M.Sc. Thesis
Master of Science in Engineering

DTU Compute
Department of Applied Mathematics and Computer Science

Detecting Tremor in Parkinson's Disease using Multiple Instance Learning on Kinetic Data Collected from Wearable Device

Jiayi Han (s164229)

Kongens Lyngby 2022

DTU Compute
Department of Applied Mathematics and Computer Science
Technical University of Denmark

Matematiktorvet
Building 303B
2800 Kongens Lyngby, Denmark
Phone +45 4525 3031
compute@compute.dtu.dk
www.compute.dtu.dk

Summary

Parkinson's Disease (PD) is a progressive neurological disease that affecting 10 million people worldwide. The current golden standard for diagnosis relies heavily on the PD-subject's memory and subjective assessments in clinics. The diagnosis methods lack objectivity and suffer from inter-rater and intra-rater inconsistency. Automated symptom detection algorithms can provide objective and more accurate diagnosis of people with Parkinson's Disease, and is therefore, a highly desirable approach. Many studies have developed impressive Machine Learning algorithms that can detect motor symptoms using data collected from wearable devices. However, these studies are done under controlled clinical settings, which implicitly the methods lack everyday life resemblances.

In this study, we examined a method for automatically detecting tremor episodes related to PD. We used kinetic data collected during daily life, combined with Ecological Momentary Assessment (EMA) completed by the PD-subjects. The EMA was used as the ground truth for model training. However, it was not possible to acquire precise labelling of the data. Therefore, we applied the Attention Based Multiple Instance Learning algorithm, which is a Deep Learning approach for handling weakly labelled data. We evaluated the model's performance with two different data inputs i.e, the raw data and the frequency representation of the data. The two approaches achieved similar results, F1-score of 0.657 with raw data input and F1-score of 0.694 with frequency input. Furthermore, in collaboration with Paragit Solutions, we tested the models' performance on the data collected from their sleeve. The model trained on raw data showed best correlation with the PD-subject's diary.

We were challenged by the variability and complexity of the data. First, the labels that the model relies on are made by the PD-subjects who are not medically trained professionals. They were asked many questions in each EMA-questionnaire throughout the day. Therefore, they were not exclusively focused on their tremor experiences. Furthermore, there was a large variability in the tremor-severity among the participants and the recordings were mixed with their individual daily activities. Hence, this represents a need for more data, and a method to acquire more robust labelling to improve the models.

Preface

This thesis is completed in agreement with the requirements for acquiring a Master of Science degree in Engineering within the field of Mathematical Modelling and Computation at the Technical University of Denmark (DTU). It is made at the Digital Health department, DTU, in collaboration with Paragit Solutions.

Kongens Lyngby, March 21, 2022

Jiayi Han (s164229)

Acknowledgements

I would like to express my gratitude to my supervisors for their knowledge, positivity, and support during our weekly meetings. Your support have been crucial.

Especially, I would like to thank my main supervisor Sadasivan Puthusserypady for his encouragement and great help with any problem I faced during this project. A special thanks to my second supervisor Silvia Tolu, for always giving me valuable inputs. I am grateful for your comments, feedback, and ideas.

I would also like to thank Cihan Uyanik, for giving me valuable advice and setting aside your time during busy days.

I would like to thank everyone in Paragit Solutions, Mohammad Filfil, Mathias Milsø and Nickolai Rasmussen-Soto for many great discussions and in-depth answers to my many questions. Thank you all for giving me this opportunity to work on this interesting project.

And lastly thanks to Tariq Zamani for his help, humor, emotional support, and feedback.

Abbreviations

Adam Adaptive Moment estimation

CE Cross-Entropy

CNN Convolutional Neural Network

DFT Discrete Fourier Transform

DL Deep Learning

DTU Technical University of Denmark

FC Fully Connected

FN False negative

FP False positive

HPC High Performance Computing

IMU Inertial Measurement Unit

KNN K-Nearest Neighbours

lr Learning Rate

MDS-UPRDS The Movement Disorder Society - Unified Parkinson's Disease Rating Scale

MIL Multiple Instance Learning

MLP Multilayer Perceptron

NMF Non-Negative Matrix factorization

PD Parkinson's Disease

PSD Power Spectrum Density

RF Random Forest

RMSprop Root Mean Square Prop

ROI Region of Interest

SGD Stochastic Gradient Descent

SN Substantia Nigra

STFT Short Time Fourier Transform

SVM Support Vector Machine

TN True negative

TP True positive

Nomenclature

$\mathbf{z}$ The result of the permutation invariant aggregation function

α The slope of Leaky ReLU

f The transformation function (Parameterised by Feature Extraction module in the MIL model)

$g(z)$ Classification function (Parameterised by Classification Extraction module in the MIL model)

L Cross-Entropy Loss

μ Mean

$P'_{xx_i}(f)$ Estimated PSD at at the i'th window

$\bar{P}_{xx}(f)$ Estimated PSD

$q(H)$ Aggregation function

ρ Dropout rate

$\mathbf{s}(\theta)_i$ Softmax function

σ Standard Deviation

$\mathbf{V}$ Learnable weight in Attention module

$\mathbf{w}$ Learnable weight in Attention module

x Time domain signal input

y Ground truth class label

$\hat{y}$ The predicted class label

List of Figures

Contents

CHAPTER 1

Introduction

1.1 Motivation

Parkinson's Disease (PD) is the second most common neurological disease after Alzheimer's. The diagnosis is based on subjective clinical assessments. To provide efficient treatment and symptomatic relief, the medication dosage needs to be adjusted according to disease progress. Therefore, close monitoring and awareness of the PD-subjects is important. The intervals between clinical visits can last up to 3-6 months, and the symptomatic progression status highly depends on the PD-subjects' own memory and diaries. During clinical tests the PD-subjects are rated with MDS-UPDRS and/or Hoehn and Yahr scale. These clinical scales are subjective and suffers from inter-rater variability between medical professionals and intra-rater variability over time [1], [2].

Wearable devices have been widely studied during the last decade. The wearable devices are non-invasive and collect data continuously. The data-based approach is highly attractive for diagnosis and stores disease history that enables progression assessments. Automated and continuous monitoring systems can ease the burden of the PD-subjects as well as clinicians. Furthermore, provide objective and precise assessments frequently. Many studies have shown impressive results with Machine Learning models for symptom detection but the majority of them do not use data that resembles everyday life [3].

The aim of this thesis is to implement and test tremor detection techniques using data collected from Parkinson's Disease subjects' everyday life. It is expensive and very difficult to acquire ground truth labels for the data when it is not collected under professional surveillance. To overcome the challenge of weakly-labelled data, we will examine the Multiple Instance Learning model's ability for detection of tremor with coarse labelling on kinetic sensory data.

1.2 Thesis Overview

The work of this project is described in the next 8 chapters. In chapter 2, we introduce the physiological background of PD with focus on tremor and the current golden standard for diagnosis. In chapter 3, related work in the field of wearable devices and diagnosis of PD will be presented. In chapter 4, the data used in this project as well as prepossessing steps will be described. In chapter 5, we go into details of the model's structure and key methods applied for optimization. In chapter 6, we describe Paragit Solution's tremor detection algorithm and the data collected by their sleeve. In chapter 7 (Results), the performance of the model will be assessed. Finally in chapter 8, we discuss the results and the project as a whole. Lastly, a conclusion of the project will be given in chapter 9.

CHAPTER 2

Background

In this chapter we introduce the tremor in Parkinson's Disease from a clinical point of view as well as the known measures used for quantifying tremor. We will assess the current methods and challenges of diagnosing Parkinson's Disease. Lastly we will introduce the inertial measurement unit (IMU) and how it can help detecting tremor events.

2.1 Parkinson's Disease

Parkinson's Disease (PD) is one of the most common neurodegenerative disorder. The prevalence of PD is estimated to be around 10 million people worldwide [4]. Incidence and prevalence are increasing with age. Most of the diagnosed PD subjects are above 60 years old. The average age of developing PD is 60-65 in Denmark [5].

PD is a neurodegenerative disease which is characterized by the gradually decrease of dopamine producing neurons in Substantia Nigra (SN). SN is a structure located in the midbrain, illustrated in figure 2.1. Dopamine is a neurotransmitter (chemical substance) that is used for communication between neurons within the brain. In subjects with PD, 70-80% of the dopamine producing cells gradually degrade. The reason behind the degradation of neuron is unknown. These neuronal factors will result in worsen communication between neurons that are responsible for muscle movement and coordination. The motor symptoms related to PD and often used for diagnosis are tremor, bradykinesia (slowness of movement) and rigidity (stiffness) [6]. Medication such as Levodopa can be very effective to dampen the symptoms, but the correct dosage is essential. Long term use of medication can lead to dyskinesia. Dyskinesia is known as involuntary and uncontrollable movements. PD requires very individual assessments as it shows wide variability in clinical expression. The disease also evolves differently from person to person [7], [8]. In this project we are focusing on tremor and will go further into details in the following sections.

2.2 Tremor

Tremor is defined as involuntary and rhythmic contraction (shaking) of the muscles that determine a state oscillatory of all or a part of the body. 70% of all subjects diagnosed with PD experience tremor [8]. It is one of the most disabling and visible symptom among the cardinal symptoms in Parkinson's Disease. More than one-third of people in their early stage of PD experience tremor-related difficulty with numerous everyday tasks, which includes writing, using a type-writer/computer, fixing small things, dressing, eating, and holding reading material. This will lead to decreased quality of life [7], [10], [11]. Tremor occurs in relation to other diseases but PD is undoubtedly resting tremor (RT). RT is defined as; when the muscles are not voluntarily contracted while the limb is relaxed. Other types of tremors such as kinetic tremor (KT) occurs while the PD-subject is performing execution of voluntary movements such as writing, grasping an object or/and touching the tip of the noise. Postural tremor (PT) occurs when the PD-subject holds a limb in stationary position in the force of gravity such as holding the arm in front of the body. PT disappears when the limb is relaxed [10], [11]. Due to the oscillatory nature of tremor, many studies have used it's frequency characteristic for quantification. Table 2.1 provides some examples of different range of frequencies that have been used for quantifying tremor signals in

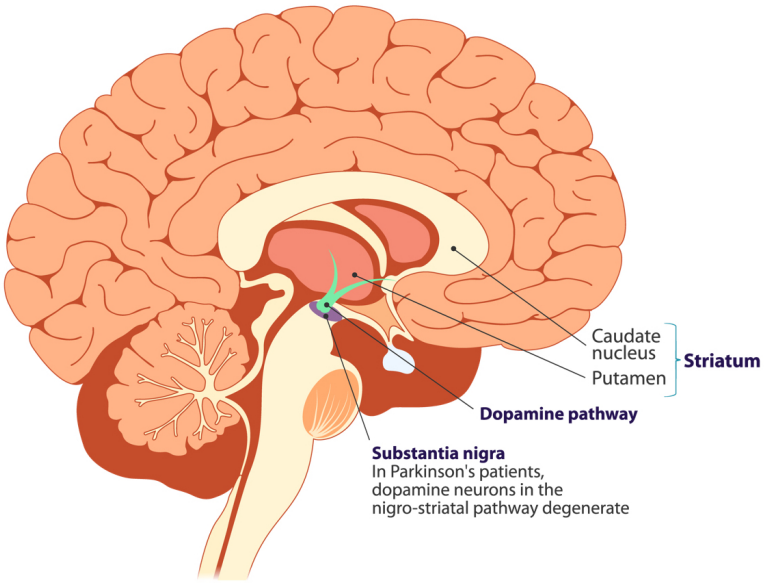


Figure 2.1: The brain region shown here is called the basal ganglia. The dopamine producing Substantia Nigra is located in the midbrain. SN has connection to Striatum [9].

different studies. Similarly, figure 2.2 shows accelerometer signals of typical rest tremor (left) and postural tremor (right) from a study done by Zhou et al. [12]. The sensors were placed on the index finger (IF) matacarpal phalangeal (MCP) joint, thumb (T) and one at the wrist (W). Linear acceleration is the combined signal from x, y, z direction. We can verify that the signal has the expected oscillatory behavior.

Table 2.1: Tremor frequencies.

Literature	Frequency
[13]	Low frequency tremor 6Hz, Medium frequency tremor 6 - 10 Hz. High frequency tremor over 10 Hz
[14] [15]	4-6 Hz
[16], [17]	3.5-7.5 Hz
[18]	3-8 Hz

2.3 Diagnosis and Assessments

Diagnosis of PD is based on the presence of 2 out of the 3 cardinal symptoms, namely tremor, bradykinesia and rigidity. Tremor is one of the most visible symptoms among the 3 cardinal symptoms. PD is a progressive disorder, therefore, the PD-subjects need to perform new tests for adjusting their medication. The duration between clinical visits can last up to 6-12 months [2]. In order to provide the neurologist better insight during clinical assessments it is very common to include PD-diary as part of the assessments. Moreover, a diary can help the neurologist monitor the PD-subjects' response to the new medication dosages. This process involves the PD-subject's own symptom awareness, engagement and sometimes help from family members.

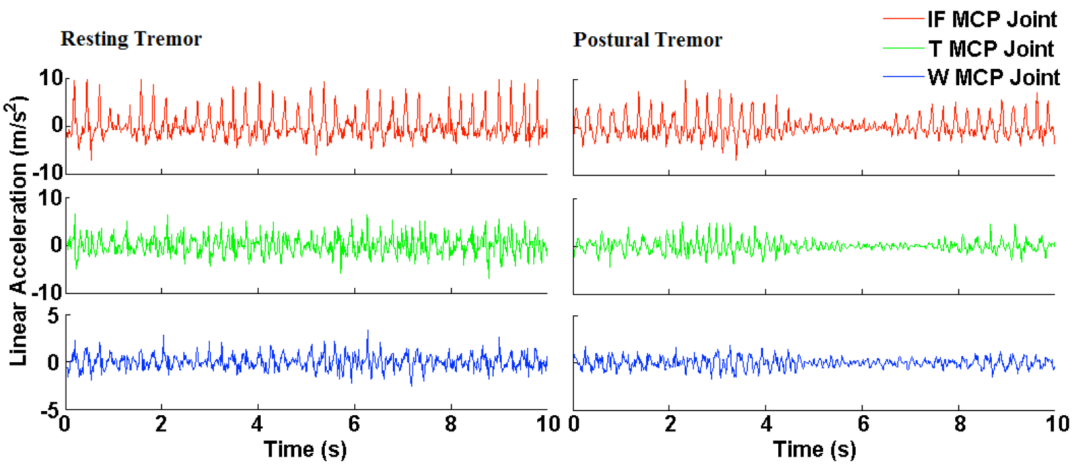


Figure 2.2: Accelerometer signals that shows tremor. Red: Measurement from the index finger (IF). Green: Measurement from the thumb (T). Blue: Measurement from the wrist (W). Source: [12].

The PD-subjects are evaluated during neurological visual examinations. During the test, the neurologist supervises the subject into performing specific motor tasks and gives them a score for each task. Very commonly used scales for disease assessment are the The Movement Disorder Society - Unified Parkinson's Disease Rating Scale (MDS-UPRDS) and/or Hoehn and Yahr (HY) scale. HY scale is a simple rating system from 1 to 5, used for scoring the PD-subject on the basis of pathological progress and level of disability. 1 represents no disease, and 5 indicates that the subject needs wheelchair or be bedridden unless assisted [19]. MDS-UPRDS consists of 65 questions distributed in 4 categories, (1) Non-motor experiences of daily living (2) Motor experiences of daily living (3) Motor examination (4) Motor complications. Each question is scored from 0 to 4, where 0 is normal and 4 is severe. Even though the MDS-UPDRS is used all over the world and is considered as the gold standard of PD diagnosis. Two out of four parts in the MDS-UPDRS rely on the PD-subject's memory, which can sometimes be inaccurate and affected by recall bias. The time between evaluation can last up to a few months, the examination is a snapshot of the subject's symptoms at the given moment. Furthermore, it has been shown that the MDS-UPDRS is very subjective, and can suffer from inter-rater variability and intra-rater variability over time [1], [2].

2.4 Inertial Measurement Unit

Wearable devices have been widely applied for detecting motor symptoms in PD in different studies with good results. The primary sensor in the devices is the Inertial Measurement Unit (IMU) that includes a accelerometer and a gyroscope which are highly correlated with UPDRS tremor ratings. Wearable devices play a crucial role for continuous monitoring of the disease progression while reducing cost that afflicts the healthcare system [2], [16], [20]–[23]. The IMU measures the acceleration $[\frac{m}{s^2}]$ and the angular velocity $\frac{degree}{s}$ in (x, y, z) direction. In some cases the acceleration is measured in $[g]$ $(9.8\frac{m}{s^2})$. Accelerometers are affected by the earth's gravity, meaning that there's a constant acceleration in the direction that is perpendicular to the ground. This artifact can be removed by using a high-pass filter. Different studies have shown that the IMU can be placed on various locations of the body, including smart clothes, watches, gloves and smartphones. IMU is

widely used to demonstrate the ability to detect PD-tremor, most common is wrist worn devices [18], [24]–[29]. The raw sensor signals are difficult to interpret, when recordings are longer and mixed with various activities, which is holding the healthcare professionals back from using the commercial devices for diagnosis and assessments [2]. A number of studies are researching different methods to quantify IMU signals for detecting motor symptoms in PD, usually they are based on the feature extraction from the signal from time and/or frequency domain and perform classification by using a supervised model e.g K-Nearest Neighbours (KNN), Support Vector Machine (SVM), Random Forest (RF) etc. [15], [30]–[32].

CHAPTER 3

Related Work

MDS-UPDRS is considered the current golden standard for diagnosis. It is used across the world but it lacks objectivity, comparability, and is sometimes inaccurate. Symptoms related to PD can occur randomly during every day activities, and not only during clinical visits, therefore using a continuous monitoring device will enable better diagnosis. Thus, the use of wearable devices for disease monitoring has become an increasing interest.

In previous years, large number of research has focused on using data and Machine Learning techniques to interpret data to increase objectivity and accurate classifications of the symptoms related to PD [33]. In this chapter, we will introduce the state-of-the art methods in the field of automated disease monitoring systems and their algorithms to interpret the data. Especially, we will focus on research related to tremor detection. The key words used for finding the materials discussed in this chapter are: *wearable AND parkinson AND tremor AND detect AND ("machine learning" OR "deep learning")* and written in English, within the last 5 years. From these search criteria, 35 papers were found, but only 19 of them were related to our study. Despite the fact that a lot of research has been done in the fields of classifying movements, PD etc. not many has done research on the basis of "in the wild data" as well as using PD-subject's own symptom assessments for classification, as shown in figure 3.1. All the studies used kinetic sensors (accelerometer, gyroscope and/or magnetometer), some papers used only accelerometer. Most of the data was collected during controlled setting, many papers applied Machine Learning for classification of tremor, while fewer studies used Neural Network structure (Deep learning/CNN/MLP).

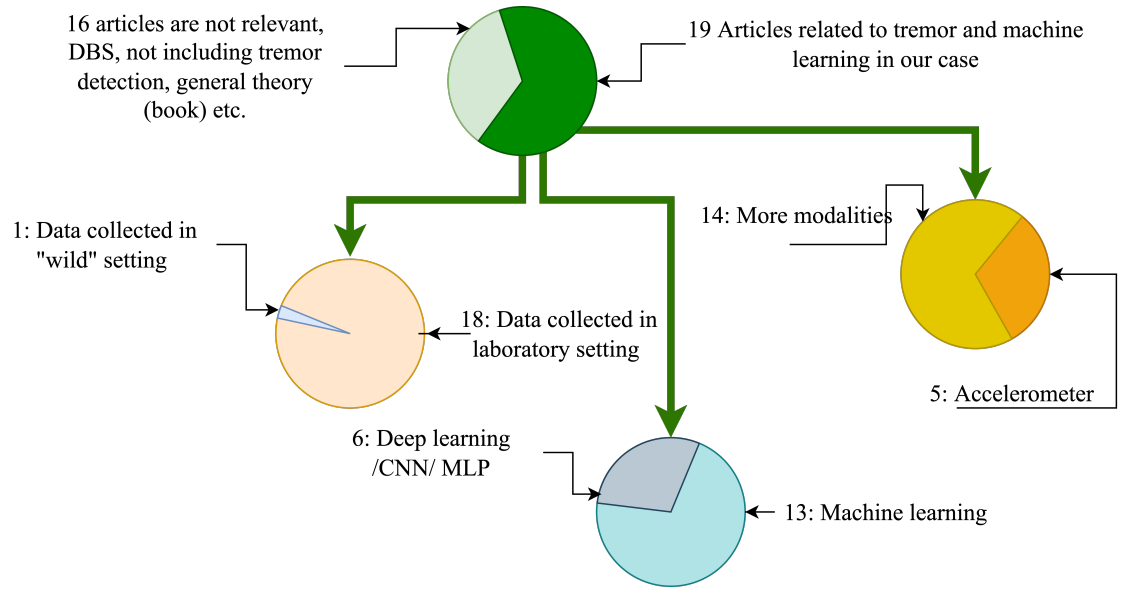


Figure 3.1: Through the initial keyword search, 35 articles were found. 19 were related to tremor and Machine Learning classification. Out of 19 papers of interest, only one study worked with data collected in wild setting.

3.1 In-The-wild Disease Monitoring Using Wearable device

Most of the analysis on PD has been performed in a clinical setting [15], [30], where large setups are required for detecting motor symptoms related to PD. Wearable devices can be used for at-home monitoring, but it is still a challenge to analyze due to the complexity of movements mixed with everyday life, and most of the research was primarily collected in controlled settings and validated by neurologists.

San-Segundo et al.[17] expected that, the data collected "in the wild" is a better resemblance of the tremor during normal living conditions. Whereas a standing drawback of using the data collected in the wild, is the labels are harder to gather. The correctness symptom classification algorithms are highly dependent on labeling, so many [34]–[36] used video recordings as reference to assign correct labels for tremor and no-tremor events. While the video recordings are applicable in laboratory settings, it cannot be done in PD-subjects' every day life. The label accuracy in study of San-Segundo et al.[17] was ensured by asking participants to record the scale of tremor within 5 minutes prior to submitting the label through an app. They rely on the PD-subject's self report measures. The app prompted the participants every hour and provided them with tree options (*Almost none, half the time and Almost always*). The three categories were then converted to a percentage score by assigning *<33%, 33–66%, and >66% tremor* to *Almost none, Half the time, and Almost always*, respectively. They experimented with using different types of data (laboratory data with correct labels, laboratory data with weak labels data and wild data with weak labels) as training set and evaluated the performance on wild data. Their result showed that, for most of the participants except one, the laboratory data with synthetically weak labels have highest correlation with wild data, while the actual wild data used in training was not performing as well.

3.2 Machine Learning for Assessing Parkinson's Disease Tremor

Machine Learning can be applied to many data modalities and helps to extract meaningful features within the data. With continuous monitoring systems, comes large amount of data that needs autonomous and systematic methods for analyzing the data. In this section we will address the machine learning algorithms that have been implemented in the literature for diagnosis of PD tremor. In the papers we have found, they were all using wearable kinematic sensors for data collection (accelerometer, gyroscope and/or magnetometer etc.). The data used in these research were sampled under controlled settings where the PD-subjects were asked to perform MDS-UPDRS related motor-tests or specific task such as resting, arm extending motion etc. which gave the data accurate labels. The most common models used were Random Forest (RF), K-Nearest Neighbours (KNN) and Support Vector Machine (SVM).

In a study done by Skaramagkas et al. [30], they used 4 sensors attached to different locations on the hand. The best performance was achieved by using SVM with 100% accuracy for detecting tremor in an extended arm position and object holding motions.

Channa et al.[15] investigated the detection of hand tremor as well as bradykinesia using Machine learning. The best result was achieved using KNN with 91.7% accuracy in detecting bradykinesia and tremor.

