Predicting the Severity of Parkinson's Disease Symptoms with Smartwatches during Daily Living

by

Marie-Philippe GILL

THESIS PRESENTED TO ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
IN PARTIAL FULFILLMENT FOR A MASTER'S DEGREE
WITH THESIS IN INFORMATION TECHNOLOGY
M.A.Sc.

MONTREAL, JUNE 1, 2021

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

BOARD OF EXAMINERS

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS

M. Patrick Cardinal, Thesis Supervisor
Department of Software Engineering and Information Technology, École de technologie supérieure

M. Laureano Moro-Velazquez, Thesis Co-supervisor
Department of Electrical and Computer Engineering, The Johns Hopkins University

Mme Sylvie Ratté, Chair, Board of Examiners
Department of Software Engineering and Information Technology, École de technologie supérieure

M. Najim Dehak, External Examiner
Department of Electrical and Computer Engineering, The Johns Hopkins University

THIS THESIS WAS PRESENTED AND DEFENDED

IN THE PRESENCE OF A BOARD OF EXAMINERS AND THE PUBLIC

ON APRIL 19, 2021

AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

ACKNOWLEDGEMENTS

I want to thank my advisors for their help and guidance. Patrick, thank you for believing in me since my first day in your class as an undergraduate student. Laureano, thank you for your patience. You will be a fantastic professor.

I am also grateful to Najim Dehak for welcoming me into his lab at Johns Hopkins University (JHU). I had an amazing time and made great friendships. Thank you to everyone who collaborated on the BEAT-PD challenge with me: Nanxin Chen, Saurabhchand Bhati, and Sonal Joshi. Your contribution was essential to the success of this project.

J'aimerais remercier mes parents qui ont fait tellement de sacrifices pour me permettre de me rendre où je suis aujourd'hui. Étudier n'a pas toujours été facile pour moi, et mes parents ont tout fait pour m'aider. Ils ont réussi leur pari. Merci à toute ma famille pour leur support.

Thank you to my partner, Desh, for your advice and support while writing this thesis.

This work started during the Biomarker & Endpoint assessment to track Parkinson's disease DREAM Challenge, which is funded by the Michael J. Fox Foundation for Parkinson's Research. This work was financially supported by funding from the Fonds de recherche - Nature et technologies of Quebec (FQRNT) #290699.

Prédire la sévérité des symptômes du Parkinson avec des montres intelligentes pendant la vie de tous les jours

Marie-Philippe GILL

RÉSUMÉ

La maladie de Parkinson affecte des millions de personnes à travers le monde et peut grandement affecter la qualité de vie des gens diagnostiqués. De nos jours, le test MDS-UPDRS évalue la progression de la maladie et est considéré comme référence. Par contre, le test est effectué en moyenne une à deux fois par année. Cela n'offre qu'une image de la sévérité des symptômes du patient la journée où le test est passé. Il y a un besoin de monitorer les symptômes de façon passive et pendant de plus grandes périodes de temps puisque les symptômes peuvent varier selon une multitude de facteurs d'une journée à l'autre. Pouvoir mesurer la sévérité des symptômes dans la vie de tous les jours pourrait permettre aux patients et aux médecins de trouver la dose et l'horaire de médication le plus optimal possible. La plupart des recherches jusqu'à maintenant ont été conduites dans des environnements contrôlés, ou à la maison, mais les patients devaient accomplir des tâches précises à certains moments planifiés dans la journée. Notre recherche vise à quantifier si la médication est efficace, ainsi que la sévérité des tremblements et de la dyskinésie à partir d'une montre intelligente portée sur le poignet. À partir de l'accéléromètre de la montre, nous proposons trois approches pour résoudre le problème à l'étude. La première est basée sur l'extraction de primitives à partir des séries chronologiques suivi d'un XGBoost, tandis que la seconde utilise des représentations vectorielles suivies de différents modèles comme PLDA et KNN, mais le SVR est celui qui finalement offre les meilleurs résultats. La troisième approche fait une simple moyenne des prédictions des deux approches. Différentes méthodes de pré-traitement des données ont été appliquées, comme l'utilisation d'un filtre passe-haut pour enlever la composante de gravité, ainsi que la détection de l'inactivité pour la retirer des données. De plus, différentes techniques pour générer artificiellement de nouvelles données ont été utilisées pour augmenter l'ensemble d'entraînement, comme la combinaison linéaire, du bruit gaussien, le ré-échantillonnage et la rotation du signal. Finalement, la troisième approche a obtenu les meilleurs résultats avec une erreur quadratique moyenne pondérée de 1.129, 0.429 et 0.462 pour le statut de la médication, les tremblements et la dyskinésie. Différents pré-traitement et techniques d'augmentation ont été utilisées et efficaces pour chaque symptôme à l'étude.

Mots-clés: Parkinson, montres intelligentes, on/off, tremblement, dyskinésie

Predicting the Severity of Parkinson's Disease Symptoms with Smartwatches during Daily Living

Marie-Philippe GILL

ABSTRACT

Parkinson's Disease (PD) is a neurodegenerative disease that affects millions of people worldwide and can significantly affect the quality of life. Currently, the Movement Disorder Society Unified Parkinson's Disease Rating Scale (MDS-UPDRS) test evaluates the disease's progression and is considered the gold standard. However, the test is done on average once or twice a year to assess motor and non-motor symptoms related to PD. It only provides a snapshot of the symptoms on a given day which is a significant limitation of the test. There is a need for passive, longitudinal, and in-the-wild home monitoring of the disease as PD symptoms can fluctuate for various reasons, such as the quality of sleep, from one day to the next. Monitoring the symptoms during daily life can provide essential insights and lead to better health care decisions for patients by providing information about how severe the symptoms are on average throughout the whole year. As a result, clinicians could find the most optimal medication schedule and dosage for patients, which has the potential to reduce side-effects of the medication like dyskinesia. Most of the work has been done either in clinical environments or at home with scripted tasks that subjects need to complete at specific times, which can be an additional burden. This is why this work aims to quantify the medication status (on/off), the severity of tremor, and dyskinesia from a single wrist-worn smartwatch during passive monitoring. Using the accelerometer of the Apple Watch, we propose three approaches to solve this problem. The first one is based on time series features extraction with an Extreme Gradient Boosting (XGBoost), while the second uses embeddings with different classifiers like Probabilistic Linear Discriminant Analysis (PLDA), K-Nearest Neighbors (KNN), and Support Vector Regression (SVR). The third approach is a fusion of both approaches using a simple average. We experimented with different data pre-processing methods, such as using a High-Pass Filter (HPF) to remove the gravity component. We also detected and removed inactivity in the signals. Furthermore, we artificially generated new samples with various data augmentation techniques such as linear combination, Gaussian noise, resampling, and rotation. Finally, the third approach was the most successful and achieved a weighted Mean Square Error (MSE) of 1.129, 0.429, and 0.462 for on/off, tremor, and dyskinesia. Thus, for each symptom, a different combination of data pre-processing and augmentation successfully improved the overall MSE.

