24].

One of the key advantages of SVM is its ability to effectively handle high-dimensional data and its robustness to overfitting.

3.3.4 K-nearest neighbour

K-nearest neighbour (KNN) is a non-parametric method based on measuring distances used for different ML tasks. In the case of classification, outputs are assigned as discrete class labels (for example, male or female) [25].

The KNN method doesn't have a training phase in its usual meaning. In this method, the training phase consists only of storing the training elements and the class labels. The classification phase of KNN method is searching distances between the unlabeled element and the training element in every class within the feature space. The unlabeled element is classified by assigning the label which is most frequent among the k training samples nearest to the query point.

Usually, k is chosen as a square root of N, where N is the number of training elements.

The KNN classifier requires a setting for k and distance function. The most popular distances used in KNN are Hamming distance(L1) and Euclidean distance(L2).

3.3.5 Metrics

For the classification task, the prediction can be either right or wrong. Analysing the number of true positives, false positives, true negatives, and false negatives predicted by the model helps to understand how well a model is trained [26], where:

- **True positives (TP):** the number of cases where the model correctly predicts a subject as true (e.g. an AD subject is identified as an AD subject).
- **False positives (FP):** the number of cases where the model incorrectly identified a subject as true (e.g. a non-AD subject is identified as an AD subject).
- **True negatives (TN):** the number of cases where the model correctly identified a subject as false (e.g. a non-AD subject is identified as a non-AD subject).

- **False negatives (FN):** the number of cases where the model incorrectly identified a subject as false (e.g. an AD subject is identified as a non-AD subject).

To have clear classification reports following calculations are needed:

- **Precision** is the proportion of TP among the total number of predicted positives (TP and FP), which shows how many of the predicted positive cases were actually positive.
- **Recall** is the proportion of TP among the total number of actual positives (TP and FN), which shows how many of the actual positive cases were correctly predicted as positive.
- **F1 Score** is the mean of precision and recall to have more balanced summarisation of model performance. The formula is:

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

- **Accuracy** is the proportion of correctly classified subjects among the total number of subjects. The formula is:

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

3.4 Regression

Regression is a supervised learning method in ML. Regression outputs are assigned as feature numerical values (for example, house price predictions). Same as for classification, regression algorithms are trained on labelled data.

3.4.1 Least Absolute Shrinkage and Selection Operator

The Least Absolute Shrinkage and Selection Operator (Lasso) regression is a linear regression model extension that has a penalty term to the cost function in order to prevent overfitting. The penalty term is proportional to the absolute value of model weights, which causes some of the weights to become zero. It means that Lasso regression can perform feature selection by identifying and discarding irrelevant or redundant input features [27].

3.4.2 Elastic net

Elastic Net regression is a linear regression model that combines L1 (Lasso) and L2 (Ridge) regularisation techniques. Like Lasso regression, Elastic Net adds a penalty term (L1) to the cost function that is proportional to the absolute value of the weights of the model, which can help with feature selection. Additionally, Elastic

Net also adds a second penalty term (L2) to the cost function that is proportional to the square of the weights of the model, like in Ridge regression. This additional term helps to address some of the limitations of Lasso regression, which selects only one feature among a group of highly correlated features [28].

3.4.3 KNN Regressor

Same as for classification tasks, in KNN regressor, the value of k represents the number of nearest neighbours to consider when making a prediction. The output value for the new data point is computed as the mean (or median) of the output values of its k nearest neighbours [25].

KNN regression can work well when the relationships between variables are complex and not easily captured by a simple mathematical model. It is also a computationally inexpensive algorithm for a low number of neighbours and small datasets.

3.4.4 Metrics

Root Mean Square Error

RMSE stands for Root Mean Square Error, which is a commonly used metric to evaluate the accuracy of a regression model's predictions. RMSE measures the difference between the predicted values and the actual values, expressed in the same units as the response variable [29].

A lower RMSE value indicates better performance, as it means that the model's predictions are closer to the actual values. However, the interpretation of the RMSE value also depends on the scale of the response variable. For example, a RMSE of 10 for a response variable with a range of 0-100 may be acceptable, but a RMSE of 10 for a response variable with a range of 0-1 would be considered very bad.

One important consideration when using RMSE is that it gives equal weight to all errors, regardless of their direction (i.e., overestimation or underestimation).

R2-score

R2-score is a metric used to evaluate the performance of a model. It measures the proportion of the variance in the dependent variable that is predictable from the independent variables used in the model [30].

The R2-score usually ranges from 0 to 1, with a higher value indicating a better fit of the model to the data: an R2-score of 1 indicates that the model perfectly predicts the dependent variable, while an R2-score of 0 indicates that the model is no better than predicting the mean of the dependent variable.

A high R2-score sometimes can mean overfitting to the training data and not generalising well to new data.

When R2-score is a negative number [31], it means that the model is performing very poorly and is actually worse than just predicting the mean value of the dependent variable. This could be due to several reasons, such as overfitting, incorrect model specification (e.g. linear regression model for nonlinear relationship between values in the dataset), outliers or missing data, dataset quality issues.

2-D scatter plots

2-D Scatter plots of actual and predicted values (as axes) with the regressed diagonal line (x=y) were used to visualise regression results. If a model has a low RMSE, all the points would be close to this diagonal line. The higher the RMSE, the weaker model is, and the more dispersed points are (away from this diagonal line).