Hyoseon et al. [35] introduced an approach to predict the MDS-UPDRS score, where the highest result was achieved using decision tree classifier with the accuracy of 85.55%.

Zhang et al. [34] presented a number of different approaches for feature extraction. Through their research, their best performing model (AUC 0.887) was to use Convolutional Neural Network (CNN) to learn the features from Non-Negative Matrix Factorization (NMF) then use Multilayer Perceptron (MLP) to do the classification.

3.3 Using Weak Labels for Assessing Parkinson's Disease Tremor

Various techniques have been applied to analyze the data collected from wearable devices. As shown in figure 3.1, most of the data was collected in a controlled setting, which means the algorithms used for tremor classification and detection are mostly based on precises labels. Therefore, in this section we decided to include papers that were more relevant for us in terms of *"weakly labeled data"*. Following papers were related to our research [17], [21], [37]–[39].

In the studies done by Papadopoulos et al. [21], [39] they used data collected in the wild. They proposed an Attention Based Multiple Instance Learning (MIL) approach to assess the weak labels. This method is designed for dealing with data with coarse labelling. For this case, whenever a long segment of data (a bag) has one or more instances (short segment) of tremor, the bag is considered positive. The data was collected using smartphone devices with IMU sensors. The data quality was ensured by a group of signal processing experts through visual inspection of the signal. The collection was performed during phone calls while an app recorded the accelerometer signal. Usually, the phone calls last approximately 75 seconds, and the data is divided into 5 second window instances. Each instance of the signal is fed into the feature extraction CNN. Using the features, each segment is assigned a weighted attention value to identify the key instance within each bag. Their research showed good results with F1 score of 0.943 ± 0.034 [39].

The studies done by Zhang et al. and San-Segundo et al. [17], [37], [38] used the same data set. The weak labels were given as the percentage of tremor experienced within a time segment (e.g., 0-24%, 25-49%, 50-74%, 75-100%) [37] or (0-33%, 33-66%, and 66-100%) [17], [38].

Zhang et al. [37] analyzed the model's performance when the labels became less precise while data lengths were increased. Their findings suggest 10-minute length segments are short enough for algorithms to learn. The participants performed various tasks including every day tasks such as walking, playing chess, cards etc. The study compared many Multiple Instance (MI) approaches (MI-SVM, ID-APR, EM-DD and MI-NN). MI-NN had the lowest mean absolute error of detected tremor percentage. In a recent study by Zhang et al. [38], they showed that model training on wild data gives better performance on wild data than training on accurate labels from laboratory data. Additionally, they showed that person specific model leads to improved performance over a generic classifier.

San-Segundo et al.[17] used CNN-T/TN method, which included the extraction of tremor spectrum using NMF and then utilized the spectrum for each window as input into CNN for further feature extraction. Lastly, the classification is done using MLP (Multilayer Perceptron) with only 9% error, whereas [37] achieved 13% error predicting percentage of tremor on wild data.

CHAPTER 4

Materials

In this chapter we include a description of the kinetic data used for this study. We will start with the reason for using an online data set. Followed with data description and exploratory visualizations of the data collected by Habets et al. Lastly, we describe the preprocessing steps applied to the data.

4.1 Online Databases with Kinetic Recordings of Parkinson's' Disease Subjects

Initially the aim of this project is to use data collected from Paragit's own sleeve device, as a part of their clinical evaluation. Due to delay of collection and unprecise labelling we were unable to use the data for training of machine learning models. The logbook is attached in appendix A.3. The two main issues with the Parkinson's logbook is that: (1) The logbook is filled out by the PD-subjects on an hourly basis, which gives us very weak labels. This problem will require more knowledge about the collection process in order to narrow down the time-interval, such as video recording etc. but it is not available for this project. (2) The fields where the PD-subjects cross off, when they experience symptoms accounts for both tremor and bradykinesia which have opposite characteristics as bradykinesia is slow movements and tremor is oscillatory rhythmic contractions. Therefore, without any further information, it will be extremely difficult to use traditional symptom diary as the primary annotation for training a detection algorithm.

The challenges made us change the original plan and instead we will use more precise labelled data from online sources for model training and investigate the model's performance on Paragit's data. The search process is carried out with following requirements. (1) The wearable device should include IMU (ideally EMG). (2) The device should be attached at the PD-subject's upper extremities. (3) Each data annotation must only include one symptom and with clear time stamps. We resulted in finding three options online: Synapse, IEEE and DataVerse which is illustrated in table 4.1.

The data from IEEE [40] and Synapse [41] both utilized wearable sensors attached to the wrist and had clear UPDRS annotations, but since we are interested in data that was collected in the wild, we had decided to use data collected by Habets et al. [42], which also has the greatest resemblance to Paragit Solution's data compared to the data from other existing sources.

4.2 Data Collected by Habet et al.

The data for model training originates from a study by Habets et al. The data collection procedure was designed for examining the use of passive continuous disease monitoring in an at-home setting [42]–[44]. Reliability of the data is validated in the study by Habets et al [43], who showed that experience sampling can be used for modelling real-life Parkinson Disease monitoring. The participants were assigned 3 wearable devices, two for the upper extremities and one located on the chest. Each device contains accelerometer and gyroscope sensors, in total constitutes to 6 degrees of freedom. The accelerometer covered an amplitude range of$\pm$8 g and the gyroscope covered

Table 4.1: Overview of the online data sources with kinetic data collected from subjects with PD.

	IEEE [40]	**DataVerse** [42]	**Synapse**[41]
Type/Modality	Accelerometer	Accelerometer and gyroscope	Accelerometer
Label	UPDRS	Subjective Questionnaire	UPDRS (day 1 and 4) and Diary (day 2 and 3)
Data collection setting	In clinic	Everyday life	Specific tasks related to UPDRS and diary every 30 minutes during day 2 and 3.
Number of participants	17 PD and 17 control	20 PD	25 PD
Symptom assessed	resting tremor on /off	tremor, slowness and stiffness	Shimmer (bradykinesia and dyskinesia) GeneActiv (0-4 for tremor and presence and absence for dyskinesia and bradykinesia)
Advantages	Small data set easy to interpret and professionally labelled with UPDRS	Resembles Paragit's Data (Everyday life)	Large amount of data with more symptoms. Both clinically and subject assessments
Disadvantages	Only assessed one symptom	Not professionally annotated.	Specific tasks, large data with different sensors

a range of ±2000 degrees/s. The data was collected with a sampling rate of 200 Hz. Our study focuses on tremor, therefore we will only include data from the wrists. The PD-subjects were asked to wear the devices for 14 days. 2 out of 20 participant's data were not able to be converted, the rest of the project only include data from 18 participants. The demographics of the participants are shown in table 4.2. The PD-subjects were instructed not to change their normal behavior nor to perform any specific actions, which makes this data representative of free-living situation.

Table 4.2: Demographics of the participants.

Variable	Mean (Std) or Proportion (n)
Gender	4 F and 16 M
Age	63 (7)
Disease duration (years), mean (std)	8 (6)
levodopa equivalent daily dosage (mg)	770 +/- 394
Hohen and Yahr scale n	-
1	2
1.5	2
2	7
2.5	3
3	3
3.5	0
4	1
Presence Motor Fluctuations	12 yes and 8 no

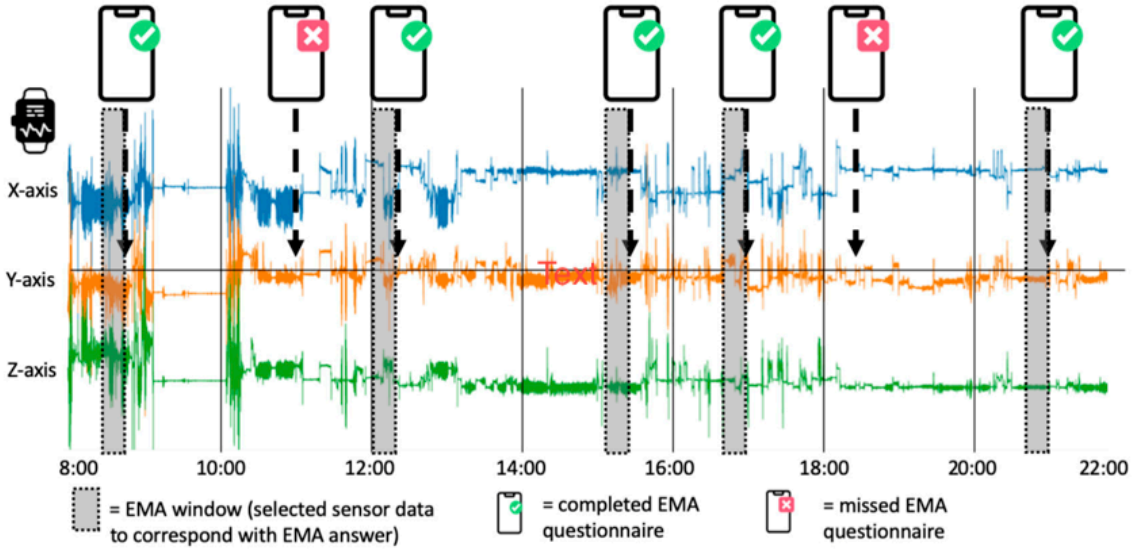


Figure 4.1: Overview over the data collection over a day. The smartphone notifies the participants with the EMA questionnaire. The answered EMA questionnaires will be compared with the corresponding accelerometer and gyroscope signals [42].

4.2.1 The Label and Class-Imbalance

The sensory data is combined with Ecological Momentary Assessments (EMA). EMA works as an electronic diary system that randomly presents the participants with a range of questions regarding mood states, contextual information and both motor and non-motor PD symptoms. The questionnaire is presented multiple times a day on the participant's phone (each 2-hour block between 8.00 and 22.00 h). The participants were instructed to answer according to their momentary experience. Thus EMA is superior compared to the standard diary because there is no recall bias. The questionnaire had to be opened within 15 min after notification to prevent procrastination. After the 15 min window, the questionnaire can no longer be opened and labeled as missed, as shown in figure 4.1. However, the window could potentially be a shorter in order to get better label precision, but it will need more investigation [44].

The complete questionnaire is attached in appendix A.4. The main annotation of interest is *"I experience tremor"*. The participant will rate the question from 1 to 7, where 7 is the most severe condition. The participants have been briefed about the meaning behind the scale from 1 to 7 but as the severity of disease progress is individual, there will potentially be bias between the ratings from different subjects. We have illustrated the distribution of the answers from the study in figure 4.2. As we can see, most participants experience 1 (least severe). Since we do not have the equal amount of labels for all classes (severity) we will binarize the labels, assuming that 1 means non-tremor events and above 1 are categorized as positive for tremor. The result of binarization is shown in figure 4.3, where we achieved more equal distribution of the classes (0 and 1), thereby reduces imbalance problem.

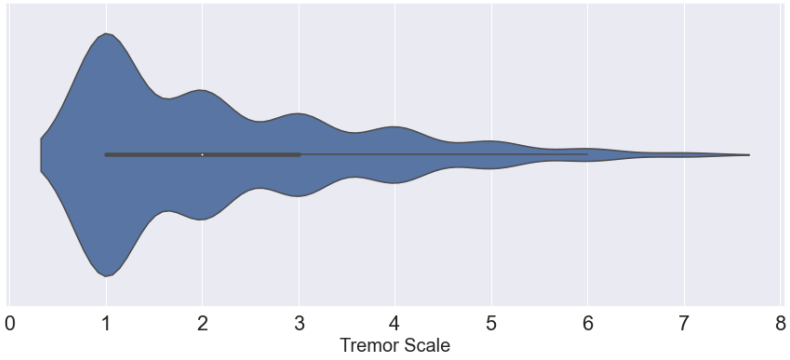


Figure 4.2: The distribution of answered tremor severity. The most frequent answer during EMA questionnaires was 1. The data has flooring effect and is unable to perform severity estimation, more appropriate approach is to use the data for binary classification.

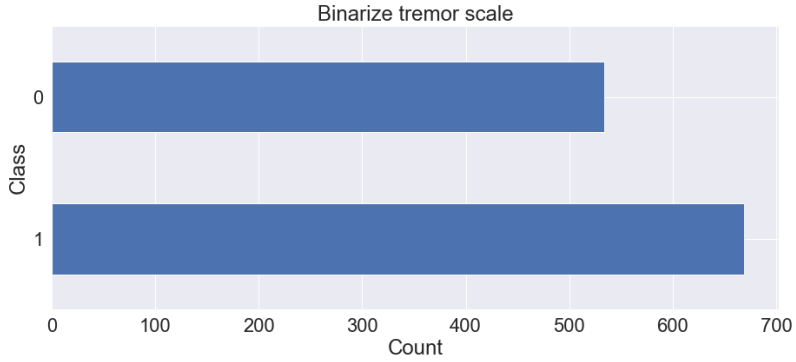


Figure 4.3: The distribution of classes, we see that the distribution of the severity of the tremor experienced were unevenly distributed, which lead to the decision of doing binary classification rather than severity estimate. The class 0 constitute 44.38% (534 observations) and class 1 55.61% (669 observations).

4.2.2 Diversity in the Data

Diving into further investigation of the data, using histograms shown in figure 4.4 and box plots shown in figure 4.5. We see that the participants are having very individual symptom progress during the data collection. Firstly, we see in figure 4.4 that some participants did not manage to

answer all the questionnaires, which might be due to their daily schedules or very severe symptoms that prevented them from answering the questionnaire within the available window of 15 minutes. From figure 4.5 we can see that many participants (110001, 110003, 110004, 110008, 110009 and 110010) had mostly no-tremor events. This means the data of class 0 mostly originates from them and might have an effect on the model's performance in classifying positive tremor from these participants and vice versa with negative tremor for other participants.

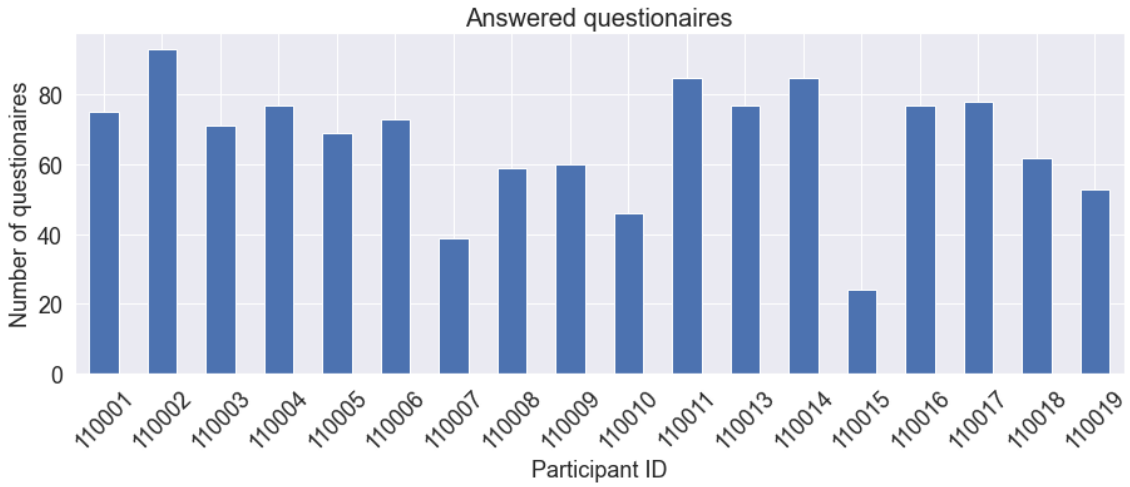


Figure 4.4: The answer from different participants are not equally distributed, participant ID 110015 has the lowest answer rate compared to the rest.

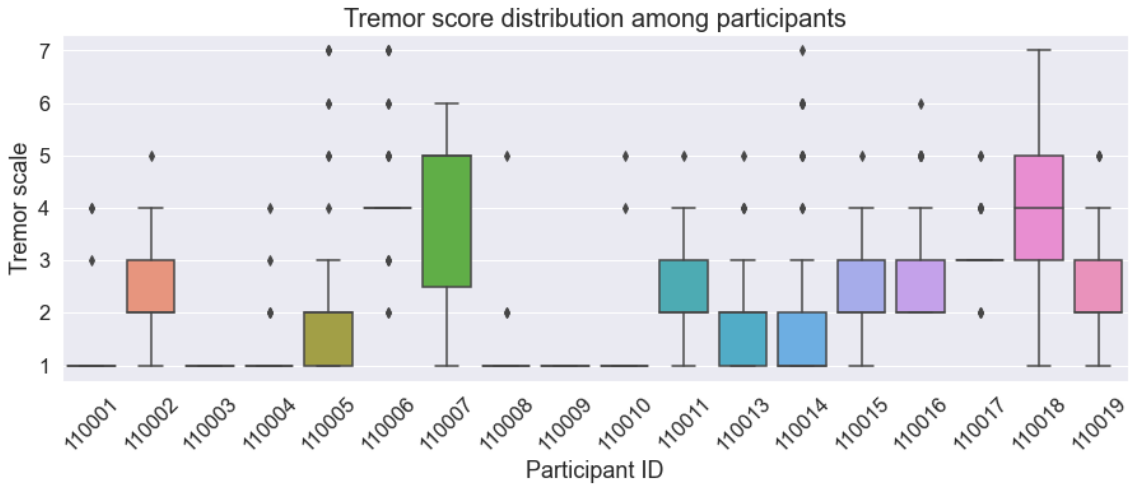


Figure 4.5: The participants experience different severity of tremor throughout the data collection period. Participants with ID 110001, 110003, 110004, 110008, 110009 and 110010 mostly experienced none-tremor episodes.

4.2.3 Frequency Representation of the Data

The frequency spectrum of the raw signal is of our interest due to the nature of tremor, therefore we use Short Time Fourier Transform (STFT) to visually inspect the data. STFT is similar to the traditional discrete Fourier transform, but the difference is that STFT enables us to assess the frequency in local sections and provides information on how the frequencies in the signal change over time. Figure 4.6 shows the STFT - signal of the accelerometer signal from one participant who experienced very severe tremor (7/7). The red horizontal lines outline the expected tremor spectrum (3.5-7.5Hz), as we can see, this region expresses the larges intensity throughout the data sequence of 15 min.

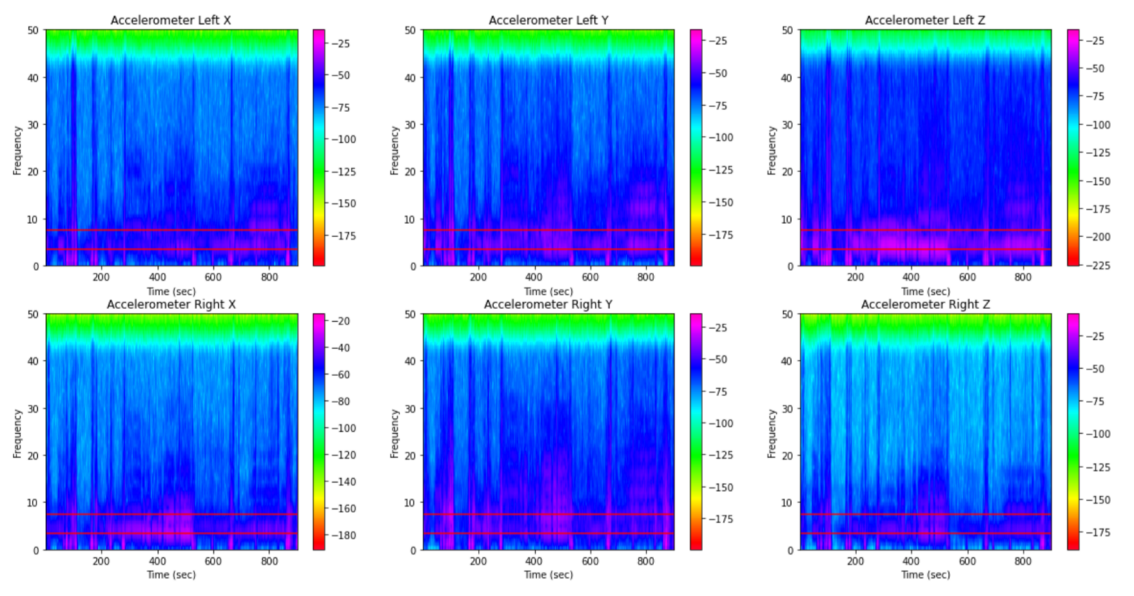


Figure 4.6: An example of the STFT visualization of the data. The spectrum of IMU signal from PD-subject who experienced tremor severity 7 out of 7. Top row: The data from left wrist. Bottom row: the Data from right wrist. The intensity is highest in the region of 3.5-7.5 Hz (marked with red lines).

4.3 Preprocessing

In the following section we will present the methods applied for processing the data. The purpose is to transform the data into a more suitable format for input in the MIL model.

4.3.1 Restructure the Data

The data from Habets et al [43] is organized in files belonging to each participant. Each file consists of 3-dimensional signal recordings in following structure

$$\text{Beep number} \quad \times \quad \text{data points} \quad \times \quad \text{sensor} \tag{4.1}$$

Beep number represents the number of successfully answered beep questionnaire. The **data points** is fixed to 90000 which is corresponding to the number of points for sampled during 900 seconds (15 minutes) with 100 Hz sampling frequency. The data originally has 18 **sensor** inputs where 6 of them are from devices located at the participant's chest, which is not interesting for

detecting tremor. We will only consider 12 of the sensor recordings from the right and the left wrist. The data is divided into separate files, such that each file only consists of one beep-answer. This way we will be able to shuffle the files for making the model training less participant dependent. Each file contains 90000 × 12 is then reshaped into 180 × 500 × 12 for dividing each 15 minute into 180 instances of 5 seconds.

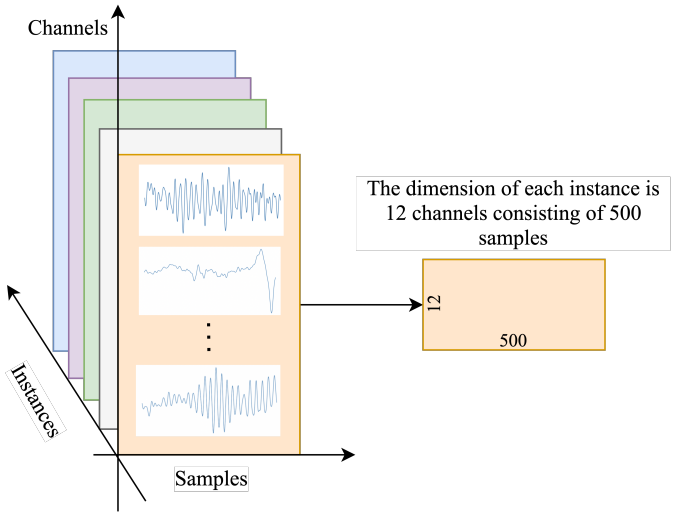


Figure 4.7: Each instance consist of 12 channel and 500 samples, in total there are 180 instances within each bag.

4.3.2 Band Pass Filtering

Filtering is a method to remove the undesired signal by only letting certain frequencies pass through the filter. A band pass filter is applied since we are interested in an interval of frequencies. Digital filters are categorized two as types of filters, namely FIR (Finite Impulse Response) or IIR (Infinite Impulse Response) filters. We are considering high frequency components as noise since human movements are expected to be low-frequent and the frequency band of interest lies between 3.5 Hz to 7.5 Hz. The gravitational component is a constant force which is around 0 Hz. In this project we applied 5th order Butterworth IIR filter with cut-off at 0.3 to 9 Hz.