Keywords: Parkinson's disease, passive monitoring, smartwatches, on/off, tremor, dyskinesia

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

ACRONYMS

ADL	Activities of daily living.
AE	Auto-Encoder.
ANN	Artificial Neural Network.
BEAT-PD	Biomarker & Endpoint assessment to track Parkinson's disease.
CIS-PD	Clinician Input Study.
CNN	Convolutional Neural Network.
DBS	Deep Brain Stimulation.
DCT	Discrete Cosine Transform.
DNN	Deep Neural Network.
DT	Decision Tree.
EM	Expectation-Maximization.
EMG	Electromyography.
ESM	Experience Sampling Method.
ETS	École de Technologie Supérieure.
FDA	Food and Drug Administration.
FFT	Fast Fourier Transform.
GB	Gradient Boosting.
GBR	Gradient Boosting Regression.

GMM	Gaussian Mixture Model.
HPF	High-Pass Filter.
JHU	Johns Hopkins University.
KNN	K-Nearest Neighbors.
LDA	Linear Discriminant Analysis.
LSTM	Long Short-term Memory.
MAP	Maximum A-Posteriori.
MDS	Movement Disorder Society.
MDS-UPDRS	Movement Disorder Society Unified Parkinson's Disease Rating Scale.
MFCCs	Mel-frequency Cepstral Coefficients.
mPDS	Mobile Parkinson Disease Score.
MSE	Mean Square Error.
PCA	Principal Component Analysis.
PD	Parkinson's Disease.
PDDB	Parkinson's Disease Digital Biomarker.
PLDA	Probabilistic Linear Discriminant Analysis.
RF	Random Forest.
RFE	Recursive Feature Elimination.
RFR	Random Forest Regressor.

SVM	Support Vector Machine.
SVR	Support Vector Regression.
Tsfresh	Time Series Feature extraction based on scalable hypothesis tests.
UBM	Universal Background Model.
UPDRS	Unified Parkinson's Disease Rating Scale.
XGBoost	Extreme Gradient Boosting.

INTRODUCTION

Parkinson's Disease (PD) affects more than 10 million people around the world, according to Parkinson's Disease Foundation (Parkinson's Foundation, 2018). It is the second most prevalent degenerative disease, after Alzheimer's. Due to the aging population, it is expected that more and more people will suffer from the disease. PD affects many areas of one's life: from motor symptoms like tremor or gait to non-motor symptoms like sleep, eating, fatigue, and more. Unfortunately, more than 50% of patients will suffer from symptoms of depression after their diagnosis, which will often stay undiagnosed. Symptoms of PD slowly develop over the years, and the progression is different for everyone.

Problem statement

The disease's progression is currently evaluated with the MDS-UPDRS test, which is considered the gold standard. It evaluates motor symptoms as well as non-motor symptoms. However, the MDS-UPDRS test is not perfect, and there are many criticisms. One of them is the fact that a clinician must complete the evaluation. In addition, as patients only go on average twice a year to be evaluated, people with PD are only assessed for one hour per year. In comparison, patients will spend the remaining 8759 hours of the year at home in self-care or with a caregiver.

The symptoms of PD can fluctuate during a given day based on medication levels, sleep, and other environmental factors, but people with PD also have good and bad days. Their cause is still a mystery to clinicians (American Parkinson Disease Association, 2015; A. LeWitt, 2018). Therefore, being evaluated only twice a year for 30 minutes is not enough to monitor the symptoms well on a long-term basis. For instance, what if the patient is assessed on the best day of the year? Medications decisions are taken depending on the tests' results, and side-effects like dyskinesia can appear if the medication is not optimal. Therefore, it is crucial for the patient's well-being to have the best evaluation possible.

Motivation

This work's motivation is to create a clinically valid system that can predict the severity of the symptoms when the patient is at home and going on about their life as usual. Individuals would only have to wear a smartwatch at home, like an Apple Watch, which records accelerometer data. A machine learning algorithm could then process this data to predict the severity on a scale of 0 to 4. This monitoring could allow patients and clinicians to make better health care decisions and find the best dosage and schedule of medicine that the patient needs.

BEAT-PD DREAM Challenge

The work presented in this thesis was started during the Biomarker & Endpoint assessment to track Parkinson's disease (BEAT-PD) DREAM challenge (Foschini *et al.*, 2020) that is funded by the Michael J. Fox Foundation for Parkinson's Research. For up-to-date information on the study, visit the challenge website [1]. It ran from January 2020 to May 2020 following the timeline presented in Figure 0.1. The challenge's objective was to advance the research of using wearable sensors and smartphones to assess the severity of the disease that can be clinically used to make better healthcare decisions from the sensor-based data. In other words, there is a need to translate sensor-based data into standard digital biomarkers and to develop standard methods to achieve this goal. Consequently, the BEAT-PD DREAM challenge was designed to benchmark methods for processing raw sensor data from daily living to predict PD severity.

It was essential for the challenge for the developed methods to be highly reproducible. Therefore, the code to execute the approaches used in this thesis can be found in the GitHub repository [2], along with a guide on how to run the approaches.