From this plot type it's possible to see model-related issues: for example, if the residuals are heteroscedastic or autocorrelated, or what class has more wrong predicted outputs and how much wrong they are.

4 Dataset

The original dataset version, which was used, has 2 classes (AD and non-AD) and 156 subjects. The dataset is divided into a 48-subject test set and a 108-subject train set. Sets are balanced by classes (AD or non-AD), age interval and gender as shown in table 4.1.

Table 4.1: Number of subjects for each age interval in train and test sets

Age interval	Train set		Test set	
	AD-subject	non-AD subject	AD-subject	non-AD subject
[50, 55]	1	2	2	2
(55, 60]	10	9	4	3
(60, 65]	13	13	5	6
(65, 70]	15	13	6	5
(70, 75]	11	12	5	7
(75, 80]	4	5	2	1
Total	54	54	24	24

Each subject has the following data:

- Label (AD or non-AD).
- Gender (female or male).
- Age (the youngest subject is 50 years old and the oldest is 80 years old),
- Audio recording:
  - .wav format file.
  - Contains a one-on-one interview between a subject and a doctor.
  - Quality of each speech recording is affected by different recording conditions such as distance between the microphone and each interview participant or background noise.
- Audio-chunks, which were extracted from the audio recording by voice activity detection:

  - Maximum duration of each speech segment is 10 seconds.
  - Not every audio segment contains speech in it.
  - An average number of speech segments for each subject is 24.86 [32].

- Transcripts of audio recording:
  - Codes for the Human Analysis of Transcripts (CHAT) transcription format was used [33]. CHAT coding system contains a lot of information about the speech to analyse it: each speaker is identified with a unique ID (e.g. ”INV” ID stands for ”investigator”); utterance boundaries, which helps to distinguish between individual statements; grammatical markers to analyse different grammatical structures.
  - When a word is unintelligible or unclear and can’t be transcripted, it’s written as ”xxx”.
  - The maximum number of words for each subject is 505 and the minimum is 27.

- MMSE (see proportion in table 4.2):
  - Integer from 0 to 30 or NaN if MMSE score wasn’t calculated, where a higher score means fewer mental and cognitive problems.

Table 4.2: Dataset proportion for MMSE score

MMSE interval	Train set		Test set	
	AD-subject	non-AD subject	AD-subject	non-AD subject
[0, 15]	20	0	6	0
(15, 20]	22	0	8	0
(20, 24]	7	0	5	1
(24, 30]	5	53	5	23

5 Experiments

For each task in this work extracted features by pre-trained models were used. The top 3 results were always selected by validation accuracy and evaluated on the testing set. Training accuracy was higher than 90% almost for each model. The gap between training accuracy and validation accuracy was always more than 12%.

5.1 Data preprocessing

All input data were pre-processed to improve the models' performance.

Labels were transformed to numeric representation (0 for non-AD and 1 for AD class).

Numeric vectors were extracted from the speech recordings by SpeechBrain pre-trained speaker verification models. Two different models were adapted for the ADReSS dataset:

- TDNN model, which was trained on Voxceleb 1+ Voxceleb2 training data to extract speaker embeddings. Each extracted speaker embedding has a length of 512.

- ECAPA-TDNN model, which was trained on Voxceleb 1+ Voxceleb2 training data as well. Each extracted embedding has a length of 192.

Before solving classification and regression tasks extracted features from both models (TDNN and ECAPA-TDNN) were compared. The feature extraction by TDNN was faster due to the more complex architecture of ECAPA-TDNN. Evaluation on the validation set has shown that features extracted by the TDNN model lead to better and more stable results for different models (for classification and regression), so for both ADReSS tasks, only features extracted by the TDNN model were used.

All transcriptions were transformed by the BERT model to get a numeric representation of each speaker of length 768. The sentence embedding was calculated with two different approaches: calculating the average of all token vectors and calculating the sum of all token vectors.

The original training dataset was split into training and validation sets. The validation set size is 1/3 of the original dataset, which means it has 36 subjects. Validation and training sets have equal labels proportion: there are 36 subjects for each label for training and 18 subjects for each label for validation.

PCA was used to visualise embeddings from different extracted models. After PCA was applied it's easy to see, that summed BERT embeddings should be successful with linear models (it can be separated by a line on y=0, see figure A.1). Audio embeddings are too complex to be successful with simple models.

The ADDReSS challenge provides sex and age information for each subject too, but the main purpose is to use audio and text features to predict labels for the test set.

5.2 Classification task

For the classification task, different algorithms were tested with different hyperparameter settings for each extracted feature:

- KNN with different k (from 1 to 10).
- SVM with different degrees (up to 5) and regularisation parameter (C).
- Decision Tree with a different maximum depth, criterion and the minimum number of samples for each leaf.
- Random Forest with a different number of estimators (trees), criterion and the maximum depth.
- SGD with different regularisation parameters.
- Multi-layer Perceptron (NN) with different learning rates, regularisation parameters, activation functions and weight optimisation.
- Gaussian Process (GP) without a kernel.

The random state was always set to 42, which is useful for the reproducibility and consistency of results. The random state wasn't needed for data splitting, because data was split before the training process and was always the same for each experiment, but it was needed to ensure that the same initial weights (and other parameters) are used during each code run to achieve a fair comparison of models.

The mean BERT embeddings achieved the most accurate and stable outcome among all the models that were tested. The best results are shown in table 5.1.