4.3.3 Standardization

After investigating the accelerometer and gyroscope values, we have discovered large range-variations. The standard deviation of gyroscope is almost 20 while the standard deviation of accelerometer is 0.08, shown in table 4.3. The model will then focus more on the gyroscope values while the accelerometer information becomes less important. In general, standardization is a good practice in Deep Learning in order to achieve better performance and speed up the learning process. Another good reason to standardize the data in this project is that we are using two datasets, namely Habets et al. and Paragit Solution's, which are collected using different devices and their attachments on the forearm. Therefore, standardization becomes crucial. Z-score standardization is chosen for this project, it is given by (4.2)

$$\tilde{\mathbf{x}} = \frac{\mathbf{x} - \mu}{\sigma} \tag{4.2}$$

Where each data point x is subtracted by the mean μ and divided by the standard deviation σ. We are standardizing the accelerometer and gyroscope values separately. This way, we will not change the original movement pattern from the two different sensory inputs. The standardization

Table 4.3: The overall mean and standard deviation of accelerometer and gyroscope before standardization.

	mean ± standard deviation
Accelerometer	$6.90 \cdot 10^{-6} \pm 0.08$
Gyroscope	$1.44 \cdot 10^{-3} \pm 19.86$

procedure is applied to each instance of 15 minutes. The mean of the accelerometers is given by:

$$\mu_A = \frac{1}{N} \sum \mathbf{ACC} \tag{4.3}$$

Where N is the length of the signal. The standard deviation of accelerometers:

$$\sigma_A = \sqrt{\frac{\sum (\mathbf{ACC} - \mu_A)^2}{N}} \tag{4.4}$$

Where **ACC** denotes the 3 axial accelerometer data (x,y,z) including both the right (R) and left (L) side and N is the total number of samples in **ACC**

$$\mathbf{ACC} = [accR_x, accR_y, accR_z, accL_x, accL_y, accL_z] \tag{4.5}$$

Similarly the gyroscope mean and standard deviation are calculated using the same formula:

$$\mu_G = \frac{1}{N} \sum \mathbf{GYRO} \tag{4.6}$$

$$\sigma_G = \sqrt{\frac{\sum (\mathbf{GYRO} - \mu_G)^2}{N}} \tag{4.7}$$

Where **GYRO** denotes the 3 axial gyroscope data (x,y,z) including both the right (R) and the left (L) side:

$$\mathbf{GYRO} = [gyroR_x, gyroR_y, gyroR_z, gyroL_x, gyroL_y, gyroL_z] \tag{4.8}$$

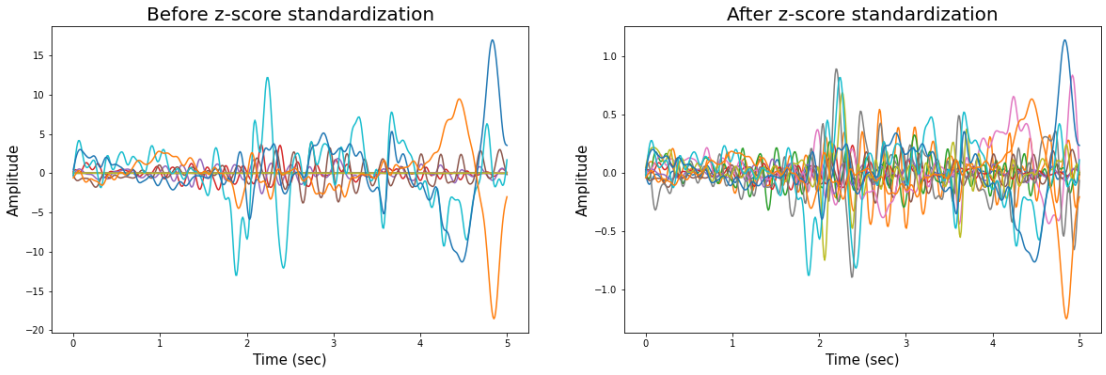


Figure 4.8: Before and after applying standardization. The signal before standardization is primarily dominated by the gyroscope data. After standardization we have added more even importance between gyroscope and accelerometer data.

4.3.4 Welch's Method

The power spectrum is found by using Welch's Method. Welch's method is an non-parametric method used for estimating the Power Spectrum Density (PSD) by rolling window over the signal [45]. Instead of taking the Discrete Fourier Transform (DFT) over the entire signal, the time domain signal will be divided into a finite number of overlapping windows. The overlapping window method prevents loss of information. For each of the segments the signal will be tapered and transformed using the DFT. The tapering operation will remove the edge effect. For each window, the PSD is computed as following:

$$P'_{xx_i}(f) = \frac{1}{MU}\left|\sum_{n=0}^{M-1} x_i[n]w[n]e^{-j2\pi fn}\right|^2, i = 0, 1, \ldots, K-1 \tag{4.9}$$

Where M is the length of each window. x_i is the data in time domain and w is the windowing function. U is a normalization factor for the power in the window function $U = \frac{1}{M}\sum_{n=0}^{M-1} w^2[n]$ [45]. The resulting power spectrum from Welch's method is the average of power spectrum from the windows.

$$\bar{P}_{xx}(f) = \frac{1}{K}\sum_{i=0}^{i=K-1} P'_{xx_i}(f) \tag{4.10}$$

The estimated PSD $\bar{P}_{xx}(f)$. $P'_{xx_i}(f)$ is the power spectrum estimation at the i'th data-segment, K denotes the number of windows. The estimated power density spectrum is smoother and easier to interpret compared to traditional DFT [46]. The PSD is computed for comparison between the model's performance on different types of data inputs. The PSD estimation is performed over a 3-second windows with overlap at 75%, over each 15 minute observations and extracted the frequencies of interest between 0 and 25 Hz, which results in a vector of 76 elements [21].

CHAPTER 5

Methods

This chapter includes the methods used for the model training process and where the tuning of model parameters is described. The focus will be on Multiple Instance Learning (MIL) and Deep Learning concepts that is relevant for this project and why we decide to use MIL. Firstly we will start by introducing the concepts of deep learning and convolutional neural network and how they are applied in the Attention Based MIL. We will describe the Attention Based MIL technique presented in [47]. Lastly, the final hyperparameters will be decided.

5.1 Multiple Instance Learning

The biggest challenge of working with healthcare data is the labelling. Often we are unable to obtain precise labelling due to the costs behind. Especially for Parkinson's disease, the symptoms do not necessarily occur during the clinical assessment with the neurologist (20-40 minutes). PD - tremor is a symptom that has a small window size (down to 3 seconds). Which means if we use traditional supervised learning models we are required to have labels that are equivalent to the time stamp of symptom outbreak which becomes the biggest challenge. By working with weakly labelled data, Multiple Instance Learning (MIL) comes in handy when Region Of Interest (ROI) is roughly given. The purpose of using MIL is to train the model to be able to predict bag-labels. Within MIL, each segment of data is considered as a bag of instances, each bag has a label which is weakly labeled. Weak label means that only a small section (an instance) of the labeled data is actually where the ROI lies.

There are different approaches to MIL. Some models are aiming to maximize the log-likelihood while some other are margin based, such as MI-SVM and Margin based models are not scaleable. MIL with neural network approach is demonstrated by [38] to show best result compared to (MI-SVM, ID-APR, EM-DD and MI-NN). The MIL (MIL)we will be applying is the Attention Based MIL introduced by Ilse at al. [47] and applied for PD tremor by Papadopoulos. et al. [39].

5.1.1 Attention Based MIL

The goal with MIL is to predict the bag label. Especially with medical data, we are interested in the key instances within a bag that contributes to the final prediction. With attention based MIL, we are able to interpret the results. The Attention Based MIL consists of 3 processes:
(1) f is a transformation of instances to a low-dimensional embedding

$$f : \mathbf{R}^N \rightarrow \mathbf{R}^M \tag{5.1}$$

(2) a permutation-invariant (symmetric) aggregation function (q) that produces a fixed-length representation

$$q : 2^{\mathbf{R}^M} \rightarrow \mathbf{R}^M \tag{5.2}$$

(3) a final transformation to the bag probability (g).

$$g : \mathbf{R}^M \rightarrow \hat{y} \tag{5.3}$$

The processes are parameters by using Neural Networks. The pipeline is visualized in figure 5.1. By using feature extraction with convolutional neural networks, we are transforming the instances

into low-dimensional embedding $f : \mathbf{R}^N \rightarrow \mathbf{R}^M$. The permutation-invariant aggregation function function $q(H)$ is given by:

$$\mathbf{z} = q(H) = \sum_{k=1}^{K} a_k \mathbf{h}_k \tag{5.4}$$

Where, for each instance k, we will calculate the attention a_k

$$a_k = \frac{\exp\left\{\mathbf{w}^\top \tanh\left(\mathbf{V}\mathbf{h}_k^\top\right)\right\}}{\sum_{j=1}^{K} \exp\left\{\mathbf{w}^\top \tanh\left(\mathbf{V}\mathbf{h}_j^\top\right)\right\}} \tag{5.5}$$

$\mathbf{w} \in R^{L\times 1}$ and $\mathbf{V} \in R^{L\times M}$ are learnable weights.
With $\mathbf{z}$ the final transformation to the bag probability $g(z)$ is performed using the last classification layer consisting of fully connected layers. We will go further into detail about each component of the model in following sections [39], [47], [48].

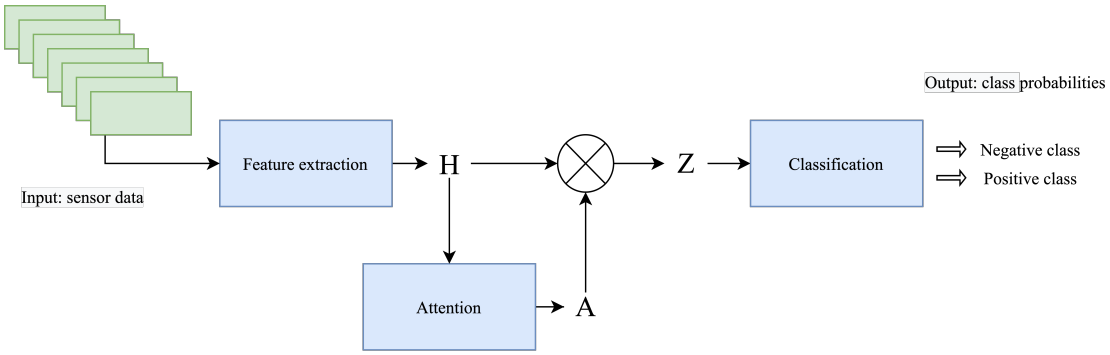


Figure 5.1: Overview of the architecture of Attention Based Multiple Instance Learning. The main components are the feature extraction, the attention and lastly the classification module.

5.1.2 Deep Learning

Deep Learning (DL) is a data driven technique that imitates human thinking process. The model learns the patterns in the data that is presented to it, in other words, the model learns by examples. The Neural Network has the ability to decompose the given data into feature maps in order to perform detection or classification. An illustration of a simple neural network is shown in figure 5.2. A network consist of an input layer, a number of hidden layers and lastly an output layer. The input of the hidden layer is the output of the previous layer.

The main processes of DL is the forward stage and the back propagation. The feature extraction process is carried out in the forward stage of model training with a series of non-linear operations in the network. The error between the prediction and ground truth is called the loss function. The optimal model is the model with the minimum loss. This is achieved by updating the internal weights and biases within the network by using the back-propagation. Back-propagation technique is used to update the weights using the partial derivatives of the loss function with respect to each parameter and reduces the loss for each iteration [49]. By combining the forward stage and the back-propagation, the model is able to train and adapt to the specific task. The purpose of applying DL for this project is to perform automatic feature detection of the kinetic signals as well as for classifying if the bag of signal contains any tremor.

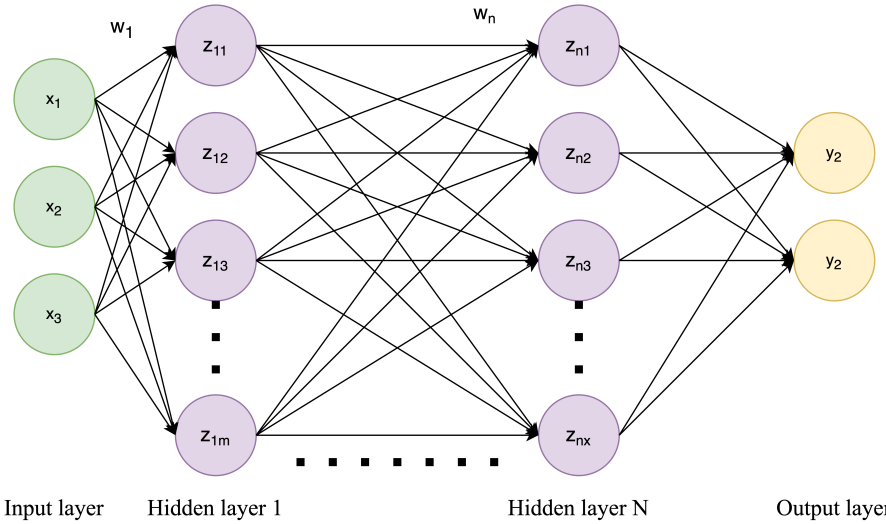


Figure 5.2: Deep Learning network. The network in general, is composed by many layers, where there is input layer, hidden layer and output layer. Usually there are a large number of hidden layers in deep learning.

5.1.3 Convolutional Neural Network

Convolutional Neural Network (CNN) is a type of DL structure that consists of alternating convolutional filters and pooling layers. The filters are good for pattern recognition and automatic feature extraction. The most common application of CNN is Natural Language Processing (NLP), image recognition and Computer Vision tasks [49], [50]. The convolution operation is done by taking the sliding sum of products over the input data. An illustration of convolution is shown in figure 5.3. The kernel values are multiplied element wise with the input data, then summed into one single value. The resulting output is the feature map. As shown here, the input is 2-D data and produces 1D feature map with 1D-kernel. The output dimension is dependent on the kernel size, padding and stride. Padding is by adding zeros to the data so edge observations are covered by the kernel. The Stride is the step size the kernel is moving. The resulting dimension after convolution is given by:

$$output = \frac{input - kernel_{size} + 2 \cdot padding}{stride} + 1 \tag{5.6}$$

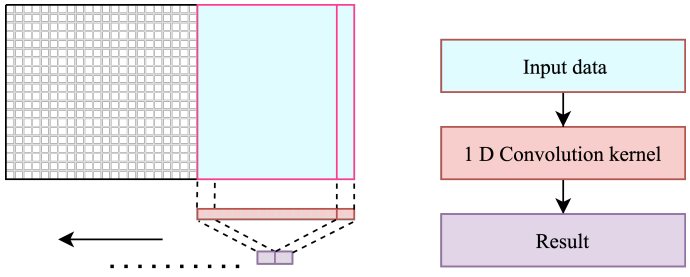


Figure 5.3: The result of 1D convolution on the 2D data is 1D.

The **feature extraction module** of the attention based MIL is shown in figure 5.4, This architecture is proposed by Papadopoulos et al. [39]. The network consists of convolution, activation

(Leaky-ReLu) and pooling layers (Max-pooling), lastly a flatten that transforms the data into 1-D array followed by fully connected layer. For this project, data is padded with 1 and stride is set to 1. The transformation in the first layer consist of 1D CNN with kernels with size 8 and 12

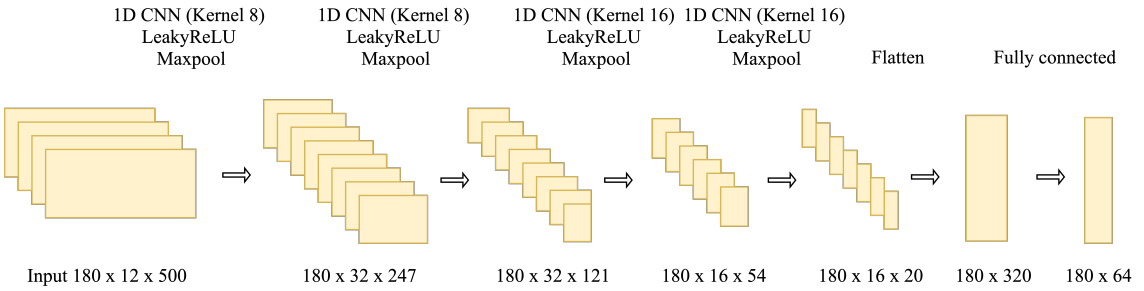


Figure 5.4: Feature extraction module. The sizes of this illustration are not representative of the original sizes. The feature extraction module consists of convolution, activation, pooling and fully connected layers.

input channels and 32 output channels, which means this layer contains 32 filters. The number of filters controls the depth of the output and simultaneously the number of features generated. Each intermediate dimension can be found by using the formula in (5.6). For example with the first layer:

$$\frac{500 - 8 + 2 \cdot 1}{1} + 1 = 495 \tag{5.7}$$

After Max-pool with kernel size of 2

$$\frac{495}{2} = 247.5 \rightarrow 247 \tag{5.8}$$

5.1.4 Fully Connected Layer

The fully connected (FC) layers, or a neural network that is fully connected means that every neuron in the network is connected to every neuron in the adjacent layer. It can also be referred to as dense layer. This connection is illustrated in figure 5.2. The FC network has a few drawbacks. (1) All input units have a separate weight to each output unit. For n inputs and m outputs the number of parameters required in this layer is $n * m$. Furthermore the bias term is added to each output, ending up with $(n+1)*m$. (2) Due to the individual weights, FC layers do not have spatial invariance as CNN. Thus, FC-layers are typically not applied solely for image or signal where we are interested in the temporal information but is usually applied after CNN for classification after the features already are extracted.

For the model structure with raw data inputs, the FC layer is applied as the last layer in the **feature extraction module** and as the main structure in the **classification module**. The classification module is shown in table 5.2, where it consist of 3 fully connected layers. The first layer takes the input of size M ($\mathbf{z} \in \mathcal{R}^M$) and outputs 32 units. The output of the first layer is the input of the second layer etc. By using these 3 layers, we are reducing the size to 1x2 and compute the probability of each class using the softmax activation, which produces output ranging from zero to one. Further details of the leaky-ReLu and dropout are provided in the next few sections.

The model that takes the frequency representation of the data as input is mainly composed by the fully connected layers. Both in the feature extraction and the classification modules. An overview of the model structure of both models are summarized in the table 5.1.

Table 5.1: Feature extraction module. Source: [21].

	Frequency data input	**Raw data input**
Input	1x76 spectrogram	12 x 500 kinetic data
Layer 1	Dense $76 \rightarrow 256$ Leaky-ReLu ($\alpha = 0.2$) Dropout $p = 0.5$	Conv 1D k=8, f=32 Leaky-ReLu ($\alpha = 0.2$) MaxPool k=2
Layer 2	Dense $256 \rightarrow 128$ Leaky-ReLu ($\alpha = 0.2$) Dropout $p = 0.5$	Conv 1D k=8, f=32 Leaky-ReLu ($\alpha = 0.2$) MaxPool k=2
Layer 3	Dense $128 \rightarrow$ M	Conv 1D k=16, f=16 Leaky-ReLu ($\alpha = 0.2$) MaxPool k=2
Layer 4		Conv 1D k=16, f=16 Leaky-ReLu ($\alpha = 0.2$) MaxPool k=2
Layer 5		Flatten Dense $320 \rightarrow$ M
Output	$\mathbf{h}_k \in \mathcal{R}^M$	$\mathbf{h}_k \in \mathcal{R}^M$

Table 5.2: Classification module consists of fully connected (dense), activation and dropout layer. Source: [21].

Input	$\mathbf{z} \in \mathcal{R}^M$
Layer 1	Dense M $\rightarrow$ 32 Leaky-ReLu ($\alpha = 0.2$) Dropout $\rho = 0.2$
Layer 2	Dense $32 \rightarrow 16$ Leaky-ReLu ($\alpha = 0.2$) Dropout $\rho = 0.2$
Layer 3	Dense $16 \rightarrow 2$ 2-way softmax
Output	$p(y \mid X)$

5.2 Parameters and Tuning

5.2.1 Activation Layer

The activation layer controls if a neuron should be activated or not and the amount of activation. The activation function adds the non-linearity to the network, thereby the neural networks are able to handle complex problems with less parameters. There exists a wide range of activation functions. The most common are rectified linear unit (ReLU) and leaky ReLU for CNNs. The functions are shown in figure 5.5. With **ReLU** the activation is equal to the input if it's positive, otherwise the activation will be zero. The downside of using ReLU is that it per definition will inactivate the neuron with negative output, as shown in the figure. This will cause the neuron's gradient will remain zero and not able to learn. The problem is solved by using **Leaky ReLU** instead it is possible to add a slope to the negative side of input [51], [52]. In this project, we will only be using leaky ReLU. The slope on leaky ReLU is a tuning parameter ($\alpha \in [0, 1]$).

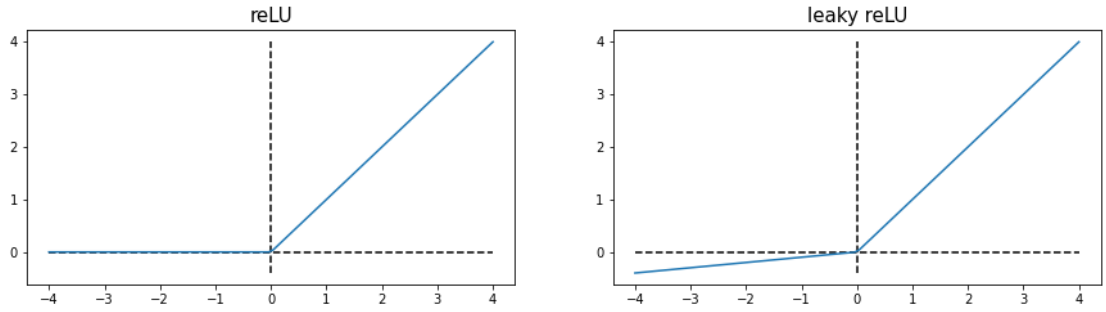


Figure 5.5: Rectified linear unit activation.

5.2.2 Softmax Layer

The softmax function $s(\theta)_i$ is also an activation function. The purpose of using softmax is for computing the normalized probability values. The output is between 0 and 1 and the sum of all the output probabilities is equal to 1. Softmax function is defined as following:

$$s(\theta)_i = \frac{e^{\theta_i}}{\sum_{j=1}^{C} e^{\theta_j}} \tag{5.9}$$

Where C is the number of classes, and θ_i is the input vector. The softmax is applied in the last step before the final classification output for finding the predicted class probability.

5.2.3 Max-Pooling Layer

Pooling layers are often applied after an activation layer. Two common pooling methods are average pooling and max-pooling. One of the disadvantages about CNN is that it is sensitive to the location of the features in the input. Typically the way to address the problem is to reduce the dimension of the feature map by using a pooling layer. Average pooling is done by taking the average over the patch on the feature map. Max-pooling is by taking the maximum value for each patch. An example of max-pooling with patch size of 2 is shown in figure 5.6.

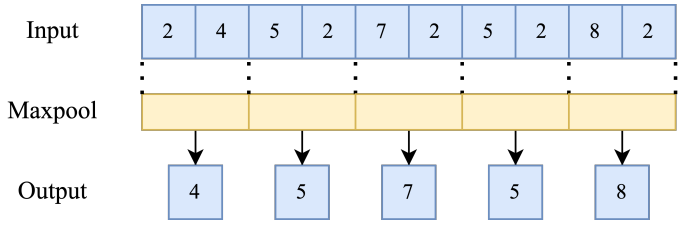


Figure 5.6: Max-pooling operation.

5.2.4 Dropout Layer

The dropout layer is very commonly applied after the activation functions. The basic idea of dropout is to remove individual activation at random while training the network thereby preventing the model from overfitting. This can make the model more robust and improve performance on test data. The dropout rate (ρ) is a tuning parameter that controls the percentage of weights that are dropped [53].