---

[1] https://www.synapse.org/!Synapse:syn20825169/wiki/600405

[2] https://github.com/Mymoza/BeatPD-CLSP-JHU

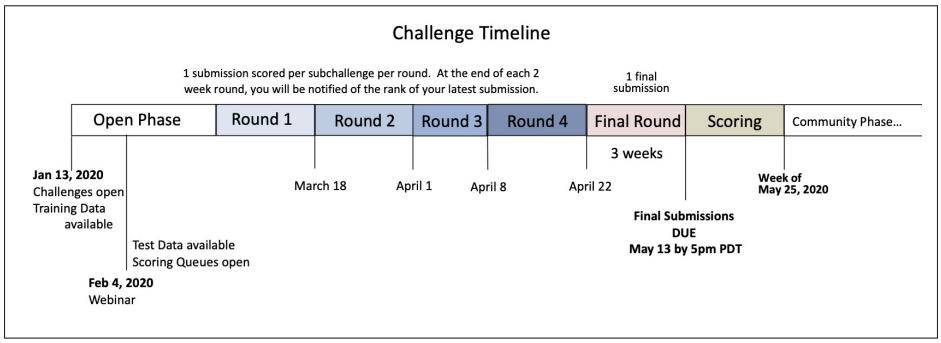


Figure 0.1 Timeline of the challenge

A similar challenge was organized in 2017, the Parkinson's Disease Digital Biomarker (PDDB) DREAM challenge (Sieberts *et al.*, 2020). Forty teams from around the world submitted different sets of features to evaluate tremor, dyskinesia, and bradykinesia. The teams used time-series data from accelerometers and gyroscopes. However, for that challenge, the data were collected while subjects were performing specific scripted tasks. This is where our work differs and precisely what makes our research novel. In the BEAT-PD DREAM challenge, the data was collected in the subjects' homes, and they were not required to perform any specific tasks.

Research questions

The research questions are :

- Can we predict the medication status (on/off) using accelerometers from a smartwatch during passive monitoring?
- Can we predict tremor severity using accelerometers from a smartwatch during passive monitoring?
- Can we predict dyskinesia severity using accelerometers from a smartwatch during passive monitoring?

Contribution

This work is novel by its use of smartwatches data to predict the severity of the on/off phenomenon and the symptoms of tremor and dyskinesia for PD. The data used in the preparation of this thesis were obtained from the Biomarker & Endpoint assessment to track Parkinson's disease (BEAT-PD) DREAM Challenge. It was collected in a completely passive manner, where the subjects were free to do what they wanted during the day. No scripted tasks were required while they were wearing the watch. In this work, we explored in parallel two approaches and concluded that their fusion provided the best results. We obtained a weighted MSE of 1.129 for on/off, 0.429 for tremor, and 0.462 for dyskinesia. Therefore, we can predict the severity of the symptoms closely. All of the code used in this work is available in a public GitHub repository. However, this study has some limitations, as the model cannot generalize to new patients, nor can it predict lower limb symptoms.

Organization

The thesis is organized as follows. Chapter 1 explains the symptoms of PD, the treatment, and how the disease is currently evaluated. Then chapter 2 reviews the literature around the use of mobile sensors for PD, by emphasizing if the evaluation uses scripted or unscripted tasks. Then, chapter 3 explains the methodology of our work. The results are presented in chapter 4, followed by a discussion in chapter 5.

CHAPTER 1

UNDERSTANDING PARKINSON'S DISEASE

This chapter will explain PD in depth. The risk factors of PD will be explored in section 1.1 and the symptoms will be discussed in section 1.2. The possible treatments are detailed in section 1.3. Finally, the current methodology to evaluate the disease's course, the MDS-UPDRS, will be explained in section 1.4.

1.1 Risk factors for Parkinson's Disease

Both men and women can have idiopathic PD. Idiopathic means that the cause is unknown and represents most of the patients diagnosed and refers to the term PD (Hoehn *et al.*, 1998). The rest of the cases are called secondary or atypical parkinsonism and are induced by drugs, strokes, and other health conditions. Idiopathic PD affects men about 1.5 times more than women (Wooten *et al.*, 2004). Symptoms of PD start appearing when cells that used to produce dopamine in the brain are damaged or die. The causes of why these cells stop producing dopamine are still widely unknown, but three factors have been identified as they might be playing an important role: genetics, environment, and aging (Ascherio & Schwarzschild, 2016; Reeve *et al.*, 2014).

First, a mutation in some genes has been identified as associated with the disease. Not all people carrying the identified genes will develop the disease, so mutations are not the only factor causing the illness and only cause a small proportion of all PD cases (De Lau & Breteler, 2006). However, the genes are more often found and identified in cases where many family members are affected. Next, the environment can play a role too. A head injury trauma or exposure to chemicals and toxins like pesticides and herbicides can trigger the disease's development. Finally, aging is still the most critical factor in developing PD, as cells get damaged as they age. Most people develop symptoms after 60 years old, but the young-onset of PD can occur in patients from 21 to 40 years old (Quinn *et al.*, 1987).

1.2 Symptoms

Individuals living with the disease can experience many symptoms. When we first think about PD, slowness of movement (bradykinesia), tremor, and rigidity are often the first cardinal symptoms that come to mind. However, those are just the tip of the iceberg. The following is a more exhaustive list of motor symptoms (American Parkinson Disease Association, 2020):

- Rest tremor: a rhythmic shaking movement around the hand or a limb at rest. When the individual is active, the tremor will disappear;
- Rigidity;
- Bradykinesia: slow movement, can also be demonstrated by reduced expressions of the face.
- Postural instability;
- Walking difficulties: especially when bradykinesia is coupled with postural instability;
- Vocal symptoms: the voice can become softer, fade away, change in volume, or it can be difficult to pronounce words;
- Trouble chewing and swallowing.

Non-motor symptoms can also affect the quality of life. The most commons are:

- Cognitive impairment: memory loss, attention deficit, lack of orientation, thinking slowly,
- Loss of smell,
- Bladder dysfunction,
- Sexual dysfunction,
- Leg swelling,
- Sleeping problems and daytime sleepiness,

- Depression: more than 50% of people diagnosed with PD will experience symptoms of depression after their diagnosis. This is also a symptom that brain damage can cause, and not only by learning a PD diagnosis.

1.3 Treatment

To date, PD cannot be cured. However, different treatments, like medication or surgery, can be used to control the symptoms (Deuschl *et al.*, 2006). Changes in the lifestyle of the patients can also improve their condition.

Many different medications can be prescribed and will be adjusted as the disease progresses (Mayo Clinic, 2020). The most common one is levodopa. It is a drug that is converted to dopamine to compensate for the lack of that hormone caused by damaged cells in the brain. It can also be used in combination with other kinds of medication (Dopamine agonists, COMT or MAO inhibitors, anticholinergic agents).

Unfortunately, although levodopa is the most effective drug to treat PD, side-effects may arise from long-term use, like the on-off phenomenon and dyskinesia. At least 50% of patients will develop these complications after 5 to 10 years of treatment (Olanow *et al.*, 2001).