MLP (settings: maximum number of iterations = 100, the strength of the l2 regularisation (alpha) = 0.00005, number of hidden layers = 100, optimisation = ADAM (Adaptive Moment Estimation), activation function = ReLU (Rectified Linear Units)) achieved the best results of 83% accuracy for testing and validation sets, which was expected as average accuracy with the k-fold method showed an accuracy of 82%.

MLP with the same settings achieved a testing accuracy of 85% for summed BERT embeddings, but when this algorithm was tested for different data splitting (training and validation splits) in the k-fold method, the average accuracy was 82%, which means that this high accuracy is achieved because of current data splitting, not by the model performance.

Gaussian process algorithm showed unexpectedly high accuracy for prepared data split of 81% when average accuracy for this method over 6 folds was only 70%. GP algorithm was more stable for mean BERT embedding than for summed BERT embeddings.

Table 5.1: Top 3 results for mean BERT embeddings

Classifier	Label	Precision	Recall	f1-score	Test acc. [%]	Valid acc. [%]
MLP	non-AD	79	92	85	83	83
	AD	90	75	82		
GP	non-AD	76	92	83	81	81
	AD	89	71	79		
SVM, degree=3	non-AD	76	92	83	81	78
	AD	89	71	79		

SVM with 3rd degree achieved an accuracy of 81% for testing, which was expected. Linear SVM achieved lower accuracy for mean BERT embeddings because data is a little bit complex to use linear classifier, as can be seen from PCA components graph A.2 in attachments A.

Linear SVM with regularisation C=1 was enough to achieve an accuracy of 77% for summed BERT embeddings, as it was expected. Higher degrees didn't affect performance a lot in this case.

SGD testing accuracy is 79% for summed BERT embeddings, which was expected. SGD classifier works well for both (summed and mean) BERT embeddings. It's stable and has high accuracy comparing to all tested methods.

Table 5.2: Top 3 results for summed BERT embeddings

Classifier	Label	Precision	Recall	f1-score	Test acc. [%]	Valid acc. [%]
MLP	non-AD	87	83	85	85	86
	AD	84	88	86		
SGD	non-AD	73	92	81	79	83
	AD	89	67	76		
Linear SVM, C=1	non-AD	76	79	78	77	75
	AD	78	75	77		

For audio embeddings the most successful and stable algorithm was KNN, but even this algorithm wasn't robust enough to deal with low quality of embeddings caused by a very high level of noise in audio recordings. KNN with k set to 2 achieved testing accuracy of 71% and validation accuracy of 61% when testing accuracy was expected to be around 62%. No classifier achieved good enough results for audio embeddings. Best-performed classifiers for audio embeddings are shown in table A.1 on page 41.

Mixing features (concatenating audio embeddings with BERT embeddings) has always led to lower accuracy.

For each algorithm, the f-1 score was higher for the non-AD class or equal to the AD class, which means that the model is better at identifying and predicting the non-AD class. It can be caused that non-AD subjects have more data (e.g. more words in speech) and the model has more information to learn from. Check tables A.1 (page 41), 5.1 and 5.2 to see precision, recall and f1-score for each feature.

To check that different dataset splitting does not affect test accuracy a lot, the K-fold method was adapted for this task, where K is the number of groups the data is split into. One group is used for validation while K-1 groups are used for training.

LOSO (leave one subject out), 3-fold and 6-fold were tested. The results weren't distinctive among different splits, which shows that dataset splits don't usually affect test accuracy. Average cross-validation (CV) and test accuracy for each classifier are shown in tables 5.3 and 5.4.

Table 5.3: Average of CV and test accuracies for BERT embeddings (K = 6)

Classifier	Test accuracy [%]		Valid. accuracy [%]	
	Mean BERT	Summed BERT	Mean BERT	Summed BERT
NN	82	82	80	78
Poly SVM	81	71	80	63
Linear SVM	78	79	80	78
Gaussian Process	70	50	70	50
SGD classifier	80	79	82	78

Overall, models trained with embeddings from transcriptions perform better than with embeddings from speech recordings. The quality of audio recordings is too poor to achieve a high level of accuracy.

5.3 Regression task

For the ADReSS regression task, various regression algorithms were tested in predicting the MMSE score for each subject. Accurate predictions can assist doctors in

Table 5.4: Average of CV accuracies and test for audio embeddings (K=6)

Classifier	Test accuracy [%]	Validation accuracy [%]
KNN, k=5	62	53
Gaussian Process	59	49
Linear SVM	57	56

making faster and more informed decisions. The performance of Lasso, Elastic Net and KNN regression was investigated. Different embeddings were used for this task with different approaches, as described in 5.1.

For the regression task, the training dataset was updated: one non-AD subject was removed from the set, as it doesn't have an MMSE score calculated (set to NaN), which is considered as invalid input data for MMSE score prediction.

Different hyperparameters settings were tested and did not affect each algorithm's performance much. The accuracy of each regression algorithm was analysed using metrics such as RMSE and R-squared value, as shown in the following tables (some of them are placed in chapter A (Attachments)):

- 5.5 (testing set) and 5.6 (validation set) for mean BERT embeddings.
- A.2 (testing set) and A.3 (validation set) for summed BERT embeddings.
- A.4 (testing set) and A.5 (validation set) for audio embeddings.