5.3 Training and Validation

Out of the 18 participants we were able to obtain 1203 bags (observations) of 15 minutes collected by Habets et al. The data is split into training + validation and test. The partition of the data is shown in figure 5.7. We are interested in using majority of the data on training and validation for improving the model's generalization properties. The data split is done by randomly selecting a subset of observations for testing (5%) while training and validation have 1143 bags. The partition of training and validation will be used for parameter tuning in a 5-fold cross validation. The data is divided into 5 folds, whereof one fold is saved for validation. Validation set is not used for training the model, but for evaluation of the model's ability to generalize performance on unseen data. K-fold Cross validation is an iterative procedure, and the training and validation is repeated K times. From Paragit's Solution's clinical trial we received data from 1 participant. This participant wore the sleeve for approximately 21 hours, which corresponds to 85 bags of 15 minutes each. This data will be used together with the testing set for the final performance assessment.

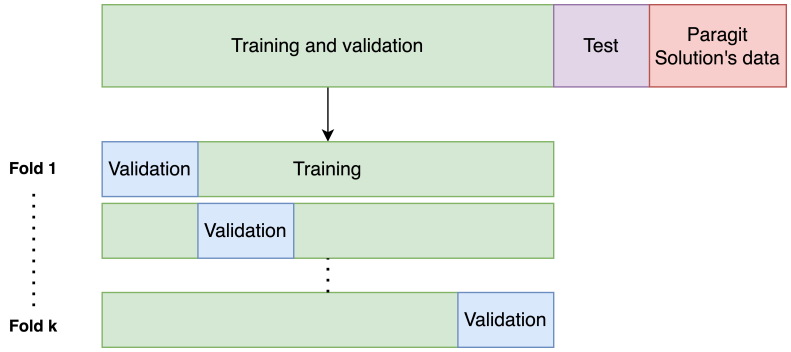


Figure 5.7: For the purpose of training, the data used for both training and validation is 95% of the total data while the test data is kept out of the loop and used for last evaluation.

5.4 High Performance Computing

Training the model requires extensive computational power. For this project we used the High Performance Computing clusters (HPC) provided by DTU Computing Center. The clusters have suitable hardware with Graphical Processing Unit (GPU) and memory storage that enables training of large models remotely, which has greatly improved the computational time for this project [54].

5.5 Hyper-Parameter Tuning Methods

The hyperparameter tuning is a large part of DL and ML since the model is very unlikely to work on different types of problems and data setups. Hyperparameter is highly crucial for the model's performance. The hyper parameters are generalized into 3 different categories. Model, optimizer and data hyperparameters. In this section we will introduce the methods that are used for finding the optimal parameters for our case and the different parameters and their effect on the model's performance

5.5.1 Grid Search and Random Search

Grid search is a very common method of finding the optimal parameters where a grid of hyper-parameters is prefixed by the user. This is a very expensive method and suffers from the curse of dimensionality, as the number of configurations is exponentially increased with the number of

hyperparameter. Thereby, with large models and many parameters, random search can be more preferable. Random search is similar to grid search, but searches in a randomized parameter-space. The randomized method can be effective when not all hyperparameter contribute to improved performance [55], [56].

5.5.2 Automated Hyper-Parameter Tuning Methods

Grid search and random search methods are very common but they do not take the previous results into account for next iteration, which leads to training on unnecessary parameter configurations. Manual tuning methods is shown in figure 5.8, which is very time consuming. It requires the user to evaluate the outcome of the previous models before executing the next training. A new suggested method is to use automated hyperparameter methods. Automated methods will ease the burden of fine tuning parameters and risk to miss out and find the optimal minimum of loss. Specifically we will introduce Optuna that is described by T. Akiba et al. [57]. Research showed that the automated method requires less time and no manual effort is necessary. It is therefore the preferred method when dealing with multiple hyperparameter problems such as in Neural Networks. **Optuna** is a Python Automated hyperparameter optimization framework. The processes behind Optuna are iterative between the sampling stage and pruning state. The procedure is illustrated in figure 5.9. Sampling is where the model decides the next set of hyperparameter from the defined search space by the user. The selection is based on gaussian optimization methods, where the optimizer configures the parameters that minimizes the objective function (loss function). Pruning (automated early stopping) procedure is designed for discarding a set of parameters by investigating the learning curves.

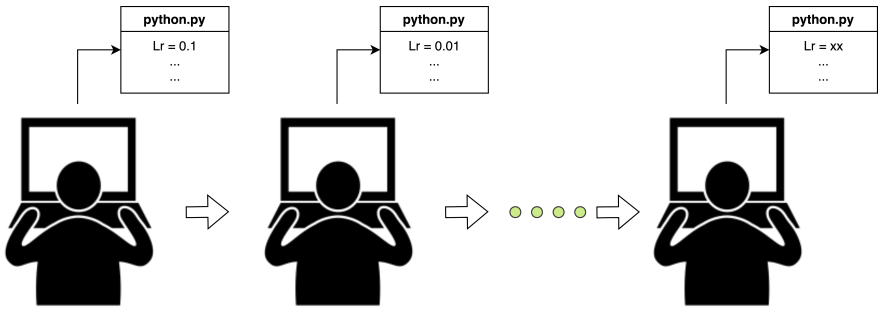


Figure 5.8: The manual optimizing process is done by evaluating the process of the results for a set of parameters and based on the results optimize and tune for the next set of parameters, which is a repeating cycle and very time consuming.

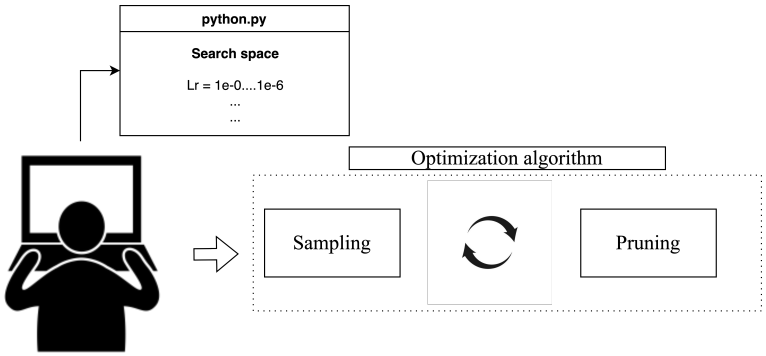


Figure 5.9: Automated hyperparameter search based on Bayesian methods.

5.5.3 Final Approach of Finding Optimal Hyper-Parameter

After a period of time where we experimented with different components in the model for finding the optimal hyperparameter, decided to use us a combined search. The downside of using 5-fold cross validation is that for each set of parameter configuration we will run the model 5-times. With our experimental run, it takes 2.8 hours to finish training of one set of parameters. Grid-search is very time consuming with many parameters, since it will train on some parameters that are less relevant. With random search we are similarly unable to reach any satisfying results within reasonable time.

Unfortunately hyperparameter tuning is not a linear process. Meaning one set of optimal parameters is not longer the best when the model structure parameters changes (e.g. applying new layer of dropout etc.). For our case the best we can do is to fix some parameters that seem to perform well and try to improve the performance with the a subset of parameters. With a combination of random and grid search, were we apply random search and experimenting different parameters, we will narrow down the possible set of optimal parameters and afterwards apply automated hyperparameter tuning on smaller parameter space. The reason behind we do not solely use automated search is due to the run-time limitation is maximum 24 hours with HPC LSF-cluster by DTU. This means that we are unable to fully rely on the automated algorithm since it will require more trails when we increase the parameter space. This means we are able to train 8 configurations on our model during 24 hours. Which is not optimal for evaluating the optimal hyperparameters.

5.6 Loss

A loss function is used for evaluating the learning process and for updating the weights in the network. Cross-Entropy (CE)is a commonly used loss function used for classification problems in Neural Networks. CE loss is applied in the paper our model is from [21], which we will also use in this project. It measures the difference between two probability distributions. If the CE is small, it suggests that two distributions are similar to each other. This score penalizes the probability based on the distance from the expected value. The total CE for each pass is used for back-propagation of the model weights. The equations below shows the CE for binary classification $y \in \{0, 1\}$:

$$L = -\sum_{i=1}^{c} y_i \log\left(\hat{y}_i\right) \tag{5.10}$$

The function takes the model's softmax output $\hat{y} \in [0, 1]$ and computes the CE L, with the ground truth $y = \{0, 1\}$. The optimal model will have a loss close to 0.

5.7 Tuning Parameters

5.7.1 Model and Data Parameters

The data parameters referrers to the different ways the data could be prepossessed. We have tried a few options, such as standardizing the signals individually by each axis and signal type. This gave the different signal sources equal importance thereby adding noise to the data. Therefore we ended up with using the standardization method described in the preprocessing section.

The possible tuning parameters inside the model's structure are the number of layers, the kernel size, dropout and activation as introduced previously. We are interested in movement features, which will involve longer segments of data, it makes sense to keep the suggested kernel-sizes of (8,8,16,16) in the 4 convolutional layers [21]. Dropout layers in theory prevent the model from overfitting, which is experimented in our search for optimal hyperparameters, by adding dropout

layers after each pooling layers in the feature extraction module. This is done by manually change the rate of dropouts and evaluate the outcome. The performance of model with added dropout layers is shown in figure 5.10. This is why we decided to add dropout layers after each max-pool layers of convolution segments in the feature detection module. Furthermore, we have adjusted the rates of dropout in the classification module. The final dropout rates are shown in table 5.3. The possibilities are endless, due to the limitation of time, we are unable to check for every type of added layers to the model's structure proposed by previous research [21].

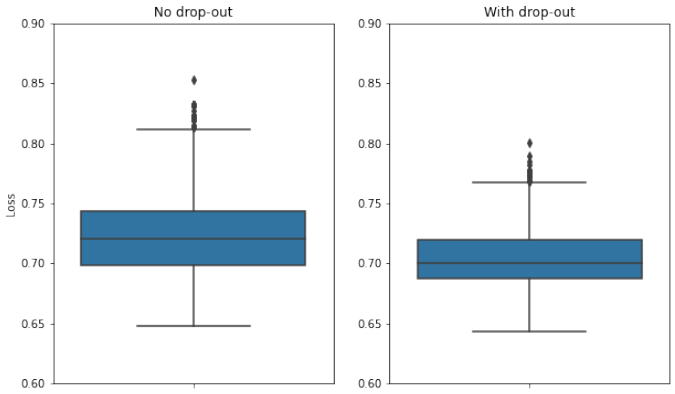


Figure 5.10: The performance of the model with dropout at $\rho = 0.1$ after each layer versus the original proposed model. It shows that adding some dropout will improve the performance slightly. The minimum validation loss are 0.648 without dropout and 0.643 with dropout.

Table 5.3: After random search, the best performing rate of dropout in the classification module.

	Raw model	Frequency model
Layer 1	Dense M $\rightarrow$ 32 Leaky-ReLu ($\alpha = 0.2$) Dropout $\rho = \mathbf{0.2}$	Dense M $\rightarrow$ 32 Leaky-ReLu ($\alpha = 0.2$) Dropout $t\rho = \mathbf{0.1}$
Layer 2	Dense 32 $\rightarrow$ 16 Leaky-ReLu ($\alpha = 0.2$) Dropout $\rho = \mathbf{0.5}$	Dense 32 $\rightarrow$ 16 Leaky-ReLu ($\alpha = 0.4$) Dropout $\rho = \mathbf{0.4}$
Layer 3	Dense 16 $\rightarrow$ 2 2-way softmax	Dense 16 $\rightarrow$ 2 2-way softmax
Output	$p(y \mid X)$	$p(y \mid X)$

5.7.2 Optimizer

Optimizer defines how the network learns. The first and most important parameter is the optimizer. The optimizer is the strategy for finding the the local minimum of the objective function (the loss). The stochastic multidimensional loss function / terrain is unpredictable. Therefore, with difference in model structure, data etc. will have influence on which optimizer is more suitable for the given problem. Some of the most popular optimizes in neural networks are Stochastic Gradient Descent (SGD), Adaptive Moment estimation (Adam), Root Mean Square Prop (RMSprop) and Adadelta.

The most intuitive method is the SGD, which is based on finding the gradient of the loss function

for each batch of data and updates the weights in the model. The learning rate defines the size of weight updates in each iteration. Proper increase in batch size can reduce the oscillations in the learning curve, since more data are contributing to the updates. Momentum can be added to SGD to increase the stability during learning, where some information from previous update is retained.

For the rest of the adaptive algorithms (Adam, RMSprop and Adadelta), the learning rate varies for each parameter. Adam, RMSprop and Adadelta uses exponential moving averages of squared past gradients to average out current gradient updates, thereby avoiding problem with vanishing gradient or/and exploding problem. In addition, Adam stores both the information about the past gradient and the past squared gradient (first and second order moments) [58]. Adam as the optimizer for the model proposed by [21] with learning rate lr= 0.005 and lr = 0.001 [21], [39]. RMSProp described by [59], is very similar to SGD with momentum, but it can restrict oscillations certain directions of parameter space, which in some case can converge faster. Adadelta restricts the previous saved gradients to a fixed number [60]. The adaptive methods are suitable for smaller datasets (under 10,000 samples) and works well with wide range of deep learning architectures [59]. As expected, Adam is the best performing optimizer for our case and we will fix this hyperparameter for simplifying the hyper-parameter search.

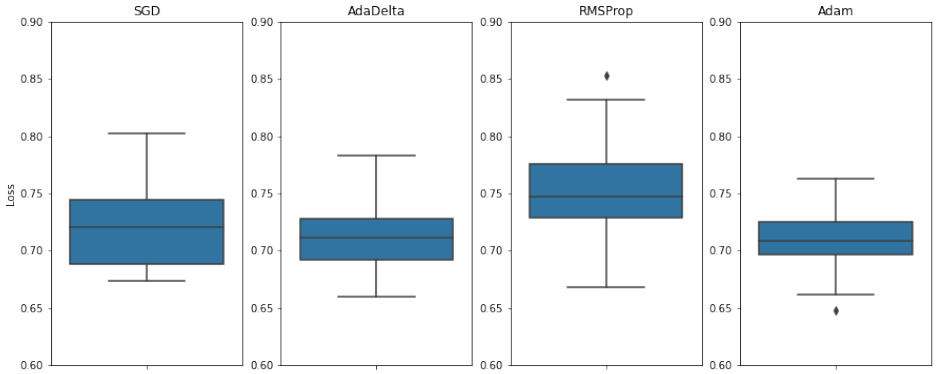


Figure 5.11: The Comparison between different optimizers with 5-fold cross validation. Here, the validation loss is illustrated in the box-plots. The optimizer that gave the minimum loss was Adam at 0.648.

5.7.3 Learning Rate

Learning rate is one of the most important hyperparameters. With all optimizers, the essential hyperparameter is the learning rate. Different choices of learning rate can have various effects on the loss curve. In a situation where the learning rate is too large, the example in figure 5.12 (right side) where the step size is dependent on the learning rate and the gradient. When the learning rate is too large it overshoots the minimum and causes oscillations. Figure 5.13 shows the loss functions with different learning rates. Low learning rate will take forever to converge towards minimum loss and too large of a learning rate tends to diverge the loss function. This means it jumps away and achieves worse solution. Good learning rate will take you smoothly down to the minimum loss. Slightly too large learning rates risk to fall a local minimum. The goal is by taking one step at the time and being able to move smoothly towards the minimum of the loss function. Learning rate is hard to tune on since a slight change in the number can have large influence on

the convergence. We want to both save time and able to find the minimum without wasting the time on unnecessary training (with grid search and random search).

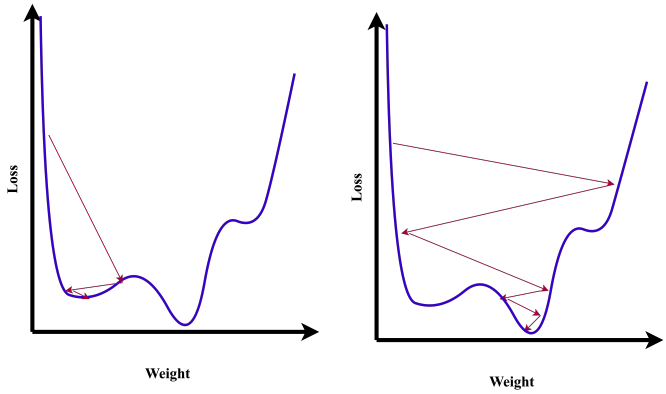


Figure 5.12: A learning rate that is not sufficiently large enough will end in a local minimum (left side). Too large learning rates can cause oscillations (right side).

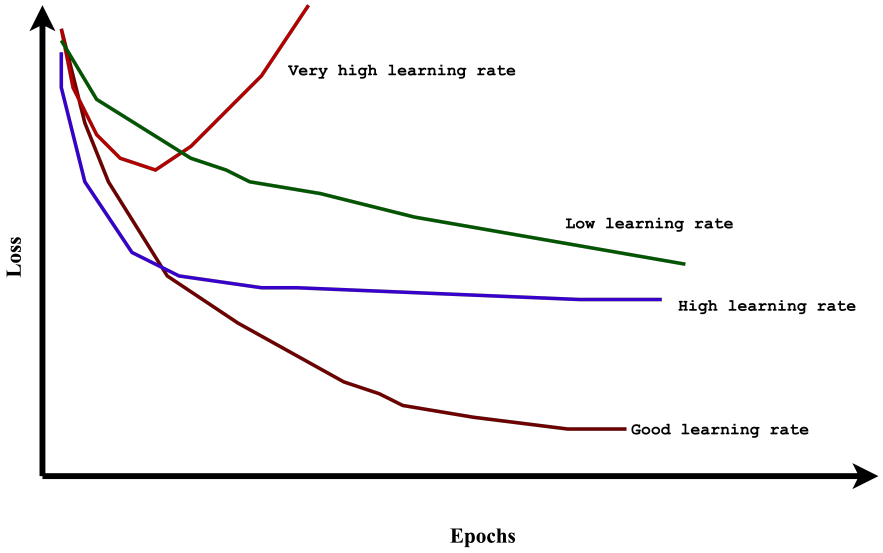


Figure 5.13: Different learning rates have great influence on the outcome of the loss function [61].

Therefore, our approach to find the minimum loss is by the use of automated hyperparameter search based on some fixed settings we have decided before, such as the dropout layer and the optimizer (Adam). The result of the automated hyperparameter tuning is displayed in table 5.4. We decide to use the learning rates 0.00016 and 0.00010 for training the model. The training and validation loss is shown in figure 5.14. It is clear that within 80 epochs the frequency model is not yet converged fully. Therefore the final run we will increase the epoch to 500. The raw model has a diverging validation loss. This means that we will apply early stopping e.g. stop the training of the raw model earlier in order to achieve best validation performance.

Table 5.4: With a large range, we found that the optimal learning range is in the area between $\{1e-4, 2e-4\}$ we then narrow down the search area. It takes shorter time for the model to train on frequency data, therefore we are able to run more trials within 24 hours.

Input data	Best Lr	Number of trials	Lr-range	Minimum loss
Raw	0.00017	8	$\{1e-4, 1e-5\}$	0.654
Frequency	0.00019	20	$\{1e-4, 1e-5\}$	0.643
Raw	0.00016	8	$\{1e-4, 2e-4\}$	0.648
Frequency	0.00010	20	$\{1e-4, 2e-4\}$	0.644

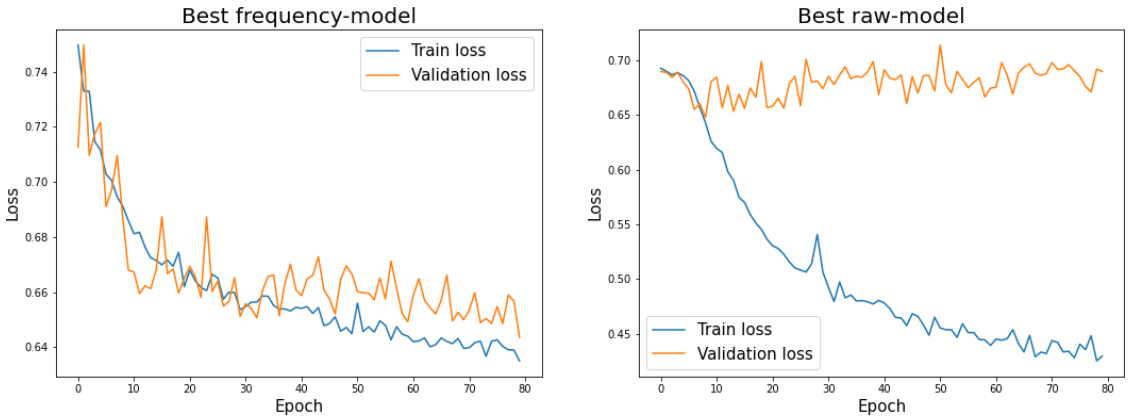


Figure 5.14: The left: The frequency model with the best learning rate. The right: The raw model with the best learning rate. The blue graph is the training loss and the orange graph is the validation loss.

5.8 Performance Evaluation

A model's performance is evaluated by looking at the predicted value and ground truth. There are various ways to evaluate the performance, in this section we will introduce F1-score. We have illustrated the different predicted outcomes in a binary classification problem in figure 5.15. On the right side (red) shown in red are the observations where ground truth is negative. On the left side (green) are the observations where the ground truth is positive. Inside the circle are the observations where the model predicts positive. The true positive (TP) and true negative (TN) are both the correctly classified classes. While false negative (FN) and false positive (FP) are the wrongly classified classes [62]. F1-score is a harmonic mean between recall and precision. Recall measures the observations that are correctly classified as positive (TP) out of all the observations that actual positive (TP and FN).

$$\text{Recall} = \frac{TP}{TP + FN} \tag{5.11}$$

Precision is a measure of the proportion of correct positive (TP) among all positive output (TP and FP).

$$\text{Precision} = \frac{TP}{TP + FP} \tag{5.12}$$

Finally, the F1-score:

$$\text{F}_1\text{-score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}, \quad \text{F}_1 \in [0, 1] \tag{5.13}$$

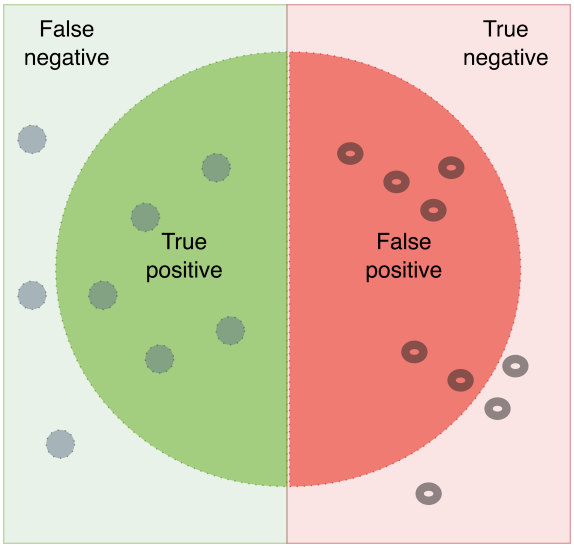


Figure 5.15: Illustration of different outcomes during binary classification. The left side shows the actual positive observations. The right side shows the actual negative observations. The observations within the circle are predicted positive.

F1-score is desirable since it combines two metrics and accounts for both FN and FP. The closer the F1-score is to 1, the better model it is.

CHAPTER 6

Paragit Solution's Data and Method

The data received from Paragit Solutions is from the PD-subject who has participated in the clinical trial of their sleeve device. The Clinical trial is done in collaboration with Odense University Hospital. Each of the PD-subjects were assigned with 2 sleeves and a diary. The diary is shown in figure 6.1 and 6.2. The PD-subjects are included in the study based on they are diagnosed with PD and excluded if they are diagnosed with Dementia. If the PD-subjects have Dementia, they will need help from caretakers which will complicate the experiment. The participants were not asked to stop medication while participating in this study. Only one subject will be included as part of the model testing due to delays and different diary qualities form the PD-subjects.