For treatment with surgery, the most common for PD is Deep Brain Stimulation (DBS). The procedure is usually for patients that do not respond well to medication or have medication-induced dyskinesia (Lee *et al.*, 2018). A neurostimulator, similar to a pacemaker, is implanted under the collarbone or in the abdomen. The device will send electrical pulses to electrodes strategically placed in the brain to lessen motor symptoms like bradykinesia, rigidity, and tremor. The patient can turn on and off the device, and the device's battery can last up to 9 years.

1.3.1 On/Off

For the first 5-10 years of treatment, levodopa will control the symptoms well with 2 or 3 daily doses. However, over time, the medication will start wearing off before the next scheduled dose

(European Parkinson's Disease Association). When that happens, symptoms can reappear or worsen: the subject is "off." These periods can also become more and more frequent as the disease progresses. On the contrary, when the medication is working and controlling symptoms well, the patient is "on."

1.3.2 Dyskinesia

Dyskinesia is another long-term side effect of taking levodopa. It describes a symptom of PD where the subject has uncontrollable and involuntary movements. It is similar to tremor. However, it is disordered, can involve the entire body instead of one articulation, and is not suppressible with action. In comparison, tremor is a rhythmic movement and particularly around one joint.

Dyskinesia usually appears when the subject is "on," as the medicine induces it. This side-effect is also more common in young patients with PD as they will have to take the medication longer. Levodopa-induced dyskinesia appears gradually and also gets worse as time goes by. Unfortunately, there is no treatment for dyskinesia yet. As the PD treatment triggers it, being able to monitor someone with PD for hours can help to minimize dyskinesia by optimizing the levodopa dosage to find a compromise between the symptoms (Tsipouras *et al.*, 2012).

1.4 UPDRS

The Unified Parkinson's Disease Rating Scale (UPDRS), proposed by Fahn (1987), evaluates the course of the disease. It is considered the gold standard. Before the development of UPDRS, multiple scales were used: Webster, Columbia, King's College, and more.

The first version of UPDRS proposed was called version 3.0. It had four parts:

- Part I: Mentation, Behavior and Mood,
- Part II: Activities of Daily Living,

- Part III: Motor,
- Part IV: Complications.

Each part contains different points where clinicians or patients have to rate on a scale between 0 (none) to 4 (most severe). Therefore, the total score can be between 0 (no disease) and 199 (worst stage of the disease).

The questionnaire was developed to be a core assessment, and additional scales could be used when needed. The Schwab and England and Hoehn and Yahr scales are also widely used and reported in the literature because they were precursors of the UPDRS.

There were several limitations to UPDRS version 3.0 that were analyzed in (Movement Disorder Society Task Force on Rating Scales for Parkinson's Disease, 2003). It was reported that there were "ambiguities in the written text, inadequate instructions for raters, some metric flaws, and the absence of screening questions on several important non-motor aspects of PD." (Movement Disorder Society Task Force on Rating Scales for Parkinson's Disease, 2003) However, the UPDRS being so widely adopted and used by clinicians, it was suggested to improve the existing scale. A modification to the original scale was proposed in 2008 by the Movement Disorder Society (MDS). The revision is called the MDS-UPDRS Goetz *et al.* (2008) and aims to improve in areas of non-motor evaluation of the patients as it lacked in the original UPDRS: anxiety, fatigue, sleep disorders are only a couple of items out of the nine that were added by Goetz *et al.* (2008). The questionnaire also offers more lengthy instructions hoping that the test becomes less subjective to the clinician performing the test.

The four parts were kept, but the questions were changed:

- Part I: Non-Motor experiences of daily living,
- Part II: Motor experiences of daily living,
- Part III: Motor examination,
- Part IV: Motor complications.

Many questions from Part I and all Part II questions require the patients to answer them, so the clinician is not required. A doctor is only needed for Part I, Part III, and Part IV as they are more complicated and require to assess symptoms. Therefore, the whole evaluation takes around 25 minutes to complete.

Part III of the MDS-UPDRS assess the motor symptoms by a clinician asking the patient to perform specific tasks. Some of them are:

- Speech: listening to the patient's free-flowing speech;
- Facial expression: observe the patient at rest for 10 seconds;
- Rigidity: patient in a relaxed position with the examiner manipulating limbs and neck;
- Finger tapping: tap the index finger on the thumb of each hand ten times as quickly as possible;
- Hand movements: make a tight fist with the arms bent at the elbow, palm facing the examiner.

1.4.1 Critics to MDS-UPDRS

Even with the modification in 2008 of the UPDRS, many critiques can be made on the updated MDS-UPDRS questionnaire (Stott, 2019).

First, the test only takes a picture of how a patient is feeling on a given day. The test takes, on average, 30 minutes to complete. Therefore, it only represents the state of a patient during that 30 minutes of a given day.

The symptoms can also vary from one day to the next, experiencing the "good day/bad day" phenomenon. It is not clear yet what could be the factors for the fluctuations in the symptoms, and we need more thorough research in that area. (American Parkinson Disease Association, 2015; A. LeWitt, 2018)

Furthermore, the test is subjective. Clinicians have to observe the symptoms of patients and make a decision on the severity. Two neurologists can give two different scores for the same patient during the same appointment. Post *et al.* (2005) found that more senior clinicians will assign a lower score than younger colleagues.

The results of the test are not linear. A patient with a total score of 50 points is not necessarily doing worst than a patient who received a score of 25. It also has no deep meaning nor can act as a point of comparison, as a depressed patient can have the same score as a happy patient but experiences other symptoms. It is impossible to say what a score of 50 means, as it means something different for every patient.

As a patient will only be evaluated on average 1 hour a day in healthcare and will spend 8765 hours in self-care per year, it would be very powerful to start evaluating the course of the disease 12 hours a day, every day, with a smartwatch.

1.5 Conclusion

In conclusion, PD is a disease that affects millions of people worldwide, and its onset has a significant impact on their quality of life. A variety of symptoms can appear as cells producing dopamine in the brain are damaged or die. Motor symptoms, like tremors, provoke shaking movements, often around the hands. However, non-motor symptoms can also be experienced: loss of smell, bladder dysfunction, and even depression.