Experiments have shown that regression algorithms work better with mean BERT embeddings than with any others features. ElasticNet has outperformed the other algorithms on testing and validation sets, achieving a total R-squared value of 0.48 and RMSE of 4.35 on the testing set, while Lasso (with alpha set to 0.1) and KNN (with k set to 6) perform with R-squared values of 0.46 and 0.23 and RMSE of 4.41 and 5.31 respectively.

Table 5.5: Results for regression algorithms for mean BERT embeddings on testing set

Regressor	RMSE	r2 score AD	r2 score non-AD	Total r2
Lasso, alpha=0.1	4.41	0.18	-6.56	0.46
ElasticNet, alpha=0.1	**4.35**	**0.22**	**-6.63**	**0.48**
KNN, k=6	5.31	-0.84	-1.43	0.23

Table 5.6: Results for regression algorithms for mean BERT embeddings on validation set

Regressor	RMSE	r2 score AD	r2 score non-AD	Total r2
Lasso, alpha=0.1	5.77	-0.15	-102.83	0.34
ElasticNet, alpha=0.1	**5.37**	**-0.20**	**-73.88**	**0.42**
KNN, k=6	6.02	-1.65	-11.06	0.28

As can be seen in tables 5.5 and 5.6 all models fail to indicate non-AD class, which can be caused by an imbalance along the MMSE score in the dataset. The fact that each class has the same number of elements does not necessarily mean that the data is balanced: in this case, the non-AD class has a much narrower range of values for the MMSE score than the AD class. This means that the distribution of the MMSE scores in non-AD class is much more compact than in AD class. Balancing the training set (undersampling AD class) didn't help to solve this problem. Increasing the dataset can possibly solve it.

Same as for classification tasks, regression algorithms worked better with mean BERT embeddings. For summed BERT embeddings results were unstable with the same settings and algorithms: KNN performed better, but for the Lasso and ElasticNet algorithms, RMSE was much worse (see tables A.2 and A.3). Among all 3 features, experiments with audio embeddings showed the worse results (see tables A.4 and A.5).

In conclusion, experiments for regression task showed, that dataset for predicting MMSE is imbalanced and no tested model can correctly predict MMSE for non-AD class. RMSE which is lower than 5 is considered as a good result, as intervals usually have a length of 5 (except for the severe stage, where the MMSE score can be from 0 to 15) for each AD stage.

6 Comparative analysis

All found works vary in their approach and methodology, but all of them used different feature extractors to prepare origin data for classification and regression tasks. All results have shown that no model is robust enough to get high accuracy for only acoustic data and no feature extractor can solve this problem.

In this work, classification and regression tasks were conducted using a pretrained models to get audio and text embeddings and different algorithms on the ADReSS dataset.

Classification task

For the classification task, in this work, MLP algorithm trained on summed BERT embeddings (linguistic) has outperformed the baseline model and achieved one of the highest accuracies of 85%.

For acoustic features only KNN (k-2) has achieved an accuracy of 71% on the testing set, however, it's important to notice that the average test accuracy in the cross-validation test for this algorithm on acoustic features was only 62%.

Table 6.1: Results for classification task

Work	Features	Model	Test acc. [%]
baseline paper, [6]	acoustic	KNN (k=1)	57
	linguistic	LDA and RF	75
Haulcy & Glass, [7]	acoustic	KNN (k=1)	56
	linguistic	**SVM and RF**	**85**
Martinec & Senja, [8]	mix	Log. regression	77
this thesis	acoustic	KNN (k=2)	71
	linguistic	**MLP**	**85**

Regression task

The best result (RMSE of 4.35) in this thesis was achieved by using mean BERT embeddings and Elastic net. RMSE of 4.35 is the lowest value among all investigated works. Martinec and Senja have achieved the highest RMSE using the feature set of NUW, Bigram, Character 4-grams, Suffixes, POS tag, GRA features and SVM

Table 6.2: Results for regression task

Work	Features	Model	RMSE
baseline paper, [6]	acoustic	DT	6.14
	linguistic	DT	5.20
Haulcy & Glass, [7]	acoustic	KNN (k=1)	5.69
	linguistic	GradBoost	4.56
Martinec & Senja, [8]	mix	SVM	4.44
this thesis	acoustic	KNN (k=6)	6.90
	linguistic	**ElasticNet**	**4.35**

model, but it performed the worst out of the four best regression models in the cross-validation setting in their work [8].

7 Conclusion

This thesis has demonstrated the potential of using different models in detecting AD from spontaneous speech. It has shown the strong and weak sides of the ADReSS dataset and different approaches.

Through the exploration of the ADReSS 2020 Challenge dataset and the application of various techniques, baseline results were outperformed for both tasks.

Pretrained models were used for feature extraction: BERT to extract features from transcriptions and SV model based on TDNN for audio recordings.

For both tasks, algorithms with different settings were tested, as it's described in chapters 3 and 5. For the classification task, the best result (test accuracy of 85%) on the testing set was achieved using summed BERT features and MLP algorithm. For the regression task the best result (RMSE of 4.35) on the testing set was achieved using mean BERT features and ElasticNet algorithm. This thesis has demonstrated comparable results to investigated works (chapter 2), as it's shown in chapter 6.