6.1 Devices

6.1.1 Sleeve

The sleeve, built by Paragit Solutions, is a wearable medical device for 24 hour monitoring of the PD-subjects. The sleeve has sensors that can detect Electromyography (EMG) and also kinetic movement signals including gyroscope and accelerometer. The EMG is located above the flexion muscle group of the under-arm, which enables muscle activity measurements from all hand and finger movements [24], [63]. The sampling rate of IMU is 500Hz and EMG 2000Hz. The range of the 3-axial IMU is $\pm 4g$. The data transfer is done using micro SD-card.

6.1.2 The Diary

The diary (figure 6.1 and 6.2) contains hourly symptom assessments during approximately 24 hours. The pink section represents the area where the PD-subject should cross off if he/she experience dyskinesia or hyperkinesia e.g. involuntary movement or loss of movement. The yellow area indicates normal movement. The blue area should be crossed off when the subject experiences Parkinson's Disease Symptoms (including bradykinesia and tremor). Both the blue and pink section have 3 severity levels, the fields that are furthest away form the normal (yellow) indicate the highest severity. We can see the participant do not experience dyskinesia or hyperkinesia, while he/she is very affected by PD-symptoms during the day.

6.2 Paragit's Current Tremor Detection Algorithm

The complete architecture is attached in section A.6. The tremor detection algorithm is similar to MDS-UPDRS, which is scale between 0 and 4, rating the severity of the PD-tremor. The algorithm uses the accelerometer data for detection. The 3 axial signal is concatenated by taking the magnitude of the signal $\sqrt{x^2 + y^2 + z^2}$. After prepossessing the data, by high-pass filtering, the data is divided into 5 second sub windows. For each of the 5 second window the signal is converted into the frequency domain.

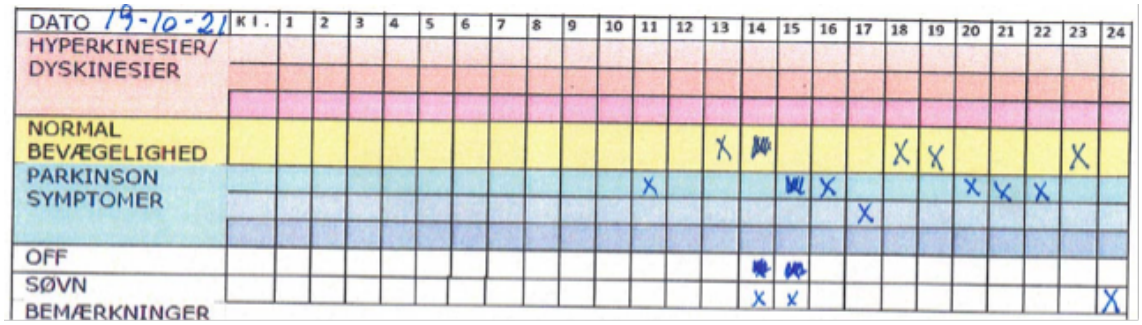


Figure 6.1: First half of diary: The pink region indicates hyperkinesia or dyskinesia. Yellow indicates normal movement. Blue indicates Parkinson's symptoms. White indicate sleep. The participant experienced symptoms around 11, 16, 17 and 20 to 22 o'clock.

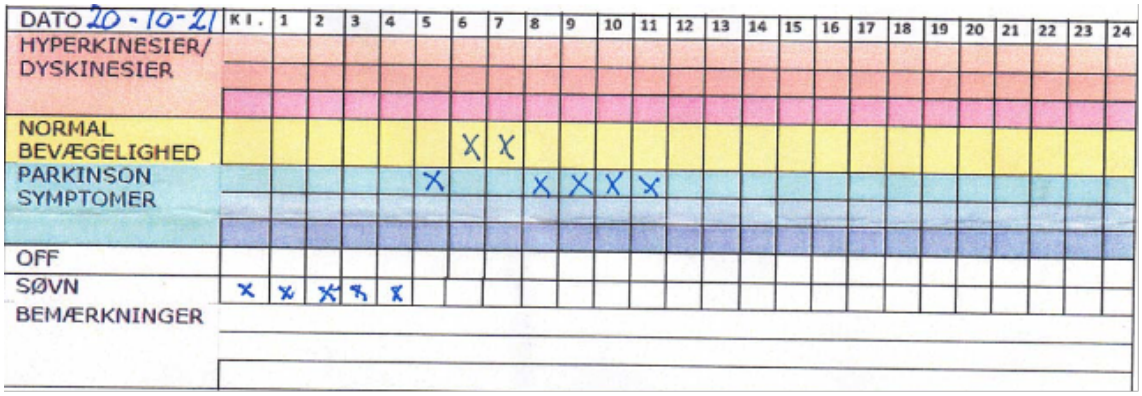


Figure 6.2: Second half of diary: The pink region indicates hyperkinesia or dyskinesia. Yellow indicates normal movement. Blue indicates Parkinson's symptoms. White indicate sleep. The participant experienced symptoms around 5 and 8 to 11 o'clock.

With the frequency representation of the smaller sub-window data, they used two thresholds for classification. (1) The amplitude of the dominant frequency within 3.5-7Hz should exceed 20, then the sub-window will be classified as positive for tremor. (2) The ratio of positive sub-windows within 5,10 or 15 minute segment should be more than 10%, then the whole segment is considered positive for tremor.

6.2.1 The Result of the Classification Algorithm

The diary showed that during 17:00 the PD-subject feels severe PD symptoms, and the algorithm detected tremor on the left side 6.4 approximately in the same time interval. In addition, there is a big difference in how prominent the tremor is on the two arms, where the left arm is clearly most dominant (see figure 6.3 and 6.4).

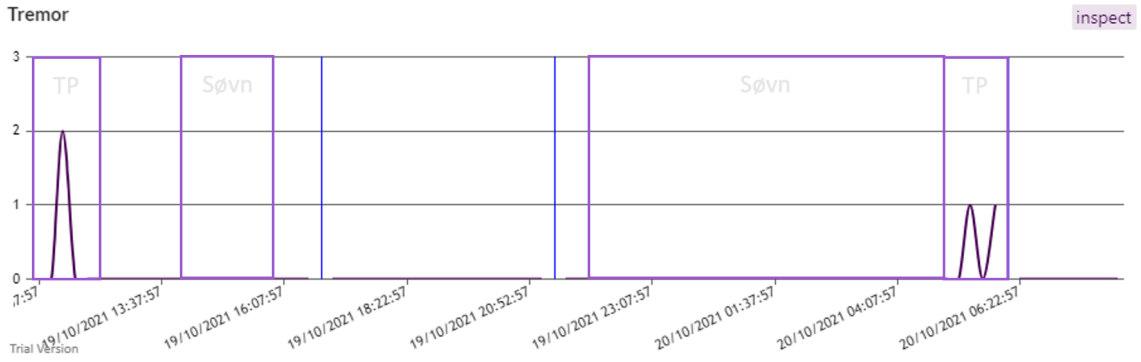


Figure 6.3: Right Side. The Y-axis is the predicted tremor severity. The x-axis shows the time stamps. The squared regions marks the model's predictions. TP (true positive), FP (false positive). Lastly the mark "søvn" indicates the hours the PD-subject was asleep.

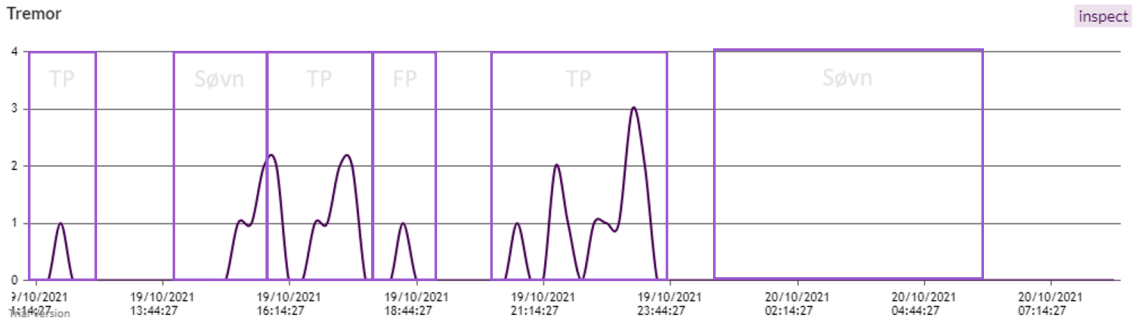


Figure 6.4: Left Side. The Y-axis is the predicted tremor severity. The x-axis shows the time stamps. The squared regions marks the model's predictions. TP (true positive), FP (false positive). Lastly the mark "søvn" indicates the hours the PD-subject was asleep.

6.3 Data Processing

There are several requirements that need to be fulfilled in order to make this data match the training data structure for testing. (1) The data needs to be (re/down) sampled at 100Hz. (2) The data should contain 3-axial accelerometer and 3 axial gyroscope data from both sides (Left and Right). (3) The data needs to be divided into 15 minute segments. (4) Each 15 minute data segment needs to be restructured into arrays of 180 x 500 x 12, saved separately into .npy files. (5) undergo same pre-processing steps, e.g. band-pass filtering and standardization described in section 4.3.

6.3.1 Procedure

6.3.1.1 Down-sampling and exclude EMG

The Paragit Sleeve has a sampling rate of 2000Hz for EMG and 500Hz for IMU, and the sleeve automatically up-samples the IMU into 2000Hz to match the EMG data. Therefore we are down-

sampling the data from 2000 Hz to 100Hz. The EMG data is stored in one of the columns, which we can easily exclude from the column when we perform our following operations.

6.3.1.2 Concatenate the Right and Left Side

Concatenating the data from the left and right side is very essential, since the data we have used for training the model takes recordings from both arms as one observation. This data from Paragit needs to be transformed into same format. For each one of the recording the sleeve device keeps meta data about the start and end time as shown in figure 6.5. We see that the start and end time differ slightly, the right side starts 6 minute and 30 seconds before the left side, while they are taken off approximately at the same time but with 1 second difference. Which means we will remove start of the right side data and end of the left side data in order to match the recordings.

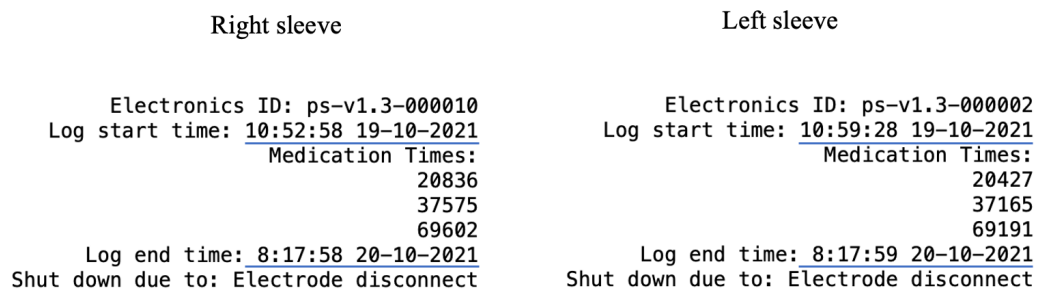


Figure 6.5: Meta data about the recordings: The log start and end time is shown here. We see the recording starts 10:52 (right) and 10:59 (left). Both sides ends around 8:17.

6.3.1.3 Filtering, Dividing into separate files and Standardization

The same pre-processing step as the training data is applied with band-pass filtering as described in the section about pre-processing 4.3, which is carried out by using 5-th order butter-worth filter with cut off at 0.3 and 9 Hz. For each 90000 observations (15 minute with sampling rate at 100Hz), we reshape the data into the desired dimension of 180x 500x 12. The standardization is performed with the Z-score standardization method, using the accelerometer and gyroscope mean and standardization for each 15 min. window. Lastly, all segments were saved into separate files.

CHAPTER 7

Results

In this chapter we will present the results of the two different Attention Based Multiple Instance Learning models we introduced previously. The major difference is in the feature extraction component of the model. The first model takes the raw signal as input and has a CNN-based feature extraction layer. The second model takes the frequency representation of the data as input and performs feature extraction using fully connected layers. In the following section we will refer to the models as "Raw" (the model that takes the raw data as input) and "Freq" (the model that takes the frequency data as input).

7.1 Overview

The results of the model described in previous sections are listed down below. This overview does not include the evaluation of the Paragit Solution's data since we do not have 1:1 corresponding labelling for the 15 minutes intervals. Therefore, we will recreate a timeline formatted evaluation on the results in section 7.6. The performance of the models are presented in table 7.1. The F1-scores for both models are almost equal when it comes to validation and testing. In order to achieve optimal performance and avoid overfitting we have applied early stopping. The Raw-model is stopped at Epoch 484 and the Freq-model is stopped slightly earlier at 467. Furthermore, the computation time for 60 test observations is also presented down below. The time is significantly less when using the frequency data compared to the raw signal. The predictions are compared to the ground truth in the confusion matrix shown in table 7.2 for the Raw-model and in table 7.3 for the Freq-model. Most of the predictions are TP for both models. The Raw-model has 23 TP out of 30 positive tremor bags. The Freq-model has 25 TP out of 30 positive tremor bags. Both models have the same performance when it comes to the TN and FN. This corresponds well to the F1-score in table 7.1, showing that the modes have similar performance. As introduced in table

Table 7.1: Overview of the results.

Input data	Training F_1	Validation F_1	Testing F_1	Epochs	Testing time (sec)
Raw	**0.985**	0.717	0.657	484	4.916
Freq	0.765	**0.728**	**0.694**	467	**0.171**

Table 7.2: Confusion matrix for the model's predictions with raw data input.

	Raw-model (0)	Raw-model (1)
Truth (0)	13	17
Truth (1)	7	23

Table 7.3: Confusion matrix for the model's predictions with frequency data input.

	Freq-model (0)	Freq-model (1)
Truth (0)	13	17
Truth (1)	5	25

2.1 the tremor events are expected to be approximately above 3Hz. Therefore, we evaluate the dominant frequencies of the predicted observations in figure 7.1. The figure to the left shows the predictions of the Raw-model and the figure to the right shows the predictions of the Freq-model. We see that there is a group of observations that are between 3-6 Hz where the models predict them as positive classes. However there's a few observations that have high dominant frequency where the PD-subjects have not indicated them as positive tremor events (Ground truth = 0).

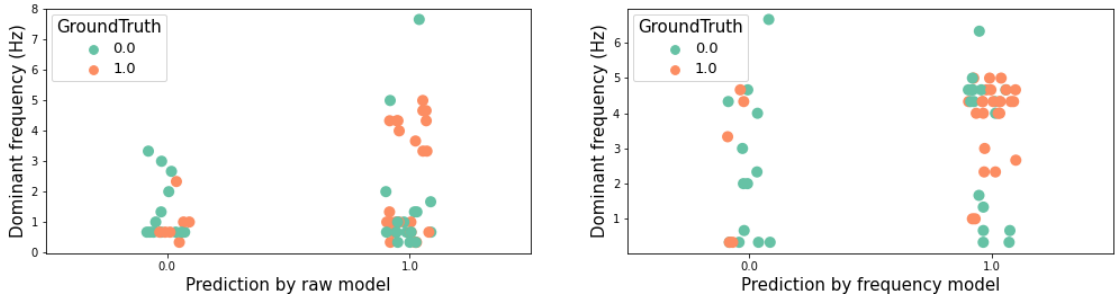


Figure 7.1: The overview of the dominant frequency of two different models. The left graph shows the results of the Raw-model and the right graph shows the results of the Freq-model.

7.2 Multiple Instance learning on Raw Kinetic Data

The Attention Based Multiple Instance Learning model on the raw data (Raw-model) is an algorithm that takes the raw kinetic data as input. In our case, the 3-axial accelerometer and gyroscope from both wrists (right and left). Each of the bag observation consists of 15 minutes of data that shares one label. Each bag observation is divided into 5 second instances. With each instance, the model outputs an attention value. The expectation is that the instance that has the largest attention is the instance that contributed the most to the final classification. In figure 7.2 we see one bag observation of the signal that is correctly classified for both models. The information about classification of this bag observation is provided in table 7.4. The blue vertical lines mark the instance with the largest attention, and the red lines mark the instance with the smallest attention. A closer look at the instances is provided in figure 7.3. We can verify visually that the observation with the largest attention has very consistent oscillations on the right side, whereas the instance with the least attention has less/none periodic behavior.

Table 7.4: Raw-model output for file number 88 by PD-subject 110002.

	Attention	Peak frequency (Hz)	Instance number
Largest	0.1236	4.67	162
Smallest	0.00032	0.667	32

7.3 Multiple Instance Learning on Frequency Domain

The Attention Based Multiple Instance Learning model on the frequency data (Freq-model) takes the estimated PSD as input. The estimated PDS is found by using the Welch's method. All PDS from the accelerometer and gyroscope are summed together, resulting in each 15 minutes bag observation has 180 instances of PDS. For comparison, we use the same bag observation as shown

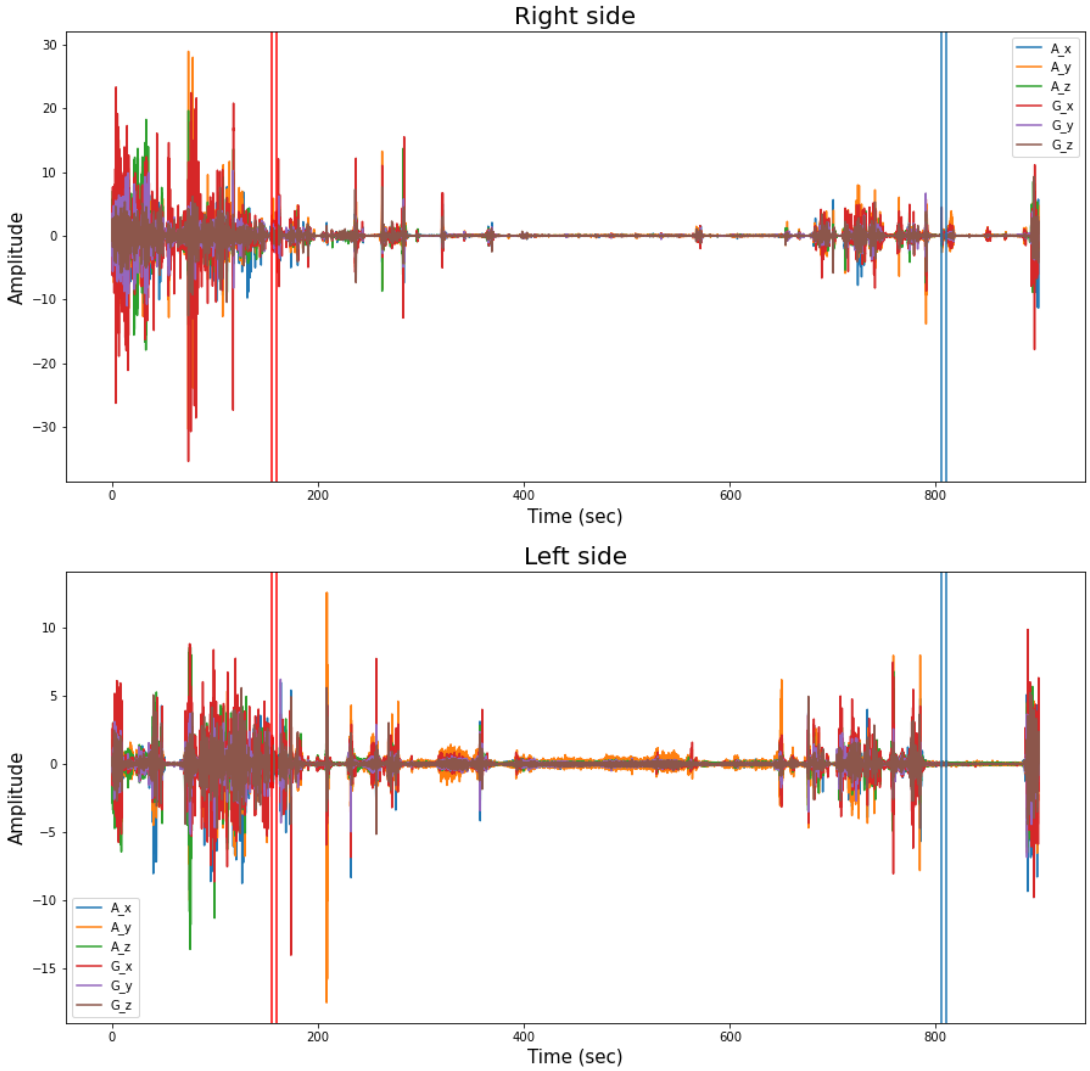


Figure 7.2: The overview of the entire 15 minute bag. The red lines mark the instance assigned with the smallest attention and the blue lines mark the instance with the largest attention. The data is from file number 88 by PD-subject 110002.

in figure 7.2. The information about classification of this bag observation is provided in table 7.5. The instances that were assigned with largest and smallest attention in the Freq-model are not the same as the Raw-model. The PDS of the instances for the Freq-model is shown in figure 7.4. The instance with the largest attention peaked at 1 and 4.67 Hz whereas the instance with the least attention peaks at 0.34 Hz.

Table 7.5: Freq-model output for file number 88 by PD-subject 110002.

	Attention	Peak frequency (Hz)	Instance number
Largest	0.0134	1. and 4.67	1
Smallest	9.1392e-05	0.34	21

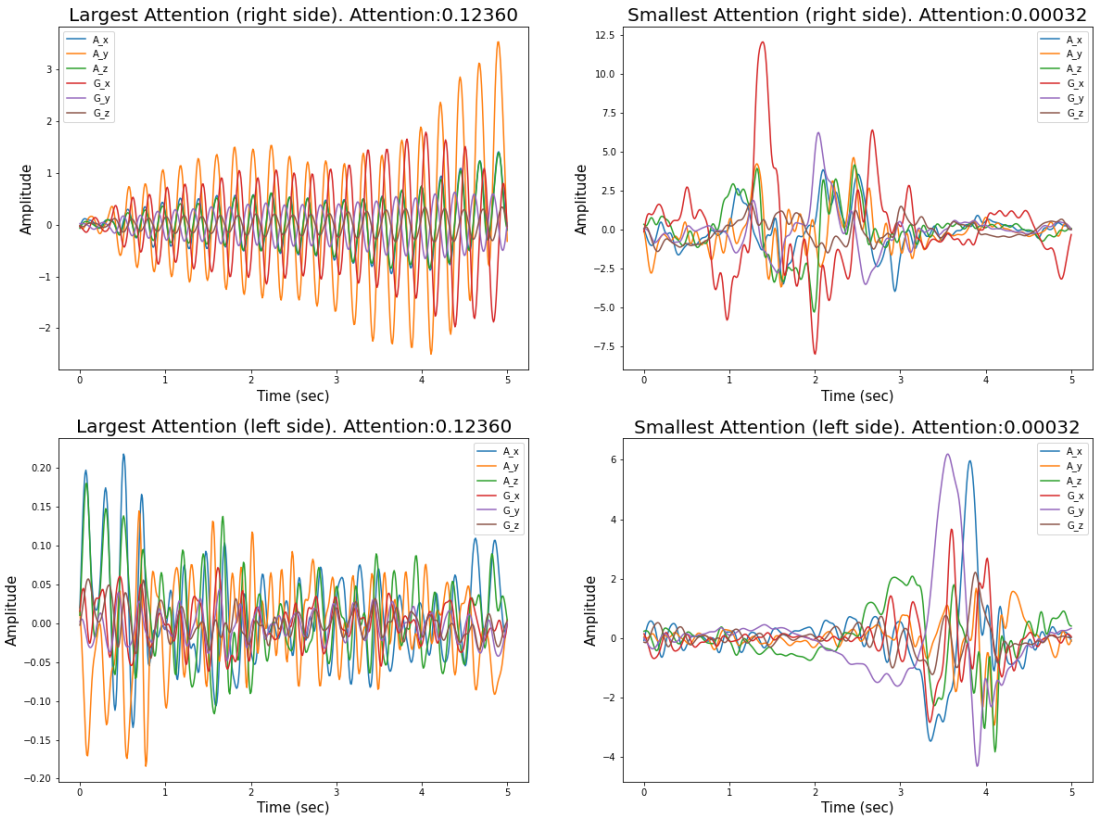


Figure 7.3: The result of one correctly classified example, where both the frequency model and the raw data model classified this bag of data for being true. The figure here is based on the raw-model's outputs. The left side shows the instance with the largest attention. The right side shows the instance with the smallest attention. Top row shows the signals from the right wrist. Bottom row shows the signals from the left wrist. The data is from file number 88 by PD-subject 110002.