The disease can not be cured or stopped yet. As of November 2020, a new treatment is undergoing clinical study in Toronto. Using a focus ultrasound technology to treat the regions directly in the brain that are affected by the disease, doctors might soon be able to stop the disease's progression and treat symptoms (Sherman & Palisoc, 2020). Until then, treatment like levodopa can control symptoms for years. Some side-effects might appear after 5-10 years, like the on/off phenomenon and dyskinesia. The MDS-UPDRS is currently used to evaluate the progression of the symptoms. However, it does not provide long-term evaluation and only provides a punctual picture of the patient in clinical care. There is a need to evaluate patients' symptoms daily as

symptoms can significantly vary from one day to the next. Doctors and patients could make better healthcare decisions by having more information about the severity of the symptoms. The work presented in this thesis aim to fill that gap by using smartwatches to predict the severity of three symptoms: on/off, tremor and dyskinesia. The next chapter will present a review of the work that has been done in using mobile sensors to predict the severity of PD symptoms.

CHAPTER 2

LITERATURE REVIEW

This section will conduct a review of the work that has been done before in regards to using sensors to detect the severity of PD symptoms. It will focus on the three symptoms subject to this thesis's research questions: the on/off phenomenon, tremor, and dyskinesia. The studies reported in this review were found using Google Scholar, Connected Papers, and PubMed, using keywords like "Parkinson's disease," "mobile sensors," "medication status," "on/off," "tremor," "dyskinesia," "passive monitoring," "in-the-wild," "free-living."

Certain studies used the terminology "task-independent" meaning that Activities of daily living (ADL) are performed instead of standard MDS-UPDRS tasks. However, for this research, task-independent means that nothing is asked of the participants, and they are free to move for hour-long recordings. Consequently, ADL are considered task-dependent as precise tasks are required during the data collection. For example, participants will be asked without moving, sitting and talking, sitting and moving their hands in specific directions, brush their teeth, etc.

Passive monitoring is an additional challenge as there is no label identifying the subject's activity during the recording. For example, our system could mistakenly identify the hand moving as dyskinesia if the subject is running. Nevertheless, suppose the data was identified as a running task. In that case, more information is available to help the model make the right prediction and assess if the movement recorded is dyskinesia or not. Furthermore, two criteria make research in this field stand out:

- Is the evaluation performed at home or in a clinical setting?
- Is the subject instructed to perform specific tasks, or is it a free setting?

Many studies used sensors in a clinical environment with task-dependent (Salarian *et al.* (2007), Keijsers *et al.* (2003), Tsipouras *et al.* (2011)) or task-independent approaches (Salarian *et al.* (2007)). However, not many have tried at-home monitoring coupled with passive monitoring,

where no precise tasks need to be performed. Del Din *et al.* (2016) aims to review the literature on free-living monitoring of PD, but only found studies using scripted tasks or trying to simulate daily life through scripted tasks. This chapter is divided into the following: the features commonly used for similar research will be explained in section 2.1. Then, section 2.2 will review task-dependent approaches, while the following section 2.3 will review the few research using unscripted tasks.

2.1 Features

Previous work have used **frequency domain features** (i.e. peak power and frequency, signal entropy), **time domain features** (i.e. RMS amplitude, auto-correlation, mean values) or **both** to evaluate on/off, tremor and dyskinesia. More research has been done on tremor than on the other symptoms, as shown in Table 2.1. It is common in the literature that quantification and detection of tremor are done in the frequency domain as rest tremor typically has a frequency between 3 to 7 Hz (Keijsers *et al.*, 2006; Patel *et al.*, 2009; Gallego *et al.*, 2010). Some authors have also successfully used features like Mel-frequency Cepstral Coefficients (MFCCs), traditionally used in speech processing, for human activity recognition (San-Segundo *et al.*, 2016; Vanrell *et al.*, 2017).

The PDDB DREAM challenge focused exclusively on finding and establishing standard feature sets to evaluate tremor, dyskinesia, and bradykinesia from accelerometers and gyroscopes. For tremor, the best set of features were using **Frequency domain features** (Fourier spectrum's intensities at frequencies between 4 and 20 Hz (Sieberts *et al.*, 2020)), **time domain features** (the python package Time Series Feature extraction based on scalable hypothesis tests (Tsfresh)[1]), and the available clinical data. As this resulted in thousands of features, a Random Forest Regressor (RFR) selected the final 81 most important features from the set.

For dyskinesia, Schaff (2017); Sieberts *et al.* (2020) used **time domain features**. In addition to using the three axes of the accelerometer, they computed each axis's absolute values, meaning

[1] https://github.com/blue-yonder/tsfresh

Table 2.1 Overview of the features used in previous work

	Frequency Domain	**Time Domain**	**Frequency and Time Domain**
On/Off		Fisher *et al.* (2016a)	Keijsers *et al.* (2006); Ramji *et al.* (2017); Heijmans *et al.* (2019)
Tremor	Salarian *et al.* (2007); Rigas *et al.* (2009); Gallego *et al.* (2010); Joundi *et al.* (2011); Kim *et al.* (2018)	Rigas *et al.* (2016); Abrami *et al.* (2020)	Keijsers *et al.* (2006); Giuffrida *et al.* (2009); Patel *et al.* (2009); Mera *et al.* (2012); Tzallas *et al.* (2014); Hssayeni *et al.* (2019); Papadopoulos *et al.* (2019); Sieberts *et al.* (2020)
Dyskinesia		Keijsers *et al.* (2003); Tsipouras *et al.* (2010); Fisher *et al.* (2016a); Sieberts *et al.* (2020)	Patel *et al.* (2009); Tsipouras *et al.* (2011); Mera *et al.* (2012); Tsipouras *et al.* (2012); Tzallas *et al.* (2014)

the difference between each data point and the one before. The standard deviation, variance, and other statistical values like the minimum or maximal values, the median, and the sum were also computed from the three axes and the three absolute axes. Finally, the Boruta package and Recursive Feature Elimination (RFE) from the caret package in R was used to reduce the number of features and are both based on the Random Forest (RF) algorithm. The former was used first to select all features that had the potential to be helpful in predictions. Then, the latter was used to select a minimal set of features.