There are a few ways to improve results for AD recognition:

- Using pauses length in subject speech as a feature.
- Summarising the meaning of the whole speech and analysing how much an individual actually has said.
- Detecting and analysing filler words (individuals with AD have a word-founding problem, which can increase the number of filler words in the speech).
- Using emotion classification as a part of speech analysis (AD can lead to difficulties in emotion control).
- Improving the quality of speech recordings and making it consistent for each subject.

It's certain that AI is able to capture subtle changes in speech patterns that are indicative of cognitive impairment, which provides a promising start for early diagnosis and intervention using AI.

References

[1] *Alzheimer's and Dementia* [online]. Alzheimer's Association [visited on 2023-02-28]. Available from: https://www.alz.org.

[2] MOLLOY, D. W. Screening for Mild Cognitive Impairment: Comparing the SMMSE and the ABCS. *Can J Psychiatry* [online]. 2005, vol. 50, no. 1, pp. 52–58 [visited on 2023-02-28]. Available from: https://journals.sagepub.com/doi/pdf/10.1177/070674370505000110.

[3] MACMILLAN, CARRIE. *Mild Cognitive Impairment: It's Not 'Normal' Aging* [online]. 2022. [visited on 2023-02-28]. Available from: https://www.yalemedicine.org/news/mild-cognitive-impairment.

[4] *The middle stage of dementia* [online]. London: Alzheimer's Society, 2021 [visited on 2023-02-28]. Available from: https://www.alzheimers.org.uk/about-dementia/symptoms-and-diagnosis/how-dementia-progresses/middle-stage-dementia.

[5] CUMMINGS, Louise. Describing the Cookie Theft picture: Sources of breakdown in Alzheimer's dementia. *Pragmatics and Society*. 2019, vol. 10, pp. 151–174. Available from DOI: 10.1075/ps.17011.cum.

[6] LUZ, Saturnino et al. Alzheimer's Dementia Recognition through Spontaneous Speech: The ADReSS Challenge. 2021. Available from DOI: 10.48550/arXiv.2004.06833.

[7] HAULCY, R'mani and James GLASS. Classifying Alzheimer's Disease Using Audio and Text-Based Representations of Speech. *Frontiers in Psychology*. 2021, vol. 11. ISSN 1664-1078. Available from DOI: 10.3389/fpsyg.2020.624137.

[8] MARTINC, Matej and Senja POLLAK. Tackling the ADReSS Challenge: A Multimodal Approach to the Automated Recognition of Alzheimer's Dementia. In: *Proc. Interspeech 2020*. 2020, pp. 2157–2161. Available from DOI: 10.21437/Interspeech.2020-2202.

[9] SHLENS, Jonathon. A Tutorial on Principal Component Analysis. *Educational*. 2014, vol. 51.

[10] SHARMA, Pranshu. *Understanding Transfer Learning for Deep Learning* [online]. 2021. [visited on 2023-02-28]. Available from: https://www.analyticsvidhya.com/blog/2021/10/understanding-transfer-learning-for-deep-learning/.

[11] DEVLIN, Jacob et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 2018.

[12] LIU, Yinhan et al. RoBERTa: A Robustly Optimized BERT Pretraining Approach. 2019.

[13] LAN, Zhenzhong et al. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. 2019.

[14] CLARK, Kevin et al. ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. 2020.

[15] SPEECHBRAIN. *SpeechBrain: A General-Purpose Speech Toolkit* [online]. [visited on 2022-10-01]. Available from: https://speechbrain.github.io/.

[16] WAIBEL, Alex et al. Phoneme recognition using time-delay neural networks. *IEEE Transactions on Acoustics, Speech, and Signal Processing*. 1989, vol. 37, no. 3, pp. 328–339.

[17] NAGRANI, Arsha, Joon Son CHUNG, and Andrew ZISSERMAN. VoxCeleb: A Large-Scale Speaker Identification Dataset. In: *Proc. Interspeech 2017*. 2017, pp. 2616–2620. Available from DOI: 10.21437/Interspeech.2017-950.

[18] CHUNG, Joon Son, Arsha NAGRANI, and Andrew ZISSERMAN. VoxCeleb2: Deep Speaker Recognition. In: *Proc. Interspeech 2018*. 2018, pp. 1086–1090. Available from DOI: 10.21437/Interspeech.2018-1929.

[19] ZHANG, Zhiqing et al. ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in Time Delay Neural Networks for Speech Recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*. 2020, vol. 28, pp. 2895–2904.

[20] WANG, Feng et al. Additive Margin Softmax for Face Verification. *IEEE Signal Processing Letters*. 2018, vol. 25, no. 7, pp. 926–930. Available from DOI: 10.1109/LSP.2018.2822810.

[21] NIELSEN, Michael A. *Neural Networks and Deep Learning* [online]. 2015. [visited on 2023-02-28]. Available from: http://neuralnetworksanddeeplearning.com.

[22] AHMADIAN, Sajad and Alireza KHANTEYMOORI. Training back propagation neural networks using asexual reproduction optimization. In: 2015. Available from DOI: 10.1109/IKT.2015.7288738.

[23] EBDEN, Mark. Gaussian Processes: A Quick Introduction. 2015. Available from DOI: 10.48550/arXiv.1505.02965.

[24] EVGENIOU, Theodoros and Massimiliano PONTIL. Support Vector Machines: Theory and Applications. In: 2001, vol. 2049, pp. 249–257. ISBN 978-3-540-42490-1. Available from DOI: 10.1007/3-540-44673-7_12.