7.4 False Positive Example

As we see in table 7.2 and 7.3 both models have a high number of false positives. Here, we will present an example to gain more insight of the model's choices. The time domain of the instances assigned with highest and lowest attention are presented in figure 7.6. The information of the instances are shown in table 7.6. The largest and smallest attentions for each model are not assigned to the same instance numbers, but we see that the peak frequency for the instances are very similar e.g. the high attention is assigned to the instance with high frequency (above 3Hz). The time domain signal for the instances of the largest and the smallest attentions are shown in figure 7.6. The PDS for the instance of largest and smallest attention predicted by the Raw-model is in figure 7.7, and similarly from the Freq-model in figure 7.8.

7.5 Predictions According to the Participant's ID

From previous plots in figure 4.5 we know that there is imbalance of the symptom severity among the participants. It should be noted that some participants are not a part of the test data due to the limited number of test samples.

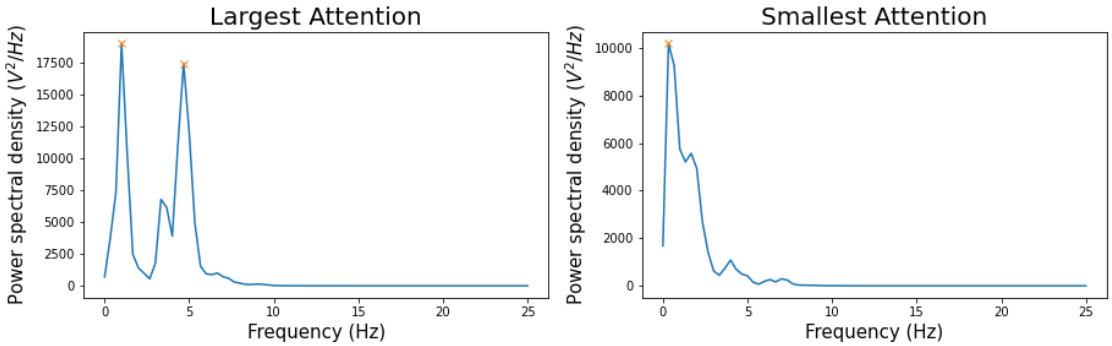


Figure 7.4: The frequency representation of the attentions assigned by the Freq-model. The power spectrum density of the instance with the largest and smallest attention. The data is from file number 88 by PD-subject 110002.

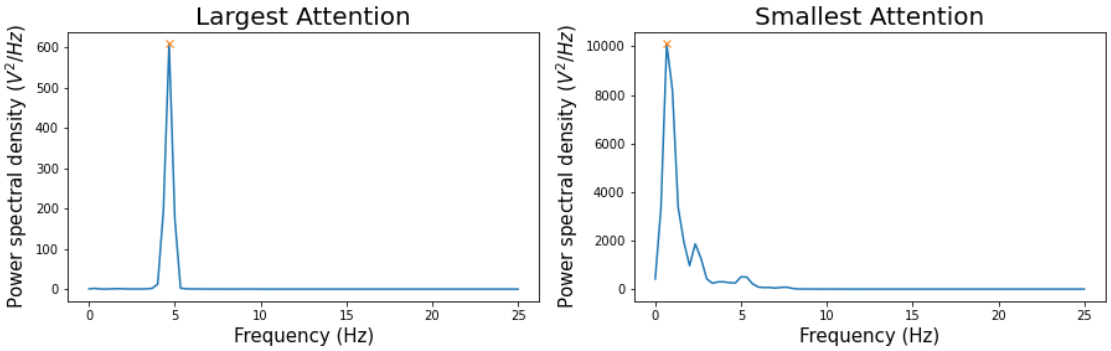


Figure 7.5: The frequency representation of the attentions assigned by the "raw" model. It peaks at 4.66 Hz for the instance with the largest and 0.67 Hz at the instance with the least attention. The data is from file number 88 by PD-subject 110002.

Table 7.6: One false positive example for the 55th file from PD-subject 110005.

	Attention	Peak frequency (Hz)	Instance number
Largest (Raw)	0.1131	5	167
Smallest (Raw)	0.0003	0.667	13
Largest (Freq)	0.0133	6.33	46
Smallest (Freq)	3.2167e-05	0.667	40

Here we visualize the outputs according to each participant ID. First we show the ground truth of the test samples in figure 7.9. The prediction of the Raw-model is shown in figure 7.10. The prediction of the frequency model is shown in figure 7.11. The distribution of ID 110002 and 110005 are very evenly represented in the data. Like before, the participants with ID 110002 and 110005 experienced different severity shown in figure 4.5. The results from ID 110002 and 110005 are mostly correct in both models.

The majority of the observations from ID 110011, 110019 and 110018 were positive. As we expected, both models (figure 7.10 and figure 7.11) predicted all observations from them as positive. In general, the models have high rates of FP. In case with data from ID 110009, both models

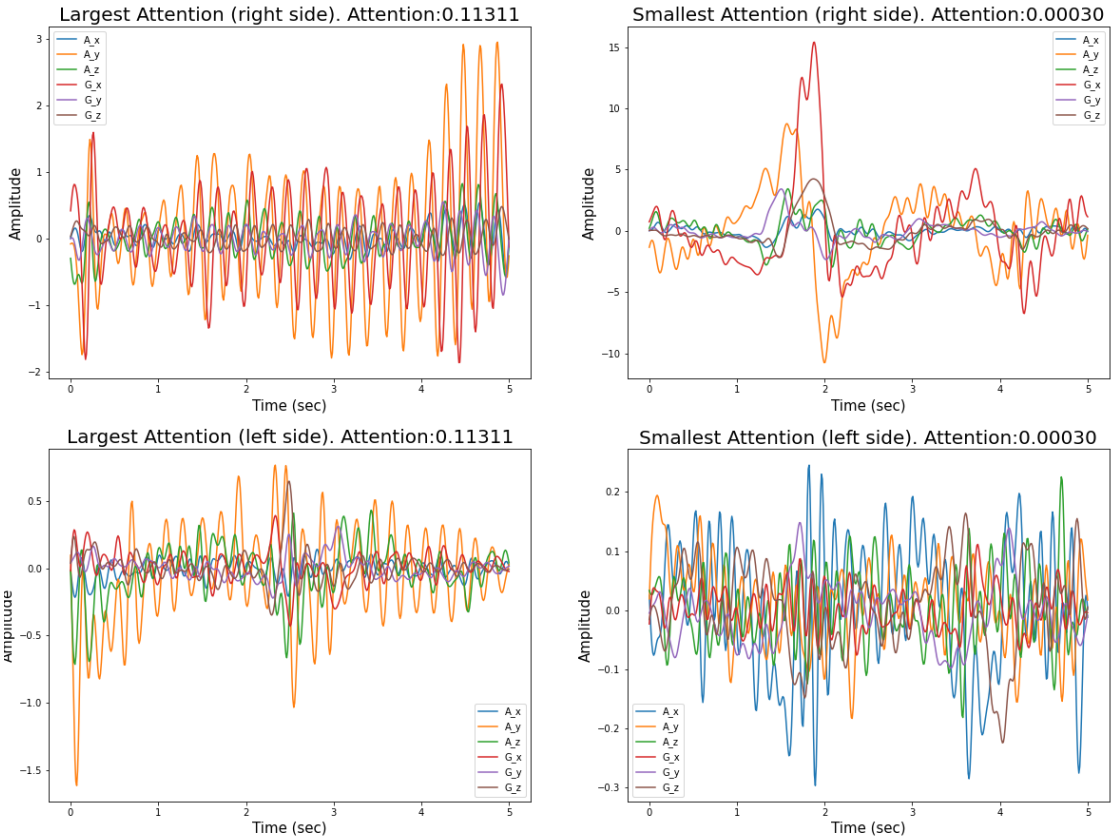


Figure 7.6: One false positive example for the 55th file from PD-subject 110005. Both Freq-model and Raw-Model predicted this observation as positive for tremor. The left side shows the instance with the largest attention. The right side shows the instance with the smallest attention. Top row shows the signals from the right wrist. Bottom row shows the signals from the left wrist.

were able to predict them correctly as negative for tremor. This could be due to the majority of the training data from ID 110009 were no-tremor observations. The participants with ID 110001, 110003, 110004, 110008 had mostly experienced none-tremor episodes but in these cases the models were predicting both true and false.

7.6 Paragit Solution Results

With the two models trained on the data from Habets et al. we will test the model's performance on Paragit Solution's data. This data recording originates from one Parkinson's Disease subject. The person had been instructed to wear the device for approximately one day while filling out the diary for symptom occurrence. More specifically, the sleeve was worn from 10:52 to 8:17 the next day (shown in figure 6.5).

The diary is treated as an approximated time-frame for the predictions as we do not have 15 minute labels but one label for each hour. The answers from the diary have been binarized and color-coded in figure 7.12. Each line represents the model's prediction for tremor in a 15 minute

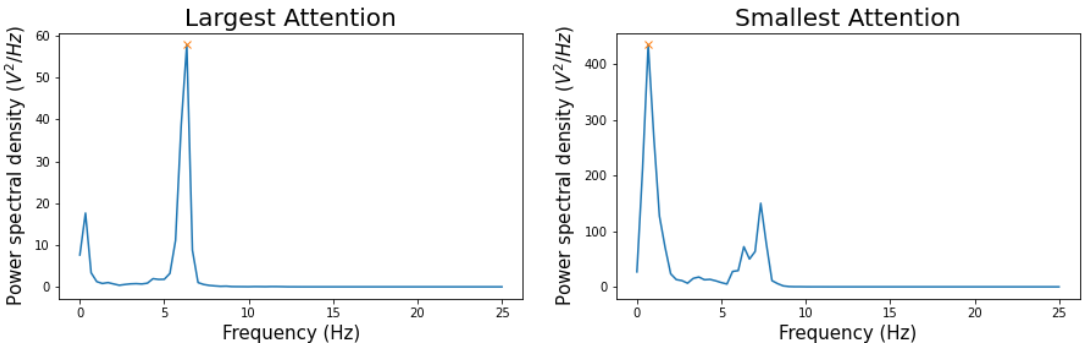


Figure 7.7: PDS of the result predicted by the Raw-model: false positive example for the 55th file from PD-subject 110005. Instance with larges attention peak at 6.3 Hz with the least attention peak at 0.67 Hz.

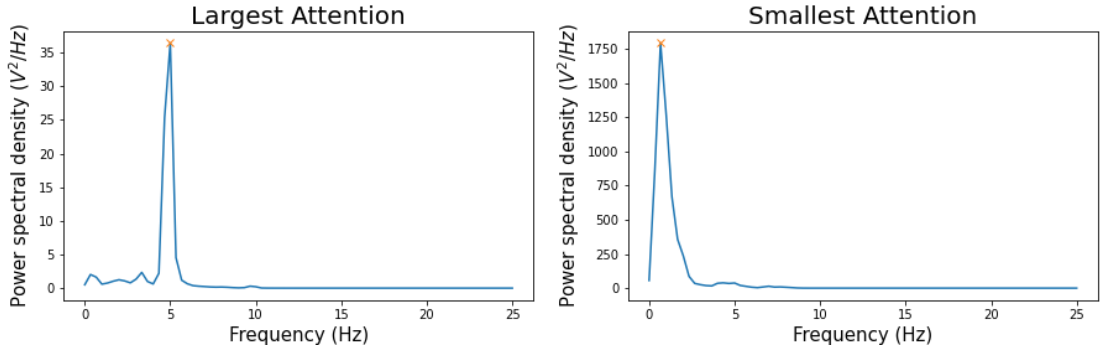


Figure 7.8: Result predicted by the Freq-model: False positive example for the 55th file from PD-subject 110005. The instance with larges attention peak at 5 Hz with the least attention peak at 0.67 Hz.

time interval. It must be noted that PD-symptom is a wide category and covers both bradykinesia and tremor.

The performance of the Raw-model is shown in figure 7.12. The first hour, where the PD-subject has answered ”yes” (marked with green) to PD-symptom, the model is able to detect 4 observations correctly with some time shift. Similarly the model has correctly found some tremor events that correlates to the diary answers during the time between 9th to 11th hour. The model did not find any tremor during the 5th hour, which potentially could be the PD-subject experienced other Parkinson’s symptoms. During the PD-subject’s sleep hours, we are not provided with any information regarding symptoms.

The result of the Freq-model is shown in figure 7.13. We see that the model is overly sensitive to the PSD signal and most of the observations are classified as positive for tremor. Furthermore we have added some examples of the Raw-model’s predictions of true in figure 7.15 and false in figure 7.16. Using the same method as previously, the overview of the dominant frequency for each model’s prediction is shown in figure 7.14, from the figure, we can see that the Freq-model has predicted most of the high-frequency instances as positive, whereas the raw-model has a more wide range of dominant frequencies across the predictions.

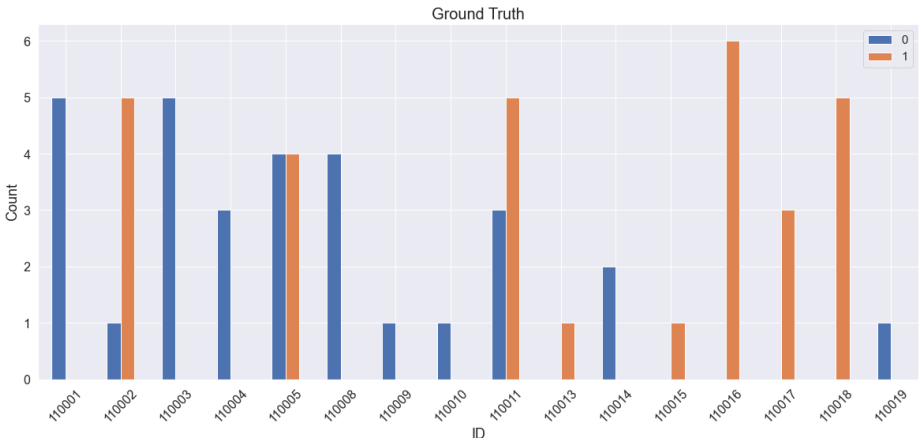


Figure 7.9: The ground truth of the labels in the test data according to each participant ID. We see we do not have evenly distributed representations of each class (0 and 1) across the participants.

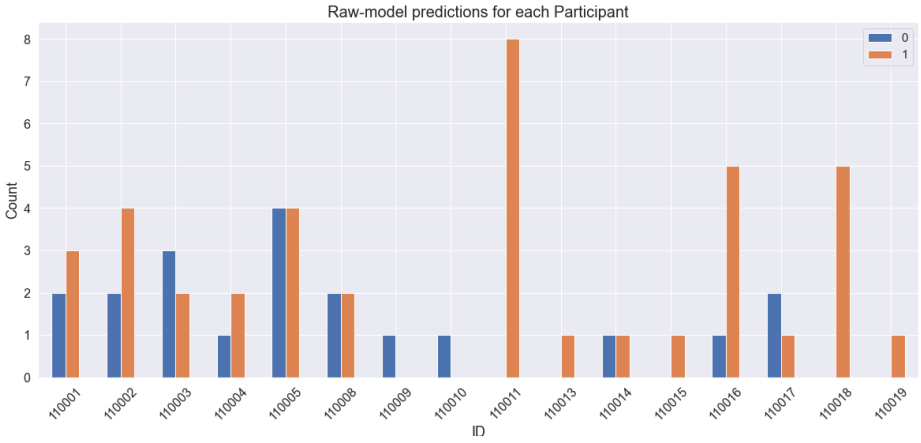


Figure 7.10: The predictions of the Raw-model according to each participant ID. The bars show the number of observations classified as negative (0) and positive (1) for tremor for each PD-subject.

In comparison, we have presented two examples from the raw-model. Figure 7.15 shows an example of a tremor prediction. Figure 7.16 shows an example of no-tremor prediction. In both cases, we can see that the instances with largest attention have largest oscillations. The no-tremor prediction has lower frequency of 1.667 Hz compared to the tremor prediction with 5.0 Hz. Paragit's own algorithm detected that the left side was dominant in terms of PD-tremor. We see in figure figure 7.15 that the instance with the largest attention was more dominant on the left side.

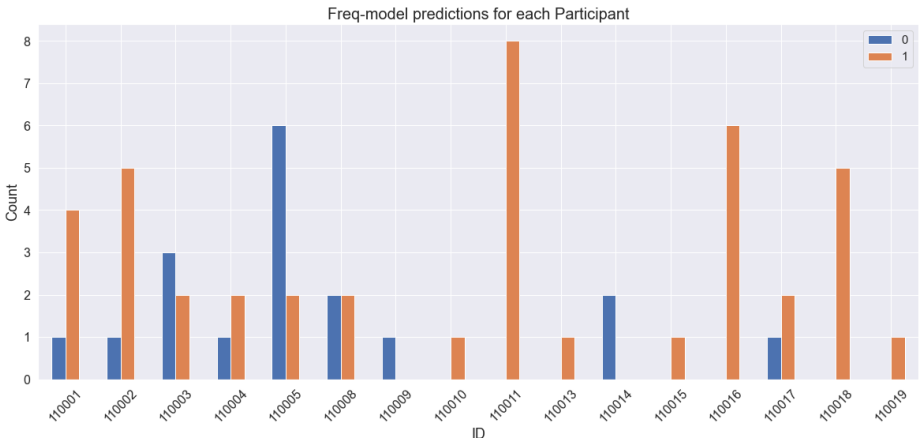


Figure 7.11: The predictions of the Freq-model according to each participant ID. The bars show the number of observations classified as negative (0) and positive (1) for tremor for each PD-subject.

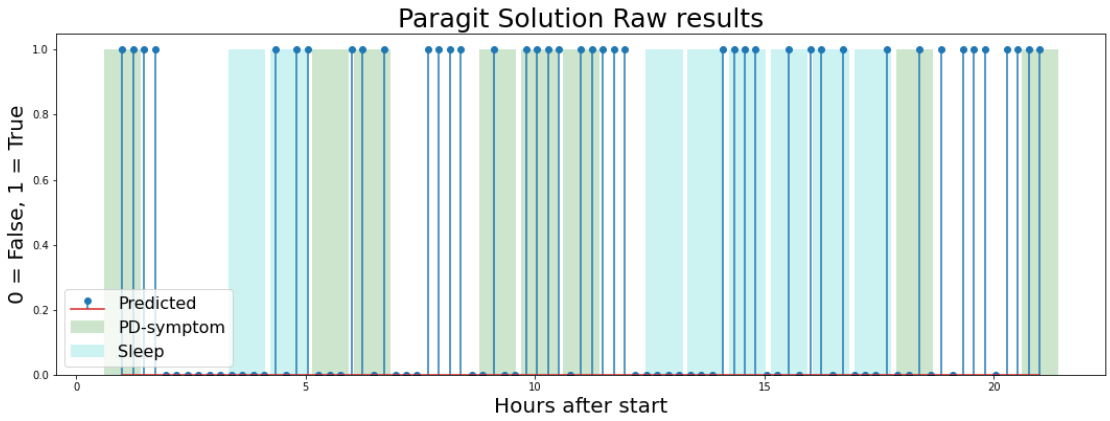


Figure 7.12: The result of the raw signal classification. The blue shaded area represents hours where the PD-subject is sleeping. The green shaded area represents the hours where the PD-subject experienced Parkinson's Symptoms. The x-axis indicates the number of hours after the sleeve has been turned on.

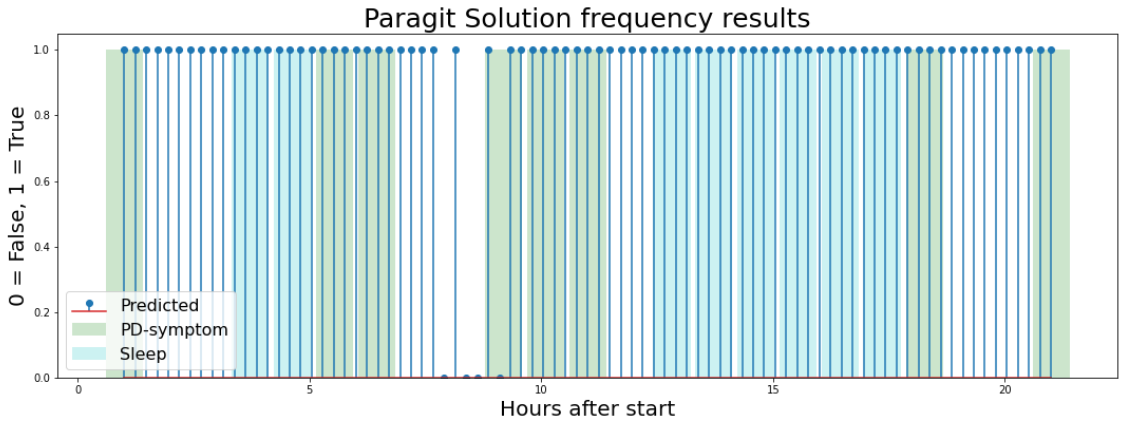


Figure 7.13: The result of the frequency signal classification. The Blue shaded area represents hours where the PD-subject is sleeping. The green shaded area represents the hours where the PD-subject experienced Parkinson's Symptoms. The x-axis indicates the number of hours after the sleeve has been turned on.

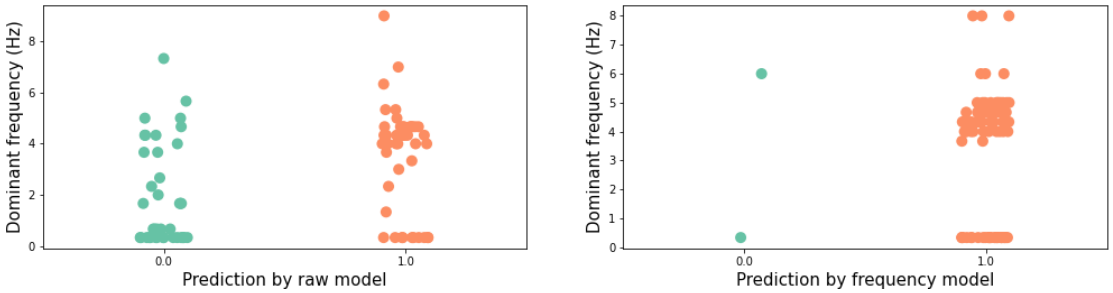


Figure 7.14: The dominant frequency of the signal and predictions of Paragit's data in two different models. The left graph shows the results of the Raw-model and the right graph shows the results of the Freq-model..