2.2 Task-dependent approaches

Electromyography and ambulatory systems

Task-dependent approaches, where the participant is asked to perform MDS-UPDRS Part III tasks or ADL, were the first attempts to quantify PD symptoms with sensors. Electromyography

(EMG) and electromagnetic devices were first used for long-term quantification of tremor with ambulatory monitoring (Foerster & Smeja, 1999), (Spieker *et al.*, 1998) (Bacher *et al.*, 1989). However, many electrodes would be needed to evaluate more complex movements from PD like dyskinesia (Tzallas *et al.*, 2014). With the developments of new sensors over the years, ambulatory systems started being developed, like the Kinesia (Giuffrida *et al.*, 2009) which uses a combination of sensors: EMG, accelerometers, and gyroscopes. This system's objective was to objectively quantify tremors instead of using the MDS-UPDRS, which is subject to clinicians' interpretation.

After being proved efficient in a clinical setting, the Kinesia was tested at home to assess tremor and bradykinesia (Mera *et al.*, 2012). The same scripted tasks from the MDS-UPDRS Part III were performed 3-6 times a day for 3 to 6 days. They concluded that objective home monitoring of PD symptoms was feasible. However, only ten patients were part of the study, and the Kinesia device is worn on the finger and wrist, so it is not convenient to wear daily.

Accelerometers and gyroscopes on the body

Accelerometers and gyroscopes units started to be used as they are small and do not limit movements. Many studies used multiple sensors placed at different positions on the subject's body for on/off (Keijsers *et al.*, 2006; Ramji *et al.*, 2017), tremor (Salarian *et al.*, 2007) and dyskinesia (Tsipouras *et al.*, 2010, 2012). Some studies used as many as eight sensors: two on each leg and two on each arm (Patel *et al.*, 2009) to monitor tremor and dyskinesia. One could argue that using many sensors might provide better results for a symptom like dyskinesia where the entire body can be involved, some even describing it as "dance-like movements of the body parts" (Tsipouras *et al.*, 2012). Patel *et al.* (2009) also suggested that accelerometers do not capture patterns that are specific to MDS-UPDRS tasks and that accelerometers could be used to estimate the severity of symptoms from ADL. However, wearing eight sensors might not be comfortable enough to be used for long hours of monitoring and requires to managing the battery level of many devices.

Keijsers *et al.* (2006) used a neural network to classify the on/off state of participants while imitating daily life tasks and achieved a sensitivity and specificity near 0.97. Using a system of 6 accelerometers, they could automatically assess the motor state of PD patients. A decade later, (Ramji *et al.*, 2017) used a combination of sensors (accelerometers, gyroscope, and a magnetometer) to predict the disease state with a Support Vector Machine (SVM) and obtained an accuracy of 78%.

Until now, most studies quantifying dyskinesia placed multiple sensors on the body, from 4 to 8 (Keijsers *et al.*, 2003; Patel *et al.*, 2009; Tsipouras *et al.*, 2010, 2011, 2012). Using various models like a SVM, Artificial Neural Network (ANN) and others, they concluded positively about quantifying levodopa-induced dyskinesia with high accuracy or high correlation to clinician MDS-UPDRS ratings.

Accelerometers and gyroscopes on wrists sensors

In an effort to use less units, wrists sensors with accelerometers, gyroscopes, or both started being used to assess tremor while participants were performing selected tasks (Gallego *et al.*, 2010; Tsiouris *et al.*, 2017; Hssayeni *et al.*, 2019) or at rest (Kim *et al.*, 2018). However, we found only one study using wrist sensors for on/off and dyskinesia, in both clinical settings and at home (Fisher *et al.*, 2016a).

Various machine learning models were used to try and quantify tremor accurately: from Decision Tree (DT) to Convolutional Neural Network (CNN), SVM, and Long Short-term Memory (LSTM). Kim *et al.* (2018) trained a CNN on accelerometers signals and compared their model with other classifiers used in previous studies and showed that their model outperforms them. They used 92 subjects, which is a considerable amount of data.

Many studies have been conducted using commercial grade accelerometers and gyroscopes sensors (Rigas *et al.*, 2009; Tsipouras *et al.*, 2010; Gallego *et al.*, 2010; Kim *et al.*, 2018). However, they are done in a controlled environment and are not easily accessible to the population. On the other hand, the efficiency of using accelerometers and gyroscopes has already been

proven in those environments, so an exciting avenue of research uses those sensors, but in devices that everybody already has at home and are far more accessible: smartwatches and smartphones.

Smartphones and smartwatches

In 2011, Joundi *et al.* (2011) suggested, in a clinical setting, that the accelerometer of iPhones can be used to assess the dominant frequency component in tremor. The iPhone was strapped to the tremorous limb and recorded with an application for 30 seconds, concurrently with EMG recordings. The iPhone score matched the score of the more sophisticated EMG analysis. However, the smartphones' accelerometers are not medical-grade, and it is also not convenient to have a phone strapped to a leg, so it is still not a long-term solution for monitoring tremor.

In contrast with other studies who were using sensors from the smartphone or smartwatches directly on a limb, Zhan *et al.* (2018) used the HopkinsPD app on the smartphone to ask subjects to perform tasks: voice (say aaah for twenty seconds), finger tapping (alternating between the index and the middle finger following a regular rhythm), gait (walk straight and come back inside a twenty seconds window), balance (stand without help for twenty seconds), and reaction time (press and hold a button as soon as it appears on the screen). From the completion of those tasks, they assessed a score that they created, the Mobile Parkinson Disease Score (mPDS), about the severity. 129 individuals participated in their study and their mPDS measure correlated well with the MDS-UPDRS score ($r=0.81$; $P<.001$). The mPDS is built to be used complementary to other standard PD measures as a way to measure PD symptoms burden rather than focus on one symptom, as MDS-UPDRS-Part III focuses gives more weight to tremor than other motor symptoms.

Papadopoulos *et al.* (2019) used smartphones at home to monitor tremor. They collected accelerometer data from smartphones when subjects with PD were on a phone call. Therefore, only the simple task of being on the phone was required to start collecting data. However, this allows collecting data at random times during the day for a limited time. In order to get labels for the data, they used multi learning instances and represented the subjects from windows of

data to which they assigned a label. The dataset consisting of 45 subjects for this study was made available with open access.

There are many benefits to monitoring symptoms with wearable sensors. They are very accessible to the general public, economical as they are not as expensive as commercial-grade sensors, and they have the potential to monitor movements for long periods. They also impose no constrain on the individuals, as they only have to wear the device, very low friction, and easy compliance to the technology.