[25] SCIKIT-LEARN. *Nearest Neighbors* [online]. [visited on 2023-02-01]. Available from: https://scikit-learn.org/stable/modules/neighbors.html.

[26] LEUNG, Kenneth. *Micro, Macro Weighted Averages of F1 Score, Clearly Explained* [online]. 2022. [visited on 2023-02-28]. Available from: https://towardsdatascience.com/micro-macro-weighted-averages-of-f1-score-clearly-explained-b603420b292f.

[27] TIBSHIRANI, Robert. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*. 1996, vol. 58, no. 1, pp. 267–288.

[28] ZOU, Hui and Trevor HASTIE. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*. 2005, vol. 67, no. 2, pp. 301–320.

[29] MOODY, James. *What does RMSE really mean?* [online]. 2019. [visited on 2023-02-28]. Available from: https://towardsdatascience.com/what-does-rmse-really-mean-806b65f2e48e.

[30] FERNANDO, JASON. *R-Squared: Definition, Calculation Formula, Uses, and Limitations* [online]. 2023. [visited on 2023-02-28]. Available from: https://www.investopedia.com/terms/r/r-squared.asp.

[31] WEI, Tien-Ning. *What Does Negative R-squared Mean?* [online]. 2021. [visited on 2023-02-28]. Available from: https://tnwei.github.io/posts/negative-r2/.

[32] LUZ, Saturnino et al. Detecting cognitive decline using speech only: The ADReSSo Challenge. *medRxiv*. 2021. Available from DOI: 10.1101/2021.03.24.21254263.

[33] MACWHINNEY, Brian. *The CHILDES project: Tools for analyzing talk*. 3rd. Lawrence Erlbaum Associates, 2000.

[34] PEDREGOSA, F. et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*. 2011, vol. 12, pp. 2825–2830.

[35] GLUHM, Shea et al. Cognitive Performance on the Mini-Mental State Examination and the Montreal Cognitive Assessment Across the Healthy Adult Lifespan. *Cognitive and Behavioral Neurology* [online]. 2013, vol. 26, no. 1, pp. 1–5 [visited on 2023-02-28]. Available from: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3638088/.

A Attachments

A.1 Top 3 results for audio embeddings (classification)

Table A.1: Top 3 results for audio embeddings (classification)

Classifier	Label	Precision	Recall	f1-score	Test acc. [%]	Valid acc. [%]
KNN K=2	non-AD	64	96	77	71	61
	AD	92	46	61		
KNN K=5	non-AD	62	54	58	60	61
	AD	59	67	63		
GP	non-AD	58	62	60	58	61
	AD	59	54	57		

A.2 Results for summed BERT embeddings (regression)

Table A.2: Results for summed BERT embeddings on testing set (regression)

Regressor	RMSE	r2 score AD	r2 score non-AD	Total r2
Lasso, alpha=0.1	7.87	-1.51	-24.02	-0.68
ElasticNet, alpha=0.1	7.73	-1.01	-28.44	-0.62
KNN, k=6	**4.98**	**-0.47**	**-3.01**	**0.32**

Table A.3: Results for summed BERT embeddings on validation set (regression)

Regressor	RMSE	r2 score AD	r2 score non-AD	Total r2
Lasso, alpha=0.1	6.90	-0.60	-150.35	0.05
ElasticNet, alpha=0.1	6.49	-0.39	-134.18	0.16
KNN, k=6	**6.21**	**-1.56**	**-30.81**	**0.23**

A.3 Results for audio embeddings (regression)

Table A.4: Results for audio embeddings on testing set (regression)

Regressor	RMSE	r2 score AD	r2 score non-AD	Total r2
Lasso, alpha=0.1	8.34	-1.59	-30.14	-0.89
ElasticNet, alpha=0.1	7.97	-1.24	-29.02	-0.73
KNN, k=6	**6.90**	**-0.95**	**-17.98**	**-0.29**

Table A.5: Results for audio embeddings on validation set (regression)

Regressor	RMSE	r2 score AD	r2 score non-AD	Total r2
Lasso, alpha=0.1	9.88	-1.72	-349.27	-0.92
ElasticNet, alpha=0.1	9.67	-1.55	-338.60	-0.84
KNN, k=6	**7.50**	**-1.20**	**-155.25**	**-0.11**