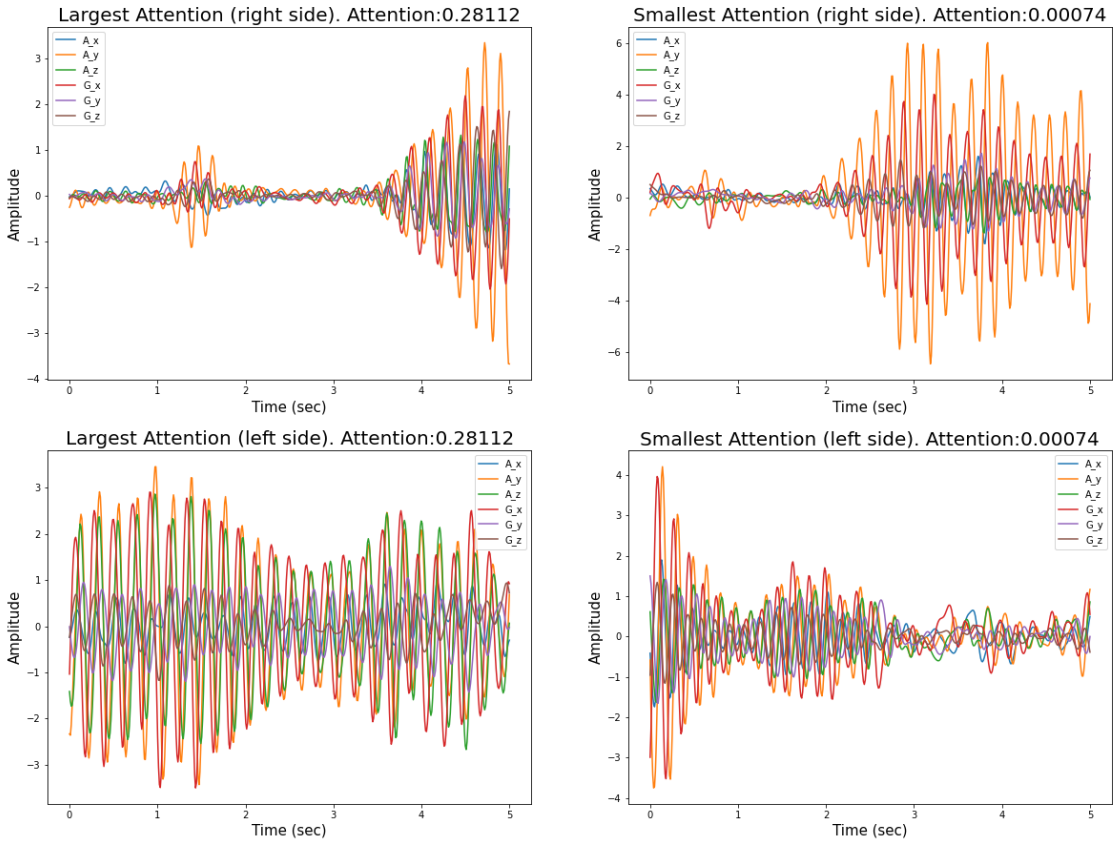


Figure 7.15: PD-subject experiences PD-symptom and correctly detected by the Raw-model. The left side shows the instance with the largest attention. The right side shows the instance with the smallest attention. Top row shows the signals from the right wrist. Bottom row shows the signals from the left wrist. The dominant frequency for the instance with the largest attention is 5.0 Hz .

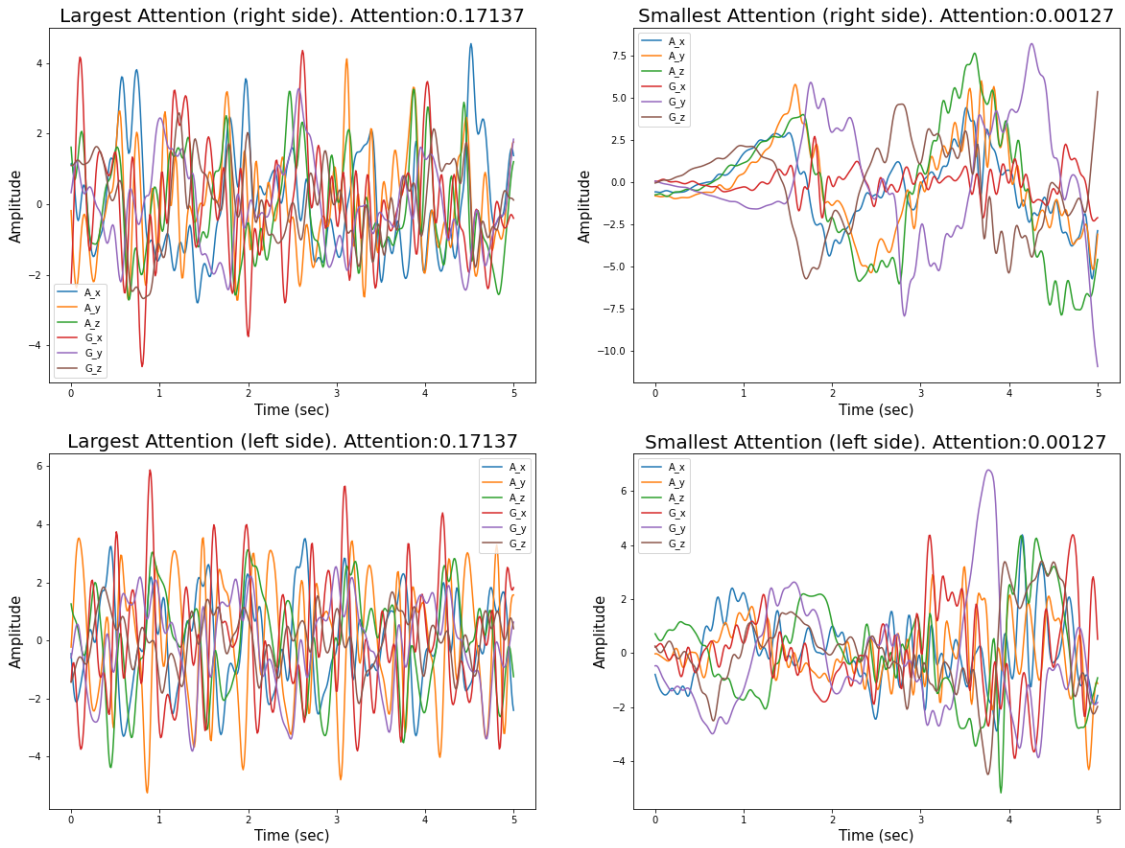


Figure 7.16: PD-subject experiences normal movement and correctly detected by the Raw-model. The left side shows the instance with the largest attention. The right side shows the instance with the smallest attention. Top row shows the signals from the right wrist. Bottom row shows the signals from the left wrist. The dominant frequency for the instance with the largest attention is 1.667 Hz.

CHAPTER 8

Discussion

In this chapter we include the discussions about the results from the proposed model and the challenges throughout this project. Furthermore we will discuss the future perspective of the data acquisition and application of symptom monitoring in the wild systems.

8.1 Summery on Results and Discussion

We examined two different models, one that takes the raw data (raw-model) as input and one with the power spectrum density as input (referred to as the frequency model). The model was evaluated based on F1-score, where the best score was obtained by the frequency model on validation and test set. Both models had similar performance in the testing phase with high TP and FP rates.

With each bag observation, we examined the instance with large and small attention, and validated that for majority of the cases. The high-attention was assigned to the instances with high dominant frequencies when the model predicted the bag as positive. The large and small attentions are not assigned to the same instance in the two models. e.g with the same bag observation, the models focus on different segments of the data. Moreover, we investigated the distribution of the predictions of the two models according to the participant ID. We note that the model's ability to predict tremor is influenced by the imbalanced data. Both models suffer from many false positives. A possible factor that could arise tremor-alike episodes in non-tremor observations, is if the wrist bands are not tighten correctly and thereby causing the oscillatory signal in the data.

We applied the models for prediction on the data collected by Paragit Solutions. Their data was collected using a different device compared to the data used for training and validation of the models. However, the raw-model showed best correspondence to the PD-subject's diary. The frequency-model was highly sensitive to the data from Paragit and predicted most of the bag observations as positive. It is hard to assign correct performance metric to the data collected by Paragit due to: First, the labels cover hourly intervals. Second, the diary field to indicate the PD-symptoms covers both tremor and bradykinesia that are opposite when it comes to the frequencies.

For both of the test cases, our test data-set was very small. We had 60 samples from the study by Habets et al. and 86 from Paragit. The data from Paragit originates from one PD-subject, therefore, it is not enough to draw any conclusions at this point. Moreover, we are highly relying on the PD-subject's own assessments of their symptom experiences during their daily life. This can affect the labels' accuracy and consistency.

Paragit Solutions' current tremor algorithm is based on thresholding the percentages of the high frequency sub-windows within larger segments. This method is static and not flexible in terms of new subjects. In order to perform a better comparison and performance evaluation in terms of generalizability we will require more data. Furthermore, the data needs to be collected under a protocol where it is possible to validate it.

8.2 Attention Based Multiple Instance Learning

The model implemented in this project was proposed by Papadopoulos et al. [21] who demonstrated satisfying results on weakly labeled data for tremor detection. The Attention Based Multiple Instance Learning model consists of 3 main modules, the feature extraction module, attention module and classification module. The training loss in the model with raw inputs tends to converge healthy but the validation loss reaches it's minimum around/under 20 epochs during 5-fold cross validation. The model seems to be vulnerable to overfitting, since we have a high risk that the model is focusing more on the daily movements than we desire. It could be considered that a reduction of the model's complexity could potentially move the focus away from the daily activities.

8.3 Optimization

The model tuning requires extensive amount of work. Sometimes the tuning process is "a black art" that requires expert knowledge, experience or in some cases brute force search [64]. In this project, we implemented a previously demonstrated model by [21], who showed promising results on tremor data collected in the wild. Therefore, we decided to keep the main components. Our loss in the raw model was converging in the training data, but for the validation, it was almost not able to move further below 0.6 which is not desirable. The loss for the frequency model was somewhat better in terms of validation loss since it was decreasing with the training loss. However, the frequency model was stagnating very early. It can be explained that the model is not complex enough, issues within the data or the hyper-parameters.

If there were more time it could potentially be that we should experiment with more of the model structure. Koutsoukas et al [65]. showed with feed forward network, that the activation function, dropout regularization, number hidden layers and number of neurons play a critical role for optimization. With further consideration, there is a need for more investigation in determining which parameters are worth tuning on. Snoek et al.[64] proposed the use of Gaussian based method which gives better estimation by using previous results. Gaussian based method, such as Tree of Parzen Estimators (TPE) implemented in Optuna has been shown to find the best configuration of hyperparameters with a fraction of the time [57]. Optuna has been applied in this project, but we were still limited by the computational time required to perform a hyperparameter search that could cover the satisfying parameter-space with the number of trials. Reimers et al. [66] showed that the optimizers have high influence on the performance. Adam and the Adam variant that incorporates Nesterov momentum gave the best result. They suggested that by adapting the learning rate, the performance and convergence time would improve.

8.4 The Data

The training data was acquired from the study done by Habets et al. [42], [43]. The data from 20 different PD-subjects has various disease stages. The subjects who are stable in PD will not have large fluctuations throughout the day, which means some participants will only give the same constant answer throughout sampling period. In order to gather data with higher variance more subjects with different disease-stages must be included. The data used in this project originates from online resources, which gives us limitations with regards of performing outlier removal and the resolution of labels corresponding to the signal are unchangeable. The app labelling method by Habets et al. was good but the 15 minute time frame for labelling was too wide compared to the window size of 3-5 seconds for tremor. The data was normalized for the purpose of achieving faster convergence and ensure the training (Habets') and testing (Paragit's) data lies in a similar range. Different approaches for standardization or normalization can be applied, we used z-score standardization where we ensured that the proportion of movements between the 3-axes were retained.

8.4.1 Two-Sided Data

The labelling of the data was generalized to both sides of the body, therefore we are unable to train the model on separate sides. This limits us from using data that are not two-sided. For instance, some of the PD-subjects who participated in Paragit's clinical trial have taken off or only used one sleeve. When the data is two sided and shares one label, we might have added added more noise to our model. Furthermore, we are unable to detect the potential symmetry or asymmetry of the tremor occurrence. For example, if the symptom breaks out at left side and we train the model using both sides, we will then have created more uncertainty to the labelling. Moreover, this sets up barriers for future use of the model for diagnosis. Symmetry of the PD can potentially shed light on the disease origin of the body and progression. Borghammer et al. [67] hypothesizes that PD can be divided into sub-types, e.g. brain-first or body-first. The body-first PD-subjects showed more symmetric distribution of α-synuclein (protein that indicates loss of dopamine) and promotes a faster disease progression compared to brain-first subjects [68].

8.4.2 Subjective Labels

Every participant got a personal briefing at their homes but the subjects are not medically trained professionals there might therefore be some concerns with regards to the accuracy of the annotations made (the EMA). From the evaluation done by Habets et al. [43] that showed the PD-subjects can have trouble understanding dyskinesia, which potentially could add noise to the data. Furthermore, the participants need to answer a range of different PD-related questions. This could potentially lead to mislabelling or not fully focusing on one specific symptom. We are unable to get the data validated professionally by a doctor or clinician. This means that we have to trust the labels as they are. Every participant is different and the meaning of scores will be individually different, meaning that the understanding of the degree of symptom severity does not apply in the same way for everyone. This creates a bias that can potentially be eliminated by training individual models for each subject [38]. This is not the most efficient implementation as we are interested in scaleable solution.

8.4.3 Data complexity

When working with data collected from the wild, it adds complexity to the problem. Even though the data is good for capturing the day to day fluctuations of Parkinson's symptoms, it is challenged by the large variety of daily activities. The unbalance of the amount of data from different PD-subjects could affect the model's ability to generalize the signals coming from the least represented subject. The data we used as training set were less accurate compared to the study where the model was proposed. Papadopoulos et al. [39] conducted their data collection using similar settings with the participants, who are picking up their phone. e.g. standing while the hand is close to their head. Papadopoulos et al. were able to achieve a F1-score of 0.897 on data that were annotated by signal processing experts. Furthermore, they experimented with using UPDRS16 (the subject's self-report for tremor) as labels, and achieved similar results like this project, which is a F1-score of 0.615. This shows that labelling has a large influence on the model's performance. Therefore, with the increased complexity and variability in the wild data, there is a great need for large amount of data with sufficient annotations.

8.4.4 Testing data

It is difficult to estimate the performance of these methods in the wild when no ground truth is available. The testing data used in this project from Paragit originates from one PD-subject. The number of available data for testing is limited by the progress in clinical trial with Odense Hospital. With data from one participant, we are unable conclude if the raw-model works best. If we were to evaluate the model, more data with accurate annotations from PD-subjects with wide variety of disease progression should be included.

8.5 Future Work

8.5.1 Real Time Processing

We are unable to validate our data properly for the time being. Real-time processing could potentially be an interesting addition to this project. The suggestion would be to use the pre-trained model in a real-time processing device, that can prompt the PD-subjects with the classification results and ask the participant to validate if the findings fit their experience. This could possibly be a way for continuing with data solely "from the wild". Many studies have used various Machine Learning techniques for discrimination of healthy controls versus PD. The most popular procedure were feature extraction and used classification models to assign the observations. Some examples of baseline features used for these classifications were Data Range, Root Mean Square, Dominant Frequency, Ratio of Energy, Signal Entropy, Peak Cross Correlation and Time-lag Peak Cross-Correlation are computed for every 2 second windows with 1 second overlap [37][69]. The time that takes for performing the computations are fairly long, which makes this approach less attractive for real-time processing purposes. Therefore, we will suggest to use simple and fast processing procedures like the ones in this study.

8.5.2 Data Collection

This project had the focus of detecting of tremor using data collected in wild setting. For future work, the data could be improved by including the specific side of symptom outbreak. Furthermore, the data should include participants with a wider range of symptom severity, in order to have equally representation of each severity category. One way to increase the number of data is to apply data augmentation. The data quality can be further improved by increase the time precision on the label, e.g. change the width of window from 15 to 5 minutes. As we are unable to know the exact ground truth, it should be investigated if there is any trade-off method between clinical and wild data collection. One suggestion could be that the model training should be based on simulated everyday-like data and test/validation should be done in relation to a clinical trial with integrated real time processing (described in section 8.5.1). This could potentially be a good combination.

CHAPTER 9

Conclusion

In this project, we have explored the possibility of implementing a tremor detecting model that was applicable on kinetic data collected from an everyday life setting. Our project included the investigation and finding of a suitable dataset and model that was able to handle the challenges of the use of data from the wild such as the noise generated from daily activities and the weak labels.

This project was conducted in collaboration with Paragit Solutions, a company that built a wearable device that samples accelerometer, gyroscope and sEMG signals from the forearm. Due to delays and labelling challenges we needed to find an online data source for conducting model training. However, we received Paragit' Sleeve-data from one PD-subject and used it as test sample along with the online data. The online data we found was collected by Habets et al. [43]. Their data resembled the data from Paragit Solution when it comes to wildness but unfortunately without any sEMG signal.

Through literature search, we found that the most suitable method for detecting tremor from data collected in wild settings was the Attention Based Multiple Instance Learning model proposed by [39]. The MIL model can handle weakly label data by considering each observation as a bag of instances. Where each positive bag contains one or more instances that are positive. The model outputs an attention parameter that was interpretable. Through examples we showed that the largest attention was assigned to the instances that contained tremor episodes. Therefore, the model was able to find the region of interest within an approximated timeframe of 15 minutes labelled by the PD-subjects themselves.

We compared the performance between a MIL model with raw data and one that uses frequency representation as input. The models achieved similar results, F1-score of 0.657 with raw data input and 0.694 with frequency input. Without precise labelling, the evaluation of the model's performance on Paragit's data was done using visual inspection and comparison of the model's predictions with the PD-diary. The model with raw input performed the best on Paragit's data. The frequency model was highly sensitive and predicted most of the observations as positive for tremor. At this moment, with limited number of data, we can not conclude finally which model has the best generalization ability.

One of the biggest challenge is the complexity of the wild data when they are mixed with daily activities. Both models have many false positive predictions. It could be that we have slightly imbalanced data from the beginning. In general we have more positive than negative tremor observations. Through visual inspection we noticed that some of them have tremor-alike patterns in the signal. This noise can potentially arise from everyday tasks that are naturally repetitive such as brushing teeth. One potential risk factor is if the wearable device is loosely attached to the wrist and thereby causing oscillation with no relation to tremor. Another reason behind the false positives could be mislabelling by the PD-subjects. The PD-subjects are asked many questions in EMA questionnaire throughout the day, their attention might not be only on their tremor experiences, which can lead to some degree of mislabelling. Furthermore, the participants experienced different levels of tremor and we have uneven number of observations from them. This means the model's performance is person specific. e.g. the performance is dependent on the data amount and tremor level variations from each participant.

There is more to be done before it can be used as the primary diagnostic algorithm, since more clarity is needed from the training data and the choices the model is making to ensure no misdiagnosis of the PD-subjects. It is in no doubt that the evolvements of wearable devices will improve the diagnosis and monitoring of the disease progression, thereby greatly improving the quality of life for the people with Parkinson's Disease.

APPENDIX A

Appendix

A.1 Code availability

The code and model used in this project is available at `https://github.com/Jiayijiayi-design/PD_Thesis`

A.2 Related work

The articles included in Chapter: Related work is found by using the key words shown in figure A.1 and filtered by papers written in English from no later than 5 years back (2016-2021).

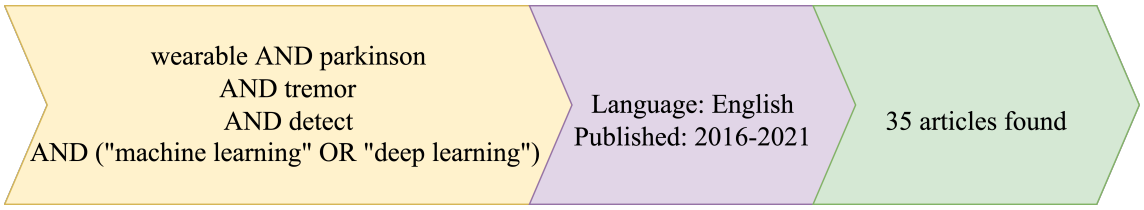


Figure A.1: The search criteria for finding the articles used in chapter: Related work.

A.3 Logbook

Parkinson dagbog

DATO	Kl.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
HYPERKINESIER/ DYSKINESIER																									
NORMAL BEVÆGELIGHED																									
PARKINSON SYMPTOMER																									
OFF																									
SØVN																									
BEMÆRKNINGER																									

DATO	Kl.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
HYPERKINESIER/ DYSKINESIER																									
NORMAL BEVÆGELIGHED																									
PARKINSON SYMPTOMER																									
OFF																									
SØVN																									
BEMÆRKNINGER																									

HYPERKINESIER: Overbevægelser
DYSKINESIER: Ufrivillige bevægelser
PARKINSYMPTOMER: Rysten, langsomme bevægelser
OFF: Pludselig indsættende svær stivhed, nedsat mobilitet

A.4 Beep Questionnaire

Beep questionnaire (semi-random repeated moments)

- I feel well
- I feel down
- I feel fearful
- I feel stressed
- I feel sleepy
- I am tired
- I am cheerful
- I am relaxed
- I can concentrate well
- I experience hallucinations
- I am at [home, work, travelling, at family/friend's place, in public]
- I am with [nobody, family, partner, colleagues, friends]
- I am doing [work, resting, household/odd jobs, sports, something else]
- I can do this without hinder
- I am comfortable walking/standing
- I can sit or stand still easily
- I can speak easily
- I can walk easily
- I experience tremor
- I am moving slow
- I experience stiffness
- My muscles are tensioned
- I am uncontrollable moving
- I feel … [1: OFF, 2: ON -> OFF, 3: ON, 4: OFF -> ON]
- I took Parkinson medication since last beep [yes, no, I don't recall]

Morning questionnaire

- I slept well
- I woke up often last night
- I feel rested
- It was physically difficult to get up
- It was mentally difficult to get up

Evening questionnaire

- I had long OFF periods today
- I had many OFF periods today
- Walking went well today
- (un)dressing went well today
- Eating/ drinking went well today
- Personal care went well today
- Household activities went well today
- I was tired today

A.5 Complete architecture

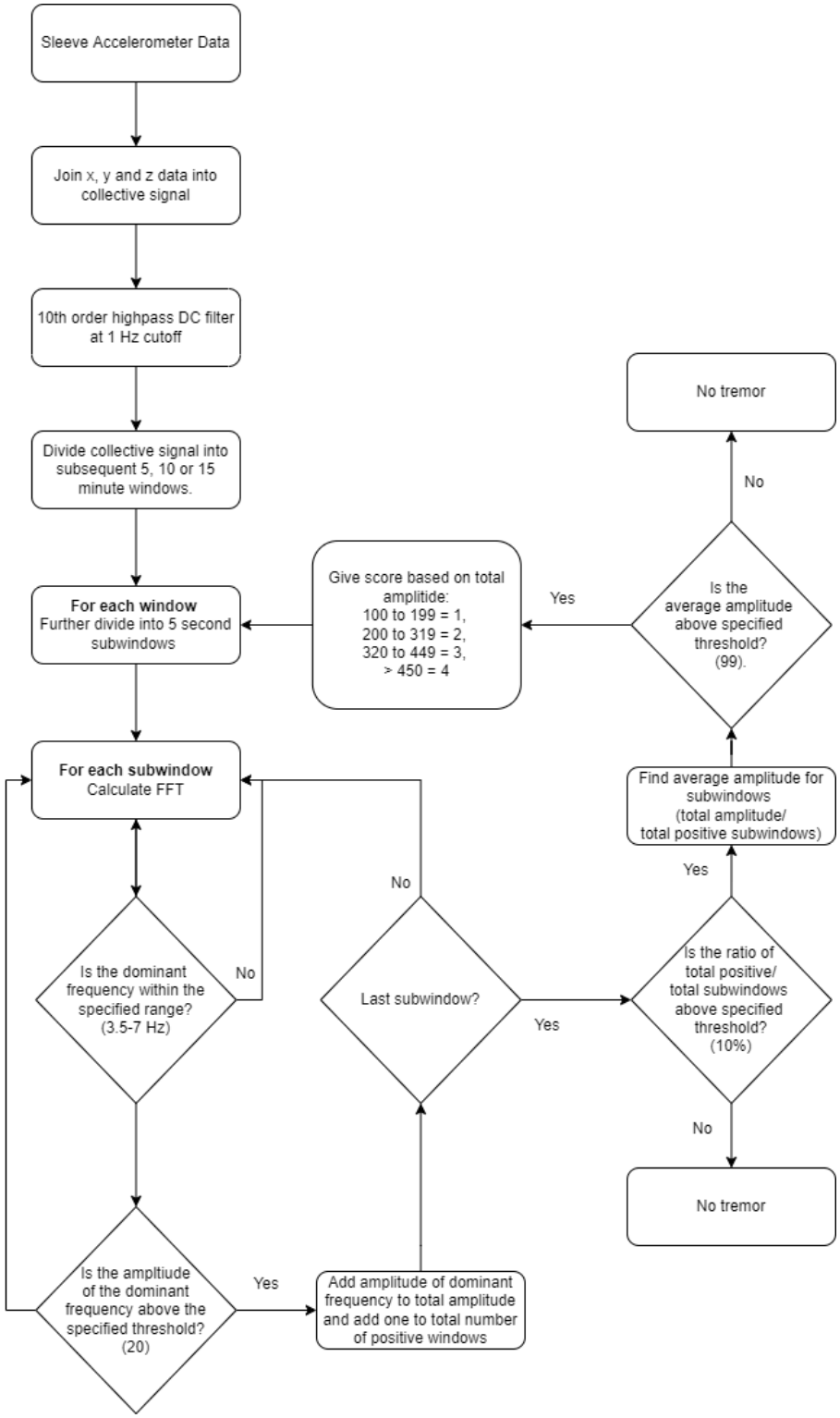
Table A.1: Feature extraction module. Source: [21].