However, there are also some challenges. As people with PD are often older, they might not be used to the technology, although studies have shown good acceptance (Fisher *et al.*, 2016b; López-Blanco *et al.*, 2019). It is also necessary to make sense of the data, make it clinically valid, and translate the recordings to an actual identification of the symptoms' severity, which is a significant challenge. Nevertheless, it is the first step required before these wearable devices can prove that their results can guide clinicians for better medication.

Furthermore, most studies instruct patients to perform specific tasks to simulate daily living or tasks required in MDS-UPDRS-Part III. Del Din *et al.* (2016) is a survey of free-living monitoring and covers many studies, but again, tasks are scripted during the collection of data. Therefore, they do not answer the research question correctly as the participants are not free of doing what they want. However, with the development of smartwatches and smartphones that already register accelerometer and gyroscope data, a new exciting and promising research direction is to assess if those accessible devices could be used for at-home monitoring and longitudinal recordings, where the subjects are not expected to perform any specific tasks. The following section will review the work and progress that has been done in passive monitoring of PD.

2.3 Task-independent approaches

Benefits of passive monitoring

Very little research has been done in task-independent settings for passive monitoring of PD. We found only a few papers meeting our criteria of monitoring patients at home, with unscripted tasks from mobile sensors. It is still a challenging area of research (Espay *et al.*, 2016). More research was also done on tremor (Rigas *et al.*, 2016; Salarian *et al.*, 2007; Abrami *et al.*, 2020) then the other symptoms studied in this thesis. However, there are many benefits to free-living monitoring:

- The symptoms of PD are often triggered by the task the subject needs to perform and by free-living environment challenges that can not be replicated in a controlled environment (Del Din *et al.*, 2016);
- No observer bias. Clinicians can have a bias when they expect to see a certain behavior;
- Avoid differences between evaluators. For example, more senior specialists typically assign lower scores than younger colleagues (Post *et al.*, 2005);
- Possibilities for longer monitoring of the symptoms and how they fluctuate, which can lead to better healthcare decisions and better dosage of medication;
- No attention compensation. During a UPDRS test, the subject knows that they're being evaluated, so they can unconsciously try to compensate.

As authors disagree on the definition of free-living monitoring, for this work, free-living means that the participants are not required to do any tasks and are entirely free in their movements.

A summary of previous task-independent work is presented in Table 2.2.

Table 2.2 Comparison with other work presented in the literature for task-independent research

Author	**Sensors**	**Symptom**	**Features**	**Results**
Salarian *et al.* (2007)	Gyroscope on each forearm	Tremor	Frequency domain	"High correlation to the MDS-UPDRS tremor subscore (r=0.87, p<0.001 for the roll axis)"
Rigas *et al.* (2016)	Accelerometer and gyroscopes on the wrist	Tremor	Time domain	Accuracy: 94%
Fisher *et al.* (2016a)	Bilateral wrist-worn accelerometers	On/Off and Dyskinesia	Time domain	**Dyskinesia** Specificity: 0.93 Sensitivity: 0.38 **On/Off** Sensitivity sub-optimal
Heijmans *et al.* (2019)	Three accelerometers and gyroscopes, one at each wrist and one on the chest	On/Off	Time and frequency domain	Feasible method
Abrami *et al.* (2020)	3-axis accelerometer on the wrist of the non-dominant hand	Tremor	Time domain	Feasible method
Our work	3-axis accelerometers on the wrist of any hand	On/Off, Tremor, Dyskinesia	Time and frequency domain	**MSE** On/Off: 1.129 Tremor: 0.429 Dyskinesia: 0.462

Gyroscopes on wrist

Salarian *et al.* (2007) used gyroscopes on each forearm to quantify tremor and performed two clinical studies. One was using scripted ADL, while the other was in a clinical setting, but subjects were free to move for 3-5 hours. They found that accurate tremor assessment can be achieved in a free-moving setting during their daily activities. However, they had very few subjects (10), and they were all males. Furthermore, patients were in a hospital, so they did not

need to perform the same tasks one would in a comfortable home environment. However, this result is promising for future work.

For tremor again, Rigas *et al.* (2016) used gyroscopes and accelerometers on one wrist and found low false positive. However, they were training their decision tree with simulated datasets, and the free-living dataset only consisted of one patient with tremor. More data should be collected to assess if their method can also provide good results for a larger population. Additionally, approaches trained on laboratory data do not represent well wild data and may decrease the performance (Zhang *et al.*, 2020).

Accelerometers on wrist

Fisher *et al.* (2016a), and Heijmans *et al.* (2019) both measured on/off using accelerometers at the wrists and did not come to the same conclusion. The former concluded that "Accurate, real-time evaluation of symptoms in an unsupervised, home environment, with this sensor system, is not yet achievable." However, Heijmans *et al.* (2019) argued that the sensor data "could reliably predict subjectively reported OFF moments." Different reasons might explain the divergence in opinions: the sensors used, the diary technique, and the model's architecture. The following paragraphs will cover and explain the differences.

In addition to wrist sensors, Heijmans *et al.* (2019) used a chest sensor. This additional sensor might have made the difference in being more accurate to predict on/off moments. There is no agreement in the literature about how many sensors would be optimal to use. However, it was surprising to learn that participants found the chest sensor more comfortable to wear than the wrist ones. Considering the excellent acceptability of the chest sensor and the promising results, it could be a good compromise to use a chest sensor in addition to a wrist one. It could be more promising in trying to predict lower limb symptoms while being minimally invasive. Another possibility is to use ankle and wrists sensors on the most affected side (Hssayeni *et al.*, 2019) as PD symptoms often begin on one side of the body (Fahn *et al.*, 2003).

When patients are monitored at home, an additional challenge is establishing a ground truth that would label the symptoms' severity during the recordings. In a clinical setting, clinicians observe patients or video recordings and rate their symptoms in compliance with the MDS-UPDRS. This is not possible anymore at home. One way to address this is to use diaries that patients can complete at home while monitoring their symptoms. Typically, subjects have to rate their symptoms on a scale of 0 (none) to 4 (severe) every 30 minutes. This technique of using diaries to establish a ground truth has some limitations. The ratings become very subjective to how each patient perceives their symptoms. Additionally, the subjects can often complete diaries at a later time. Therefore, they might not accurately remember the severity of the symptoms they experienced earlier during the day and introduce a recall bias (Myin-Germeys *et al.*, 2009). Another way to establish labels is to use weekly supervised approaches (Zhang *et al.*, 2017a).