A.4 PCA: train and test sets of different embeddings

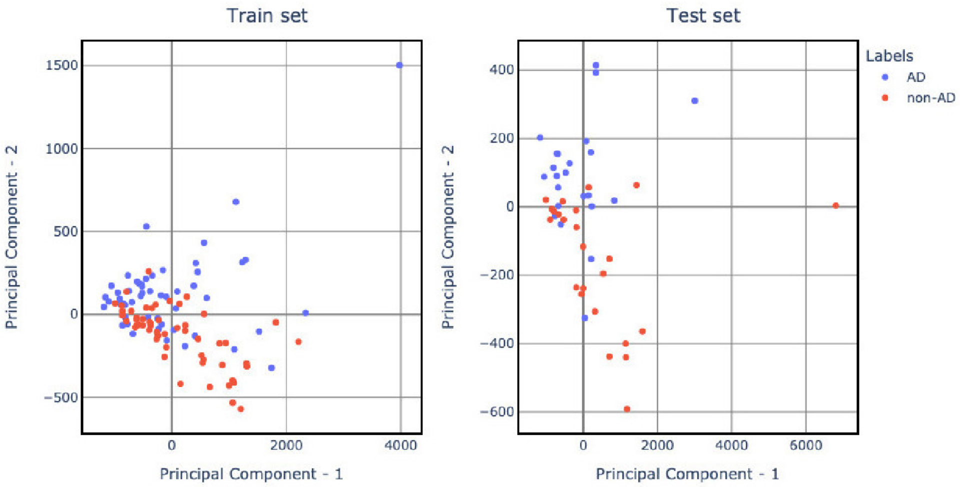

Figure A.1: PCA: train and test sets of summed BERT embeddings

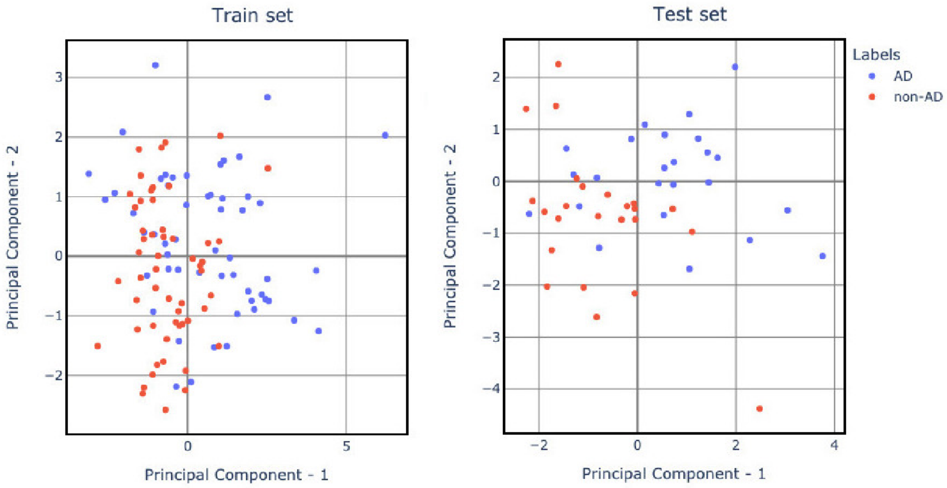

Figure A.2: PCA: train and test sets of mean BERT embeddings

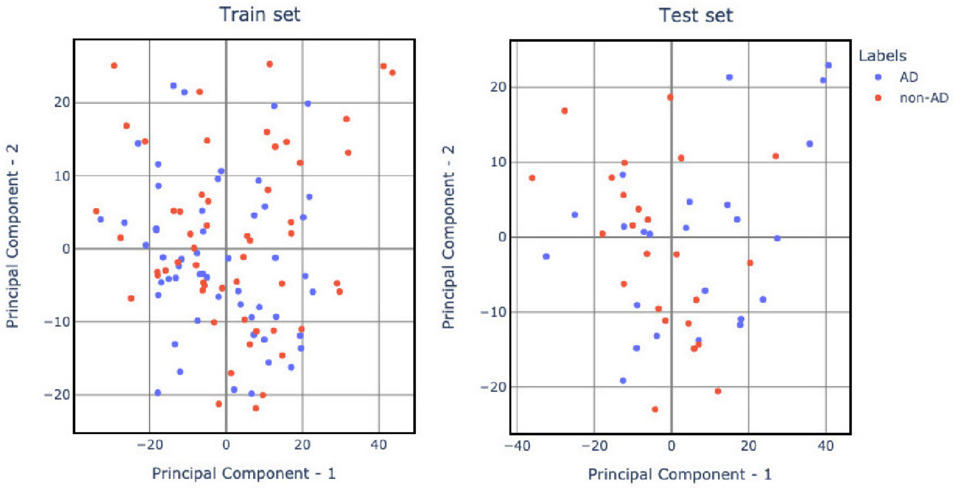

Figure A.3: PCA: train and test sets of audio embeddings

A.5 2-D scatter plots of actual and predicted values for different regression algorithms