	Fully connected	CNN
Input	1x76 spectrogram	12 x 500 kinetic data
Layer 1	Dense $76 \rightarrow 256$ Leaky-ReLu ($\alpha = 0.2$) Dropout $p = 0.5$	Conv 1D k=8, f=32 Leaky-ReLu ($\alpha = 0.2$) MaxPool k=2
Layer 2	Dense $256 \rightarrow 128$ Leaky-ReLu ($\alpha = 0.2$) Dropout $p = 0.5$	Conv 1D k=8, f=32 Leaky-ReLu ($\alpha = 0.2$) MaxPool k=2
Layer 3	Dense $128 \rightarrow$ M	Conv 1D k=16, f=16 Leaky-ReLu ($\alpha = 0.2$) MaxPool k=2
Layer 4		Conv 1D k=16, f=16 Leaky-ReLu ($\alpha = 0.2$) MaxPool k=2
Layer 5		Flatten Dense $320 \rightarrow$ M
Output	$\mathbf{h}_k \in \mathcal{R}^M$	$\mathbf{h}_k \in \mathcal{R}^M$

Table A.2: Classification module. Source: [21].

Input	$\mathbf{z} \in \mathcal{R}^M$
Layer 1	Dense M $\rightarrow$ 32 Leaky-ReLu ($\alpha = 0.2$) Dropout $p = 0.2$
Layer 2	Dense $32 \rightarrow 16$ Leaky-ReLu ($\alpha = 0.2$) Dropout $p = 0.2$
Layer 3	Dense $16 \rightarrow 2$ 2-way softmax
Output	$p(y \mid X)$

A.6 Flow chart of Paragit Solution's Tremor algorithm



Bibliography

[1] A. Merola, A. Sturchio, S. Hacker, S. Serna, J. A. Vizcarra, L. Marsili, A. Fasano, A. J. Espay, A. J. Espay James J, and J. A. Gardner, "Expert Review of Neurotherapeutics Technology-based assessment of motor and nonmotor phenomena in Parkinson disease Technology-based assessment of motor and nonmotor phenomena in Parkinson disease," 2018, ISSN: 1744-8360. DOI: 10.1080/14737175.2018.1530593. [Online]. Available: https://www.tandfonline.com/action/journalInformation?journalCode=iern20.

[2] G. Almahadin, A. Lotfi, E. Zysk, F. L. Siena, M. M. Carthy, and P. Breedon, "Parkinson's disease: current assessment methods and wearable devices for evaluation of movement disorder motor symptoms-a patient and healthcare professional perspective," DOI: 10.1186/s12883-020-01996-7. [Online]. Available: http://creativecommons.org/licenses/by/4.0/.TheCreativeCommonsPublicDomainDedicationwaiver%20http://creativecommons.org/publicdomain/zero/1.0/.

[3] J. Mei, C. Desrosiers, and J. Frasnelli, *Machine Learning for the Diagnosis of Parkinson's Disease: A Review of Literature*, May 2021. DOI: 10.3389/fnagi.2021.633752.

[4] Parkinson's News Today, *Parkinson's Disease Statistics*. [Online]. Available: https://parkinsonsnewstoday.com/parkinsons-disease-statistics/?cn-reloaded=1 (visited on 19 August 2021).

[5] Lægehåndbogen, *Parkinsons sygdom - Lægehåndbogen på sundhed.dk*. [Online]. Available: https://www.sundhed.dk/sundhedsfaglig/laegehaandbogen/neurologi/tilstande-og-sygdomme/oevrige-sygdomme/parkinsons-sygdom/ (visited on 19 August 2021).

[6] *Parkinson's disease - Diagnosis - NHS*. [Online]. Available: https://www.nhs.uk/conditions/parkinsons-disease/diagnosis/.

[7] European Parkinson's Disease Association, *Motor symptoms*. [Online]. Available: https://www.epda.eu.com/about-parkinsons/symptoms/motor-symptoms/ (visited on 19 August 2021).

[8] Parkinson's Foundation, *Understanding Parkinson's*. [Online]. Available: https://www.parkinson.org/understanding-parkinsons (visited on 19 August 2021).

[9] *What Causes Parkinson's Disease?* [Online]. Available: https://parkinsonsdisease.net/basics/pathophysiology-what-is-it.

[10] E. D. Louis and D. G. Machado, "Tremor-related quality of life: A comparison of essential tremor vs.Parkinson's disease patients," *Parkinsonism and Related Disorders*, volume 21, number 7, pages 729–735, Jul. 2015, ISSN: 18735126. DOI: 10.1016/j.parkreldis.2015.04.019.

[11] L. E. Heusinkveld, M. L. Hacker, M. Turchan, T. L. Davis, and D. Charles, "Impact of Tremor on Patients With Early Stage Parkinson's Disease," *Frontiers in Neurology*, volume 0, number AUG, page 628, August 2018, ISSN: 1664-2295. DOI: 10.3389/FNEUR.2018.00628.

[12] Y. Zhou, M. E. Jenkins, M. D. Naish, and A. L. Trejos, "The measurement and analysis of Parkinsonian hand tremor," in *3rd IEEE EMBS International Conference on Biomedical and Health Informatics, BHI 2016*, Institute of Electrical and Electronics Engineers Inc., April 2016, pages 414–417, ISBN: 9781509024551. DOI: 10.1109/BHI.2016.7455922.

[13] P. Pierleoni, L. Palma, A. Belli, and L. Pernini, "A real-time system to aid clinical classification and quantification of tremor in Parkinson's disease," in *2014 IEEE-EMBS International Conference on Biomedical and Health Informatics, BHI 2014*, IEEE Computer Society, 2014, pages 113–116, ISBN: 9781479921317. DOI: 10.1109/BHI.2014.6864317.

[14] A. Salarian, H. Russmann, C. Wider, P. R. Burkhard, F. J. Vingerhoets, and K. Aminian, "Quantification of tremor and bradykinesia in Parkinson's disease using a novel ambulatory monitoring system," *IEEE Transactions on Biomedical Engineering*, volume 54, number 2, pages 313–322, February 2007, ISSN: 00189294. DOI: 10.1109/TBME.2006.886670.

[15] A. Channa, R. C. Ifrim, D. Popescu, and N. Popescu, "A-wear bracelet for detection of hand tremor and bradykinesia in parkinson's patients," *Sensors (Switzerland)*, volume 21, number 3, pages 1–23, February 2021. DOI: 10.3390/S21030981.

[16] H. Dai, P. Zhang, and T. C. Lueth, "Quantitative Assessment of Parkinsonian Tremor Based on an Inertial Measurement Unit," *Sensors (Basel, Switzerland)*, volume 15, number 10, page 25 055, 2015. DOI: 10.3390/S151025055. [Online]. Available: /pmc/articles/PMC4634500/%20/pmc/articles/PMC4634500/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4634500/.

[17] R. San-Segundo, A. Zhang, A. Cebulla, S. Panev, G. Tabor, K. Stebbins, R. E. Massa, A. Whitford, F. de la Torre, and J. Hodgins, "Parkinson's disease tremor detection in the wild using wearable accelerometers," *Sensors (Switzerland)*, volume 20, number 20, pages 1–23, October 2020, ISSN: 14248220. DOI: 10.3390/s20205817.

[18] N. Mahadevan, C. Demanuele, H. Zhang, D. Volfson, B. Ho, M. K. Erb, and S. Patel, "Development of digital biomarkers for resting tremor and bradykinesia using a wrist-worn wearable device," *npj Digital Medicine*, volume 3, number 1, December 2020. DOI: 10.1038/S41746-019-0217-7.

[19] *European Parkinson's Disease Association*. [Online]. Available: https://www.epda.eu.com.

[20] A. Baraka, H. Shaban, M. A. El-Nasr, and O. Attallah, "Wearable accelerometer and sEMG-based upper limb BSN for tele-rehabilitation," *Applied Sciences (Switzerland)*, volume 9, number 14, Jul. 2019. DOI: 10.3390/APP9142795.

[21] A. Papadopoulos, K. Kyritsis, L. Klingelhoefer, S. Bostanjopoulou, K. R. Chaudhuri, and A. Delopoulos, "Detecting Parkinsonian Tremor from IMU Data Collected In-The-Wild using Deep Multiple-Instance Learning," *IEEE Journal of Biomedical and Health Informatics*, 2019, ISSN: 21682208. DOI: 10.1109/JBHI.2019.2961748.

[22] I. Mazzetta, A. Zampogna, A. Suppa, A. Gumiero, M. Pessione, and F. Irrera, "Wearable sensors system for an improved analysis of freezing of gait in Parkinson's disease using electromyography and inertial signals," *Sensors (Switzerland)*, volume 19, number 4, February 2019, ISSN: 14248220. DOI: 10.3390/s19040948.

[23] C. Bhavana, J. Gopal, P. Raghavendra, K. M. Vanitha, and V. Talasila, "Techniques of measurement for Parkinson's tremor highlighting advantages of embedded IMU over EMG," in *2016 International Conference on Recent Trends in Information Technology, ICRTIT 2016*, Institute of Electrical and Electronics Engineers Inc., Sep. 2016, ISBN: 9781467398022. DOI: 10.1109/ICRTIT.2016.7569560.

[24] D. Powell, A. Joseph Threlkeld, X. Fang, A. Muthumani, and R. Xia, "Amplitude- and velocity-dependency of rigidity measured at the wrist in Parkinson's disease," *Clinical Neurophysiology*, volume 123, number 4, pages 764–773, April 2012, ISSN: 13882457. DOI: 10.1016/j.clinph.2011.08.004.

[25] J. R. Williamson, B. Telfer, R. Mullany, and K. E. Friedl, "Detecting parkinson's disease from wrist-worn accelerometry in the U.K. biobank," *Sensors*, volume 21, number 6, pages 1–18, March 2021. DOI: 10.3390/S21062047.

[26] A. P. S. Paixão, L. B. Peres, and A. O. Andrade, "Parameter estimate from accelerometer and gyroscope for characterization of wrist tremor in individuals with parkinson's disease," *IFMBE Proceedings*, volume 70, number 1, pages 513–517, 2019, ISSN: 14339277. DOI: 10.1007/978-981-13-2119-1{_}79.

[27] L. Raiano, G. Di Pino, L. Di Biase, M. Tombini, N. L. Tagliamonte, and D. Formica, "PDMeter: A Wrist Wearable Device for an at-Home Assessment of the Parkinson's Disease Rigidity," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, volume 28, number 6, pages 1325–1333, Jun. 2020, ISSN: 15580210. DOI: 10.1109/TNSRE.2020.2987020.

[28] H. B. Kim, W. W. Lee, A. Kim, H. J. Lee, H. Y. Park, H. S. Jeon, S. K. Kim, B. Jeon, and K. S. Park, "Wrist sensor-based tremor severity quantification in Parkinson's disease using convolutional neural network," *Computers in Biology and Medicine*, volume 95, pages 140–146, April 2018, ISSN: 18790534. DOI: 10.1016/j.compbiomed.2018.02.007.

[29] Y. N. Zhang, "Can a Smartphone Diagnose Parkinson Disease? A Deep Neural Network Method and Telediagnosis System Implementation," *Parkinson's Disease*, volume 2017, 2017, ISSN: 20420080. DOI: 10.1155/2017/6209703.

[30] V. Skaramagkas, G. Andrikopoulos, Z. Kefalopoulou, and P. Polychronopoulos, "Towards differential diagnosis of essential and parkinson's tremor via machine learning," *2020 28th Mediterranean Conference on Control and Automation, MED 2020*, pages 782–787, 2020. DOI: 10.1109/MED48518.2020.9182922.

[31] L. Lonini, A. Dai, N. Shawen, T. Simuni, C. Poon, L. Shimanovich, M. Daeschler, R. Ghaffari, J. A. Rogers, and A. Jayaraman, "Wearable sensors for Parkinson's disease: which data are worth collecting for training symptom detection models," *npj Digital Medicine*, volume 1, number 1, December 2018, ISSN: 2398-6352. DOI: 10.1038/s41746-018-0071-z.

[32] A. Talitckii, E. Kovalenko, A. Anikina, O. Zimniakova, M. Semenov, E. Bril, A. Shcherbak, D. V. Dylov, and A. Somov, "Avoiding Misdiagnosis of Parkinson's Disease with the Use of Wearable Sensors and Artificial Intelligence," *IEEE Sensors Journal*, volume 21, number 3, pages 3738–3747, February 2021, ISSN: 15581748. DOI: 10.1109/JSEN.2020.3027564.

[33] J. Mei, C. Desrosiers, and J. Frasnelli, "Machine learning for the diagnosis of Parkinson's disease: A systematic review," *arXiv*, October 2020. [Online]. Available: https://arxiv.org/abs/2010.06101v1.

[34] A. Zhang, R. San-Segundo, S. Panev, G. Tabor, K. Stebbins, A. Whitford, F. De La Torre, and J. Hodgins, "Automated tremor detection in Parkinson's disease using accelerometer signals," *Proceedings - 2018 IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies, CHASE 2018*, volume 18, pages 13–14, February 2019. DOI: 10.1145/3278576.3278582. [Online]. Available: https://doi.org/10.1145/3278576.3278582.

[35] H. Jeon, W. Lee, H. Park, H. J. Lee, S. K. Kim, H. B. Kim, B. Jeon, and K. S. Park, "Automatic classification of tremor severity in Parkinson's disease using awearable device," *Sensors (Switzerland)*, volume 17, number 9, Sep. 2017, ISSN: 14248220. DOI: 10.3390/s17092067.

[36] J. E. Thorp, P. G. Adamczyk, H. L. Ploeg, and K. A. Pickett, *Monitoring Motor Symptoms During Activities of Daily Living in Individuals With Parkinson's Disease*, December 2018. DOI: 10.3389/fneur.2018.01036.

[37] A. Zhang, A. Cebulla, S. Panev, J. Hodgins, and F. De La Torre, "Weakly-supervised learning for Parkinson's Disease tremor detection," *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, pages 143–147, 2017. DOI: 10.1109/EMBC.2017.8036782.

[38] A. Zhang, F. De La Torre, and J. Hodgins, "Comparing laboratory and in-the-wild data for continuous Parkinson's Disease tremor detection," *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, volume 2020-July, pages 5436–5441, Jul. 2020. DOI: 10.1109/EMBC44109.2020.9176255.

[39] A. Papadopoulos, K. Kyritsis, S. Bostanjopoulou, L. Klingelhoefer, R. K. Chaudhuri, and A. Delopoulos, "Multiple-Instance Learning for In-The-Wild Parkinsonian Tremor Detection," *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, 2019.

[40] *PD-BioStampRC21: Parkinson's Disease Accelerometry Dataset from Five Wearable Sensor Study | IEEE DataPort*. [Online]. Available: https://ieee-dataport.org/open-access/pd-biostamprc21-parkinsons-disease-accelerometry-dataset-five-wearable-sensor-study-0 (visited on 25 October 2021).

[41] *MJFF Levodopa Response Study - syn20681023 - Wiki*. [Online]. Available: https://www.synapse.org/#!Synapse:syn20681023/wiki/594678 (visited on 25 October 2021).

[42] J. G. V. Habets, M. Heijmans, A. F. G. Leentjens, C. J. P. Simons, Y. Temel, M. L. Kuijf, P. L. Kubben, and C. Herff, "A Long-Term, Real-Life Parkinson Monitoring Database Combining Unscripted Objective and Subjective Recordings," *Data 2021, Vol. 6, Page 22*, volume 6, number 2, page 22, February 2021. DOI: 10.3390/DATA6020022. [Online]. Available: https://www.mdpi.com/2306-5729/6/2/22/htm%20https://www.mdpi.com/2306-5729/6/2/22.

[43] J. Habets, M. Heijmans, C. Herff, C. Simons, A. F. Leentjens, Y. Temel, M. Kuijf, and P. Kubben, "Mobile health daily life monitoring for parkinson disease: Development and validation of ecological momentary assessments," *JMIR mHealth and uHealth*, volume 8, number 5, May 2020, ISSN: 22915222. DOI: 10.2196/15628.

[44] M. Heijmans, J. G. Habets, C. Herff, J. Aarts, A. Stevens, M. L. Kuijf, and P. L. Kubben, "Monitoring Parkinson's disease symptoms during daily life: a feasibility study," *npj Parkinson's Disease*, volume 5, number 1, December 2019, ISSN: 23738057. DOI: 10.1038/s41531-019-0093-5.

[45] S. Puthusserypady, *Applied Signal Processing*. Now Publishers, 2021. DOI: 10.1561/9781680839791.

[46] *Welch Method - an overview | ScienceDirect Topics*. [Online]. Available: https://www.sciencedirect.com/topics/mathematics/welch-method.

[47] M. Ilse, J. M. Tomczak, and M. Welling, "Attention-based Deep Multiple Instance Learning," 2018.

[48] A. Papadopoulos, D. Iakovakis, L. Klingelhoefer, S. Bostantjopoulou, K. R. Chaudhuri, K. Kyritsis, S. Hadjidimitriou, V. Charisis, L. J. Hadjileontiadis, and A. Delopoulos, "Unobtrusive detection of Parkinson's disease from multi-modal and in-the-wild sensor data using deep learning techniques," 2020. DOI: 10.1038/s41598-020-78418-8. [Online]. Available: https://doi.org/10.1038/s41598-020-78418-8.

[49] Michael Nielsen, *Neural Networks and Deep Learning*. [Online]. Available: http://neuralnetworksanddeeplearning.com/chap6.html (visited on 11 Sep. 2021).

[50] *Understanding of Convolutional Neural Network (CNN) — Deep Learning | by Prabhu | Medium*. [Online]. Available: https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148.

[51] *Machine Learning Mastery*. [Online]. Available: https://machinelearningmastery.com/.

[52] *A Practical Guide to ReLU. Start using and understanding ReLU… | by Danqing Liu | Medium*. [Online]. Available: https://medium.com/@danqing/a-practical-guide-to-relu-b83ca804f1f7.

[53] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors,"

[54] *HPC LSF10*. [Online]. Available: https://www.hpc.dtu.dk/?page_id=2759.

[55] *Hyper-parameter optimization algorithms: a short review | by Aloïs Bissuel | Criteo R&D Blog | Medium*. [Online]. Available: https://medium.com/criteo-engineering/hyper-parameter-optimization-algorithms-2fe447525903#e4fa.

[56] J. Bergstra, J. B. Ca, and Y. B. Ca, "Random Search for Hyper-Parameter Optimization Yoshua Bengio," *Journal of Machine Learning Research*, volume 13, pages 281–305, 2012. [Online]. Available: http://scikit-learn.sourceforge.net..

[57] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A Next-generation Hyperparameter Optimization Framework," *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2623–2631, Jul. 2019. DOI: 10.1145/3292500.3330701. [Online]. Available: https://arxiv.org/abs/1907.10902v1.

[58] D. P. Kingma and J. Lei Ba, "ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION,"

[59] G. Hinton, N. Srivastava, and K. Swersky, *Neural Networks for Machine Learning Lecture 6a-6e*.

[60] M. D. Zeiler, "ADADELTA: AN ADAPTIVE LEARNING RATE METHOD,"

[61] *CS231n Convolutional Neural Networks for Visual Recognition*. [Online]. Available: https://cs231n.github.io/neural-networks-3/#accuracy.

[62] M. Sokolova, N. Japkowicz, and S. Szpakowicz, "Beyond Accuracy, F-Score and ROC: A Family of Discriminant Measures for Performance Evaluation," *AAAI Workshop - Technical Report*, volume WS-06-06, pages 1015–1021, 2006. DOI: 10.1007/11941439{_}114. [Online]. Available: https://link.springer.com/chapter/10.1007/11941439_114.

[63] R. Xia, J. Sun, and A. J. Threlkeld, "Analysis of interactive effect of stretch reflex and shortening reaction on rigidity in Parkinson's disease," *Clinical Neurophysiology*, volume 120, number 7, pages 1400–1407, Jul. 2009, ISSN: 13882457. DOI: 10.1016/j.clinph.2009.05.001.

[64] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian Optimization of Machine Learning Algorithms,"

[65] A. Koutsoukas, K. J. Monaghan, X. Li, and J. Huan, "Deep-learning: Investigating deep neural networks hyper-parameters and comparison of performance to shallow methods for modeling bioactivity data," *Journal of Cheminformatics*, volume 9, number 1, pages 1–13, Jun. 2017, ISSN: 17582946. DOI: 10.1186/S13321-017-0226-Y/FIGURES/5. [Online]. Available: https://jcheminf.biomedcentral.com/articles/10.1186/s13321-017-0226-y.

[66] N. Reimers and I. Gurevych, "Optimal Hyperparameters for Deep LSTM-Networks for Sequence Labeling Tasks," [Online]. Available: www.ukp.tu-darmstadt.de.

[67] P. Borghammer and N. Van Den Berge, "Brain-First versus Gut-First Parkinson's Disease: A Hypothesis," *Journal of Parkinson's Disease*, volume 9, number Suppl 2, S281, 2019, ISSN: 1877718X. DOI: 10.3233/JPD-191721. [Online]. Available: /pmc/articles/PMC6839496/%20/pmc/articles/PMC6839496/?report=abstract%20https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6839496/.

[68] P. Borghammer, "The α-Synuclein Origin and Connectome Model (SOC Model) of Parkinson's Disease: Explaining Motor Asymmetry, Non-Motor Phenotypes, and Cognitive Decline," *Journal of Parkinson's disease*, volume 11, number 2, pages 455–474, 2021, ISSN: 1877-718X. DOI: 10.3233/JPD-202481. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/33682732/.

[69] J. M. Fisher, N. Y. Hammerla, T. Ploetz, P. Andras, L. Rochester, and R. W. Walker, "Unsupervised home monitoring of Parkinson's disease motor symptoms using body-worn accelerometers," 2016. DOI: 10.1016/j.parkreldis.2016.09.009. [Online]. Available: http://dx.doi.org/10.1016/j.parkreldis.2016.09.009.