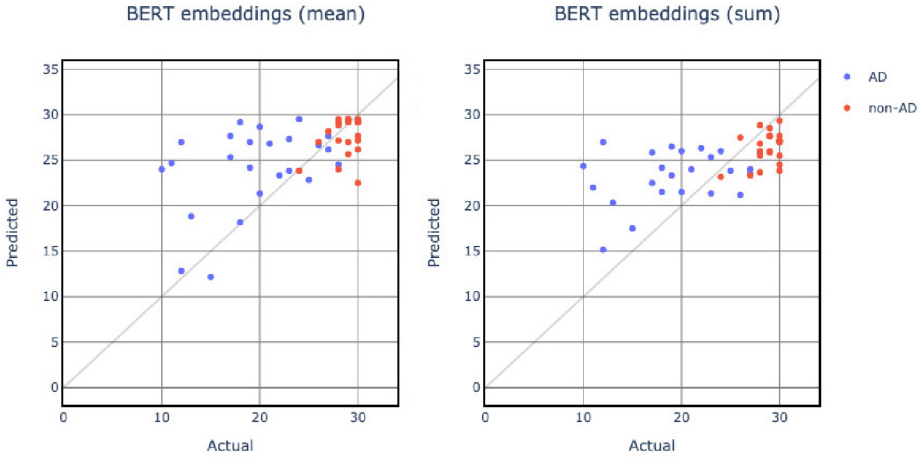

Figure A.4: KNN regression results for BERT embeddings

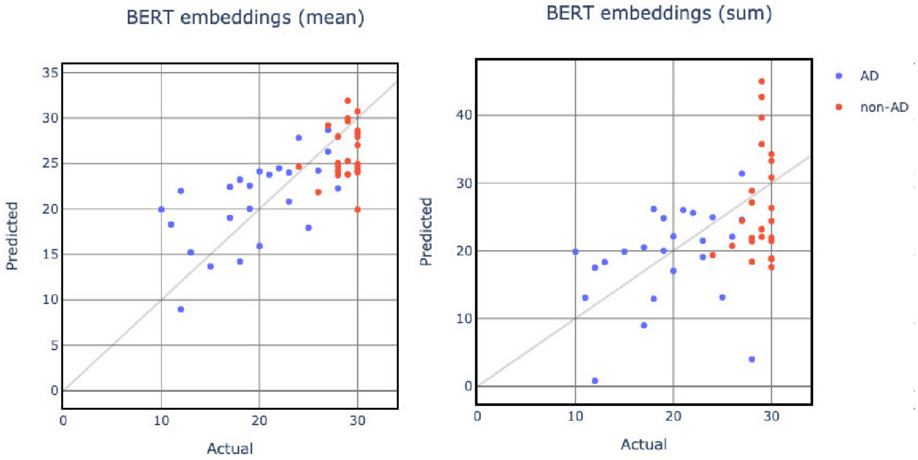

Figure A.5: ElasticNet regression results for BERT embeddings

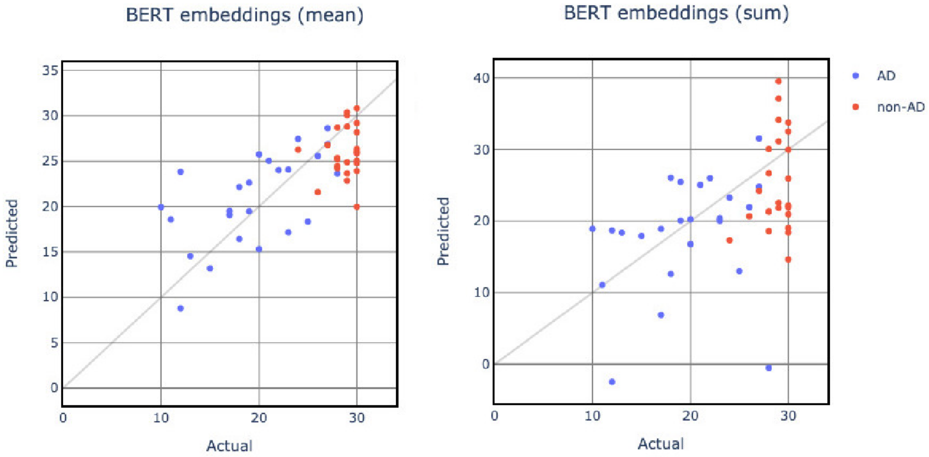

Figure A.6: Lasso regression results for BERT embeddings

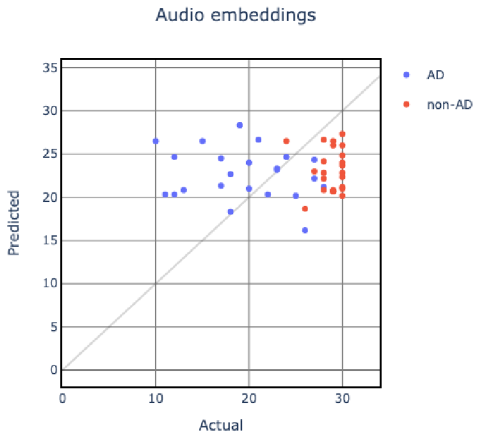

Figure A.7: KNN regression results for audio embeddings

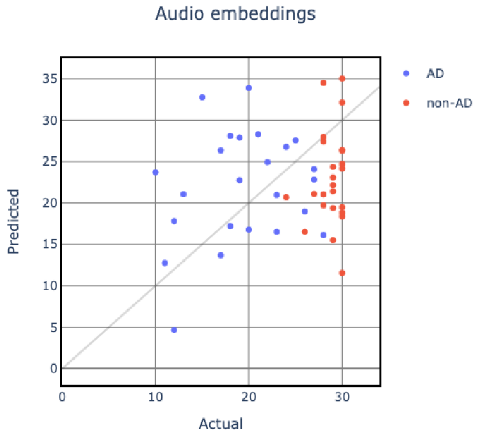

Figure A.8: ElasticNet regression results for audio embeddings

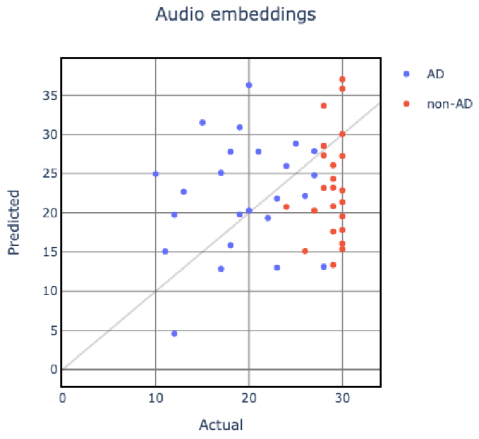

Figure A.9: Lasso regression results for audio embeddings