Furthermore, both studies used different techniques for diary entries. Fisher *et al.* (2016a) required participants to fill diaries every hour to reduce the burden for the participants to complete. Consequently, it can introduce errors as a patient can experience fluctuations in the disease state in a single hour. Heijmans *et al.* (2019) instead used an Experience Sampling Method (ESM) to address the limitations of diaries. ESM is a digital diary method that, instead of pre-defined intervals, is done at semi-random times during the day. It also requires fewer diaries for the participants to complete, with one in the morning, one at night, and seven at random times during the day. There are many more diaries to complete in a setting where patients have to complete diaries every 30 minutes for 48 hours. Heijmans *et al.* (2019) concluded that ESM could be used to predict on/off state accurately. However, they only evaluated one patient out of the entire population. They used a logistic regression classifier and were able to predict reliably on/off moments. They do not give any evaluation metric measures, so it is impossible to quantify how reliable their results are. Further analysis should be done to evaluate feasibility with more subjects.

Additionally, Fisher *et al.* (2016a) used an ANN to predict the probability of the recording having either "asleep", "off", "on", or "dyskinesia". The class with the higher probability was chosen as the prediction for what the recording contained. However, this conception choice is

questionable because people with PD can experience being both on or off medication while experiencing dyskinesia, while their system only allows one or the other. The authors argue that a reason explaining their unsatisfying results might be because patients can experience lower limb dyskinesia, which would not be picked up by sensors on the wrists.

Recent advances

Free-living work focused on extracting features from 15 to 20 minutes window of sensor recordings between the time subjects filled diaries. However, Abrami *et al.* (2020) recently proposed a different approach and leverages how human motor systems rely on discrete movements that can create patterns. They used k-means in an unsupervised manner to separate movements into a series of basic motions called syllables (see Figure 2.1). Healthy subjects perform similar patterns when walking. For example, however, they found that subjects with PD show more disorganized patterns than healthy subjects.

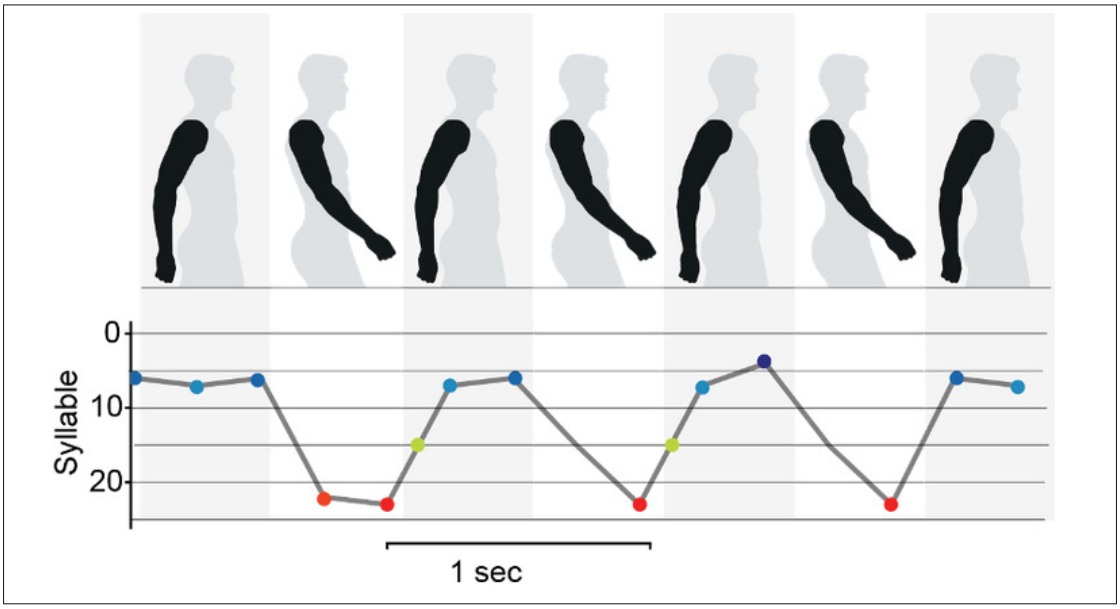


Figure 2.1 The movement of the arm while walking can be translated to a sequence of movements
Taken from IBM (Abrami *et al.*, 2020)

The work presented in this thesis goes a step further than these studies and performs passive monitoring in an uncontrolled environment for a prolonged time (6 months). The 20 minutes window recordings we are working with are done when the subject enters the severity of their symptom in the diary. However, there is no identification of any task the subject performed in that window of time. Parkinson@Home study concluded in 2017 that "it is feasible to collect objective data using multiple wearable sensors in PD during daily life in a large cohort." (De Lima *et al.*, 2017). Therefore, the next step is to use this data and make it clinically valid.

A startup from Germany, MedEngine, is also attempting to solve this challenge. They created their app, Flytta, and engineered their smartwatches to capture motor and non-motor symptoms of PD. In addition, they provide people with a dashboard that identifies trends and patterns in their condition.

2.4 Conclusion

In conclusion, most of the work so far that aims to objectively quantify PD symptoms has been studying subjects while they were performing scripted tasks, either from MDS-UPDRS or from ADL. The next challenge is to develop biomarkers that can be used to assess the symptoms during passive monitoring when subjects are at home and free to do what they need to during the day. We address this gap by using data that was collected at home using a smartwatch. This work aims to accurately predict the severity of the on/off phenomenon, tremor, and levodopa-induced dyskinesia. We do so by using wearable sensors and passive monitoring as they offer significant benefits. Wearable devices are accessible, comfortable, and cheaper. However, it is not easy to make them clinically valid. Passive monitoring allows for more extended monitoring of the symptoms and a better understanding of how they fluctuate. The assessment is objective, so it also avoids attention compensation.

CHAPTER 3

METHODOLOGY

This chapter introduces the methods used to answer the three research questions: how to predict the on/off medication status, the severity of tremor, and dyskinesia from data collected with accelerometers from wearable sensors. First, the databases will be introduced in section 3.1, followed by the data pre-processing steps in section 3.2. Then, we will introduce the three main approaches we explored in parallel. Figure 3.1 includes a diagram of the three approaches analyzed in this thesis:

- Approach I : Tsfresh package to extract features and gives those features to an XGBoost classifier (section 3.4),
- Approach II : different embeddings and backends (section 3.5),
- Approach III : a fusion of those previous two approaches (section 3.6).

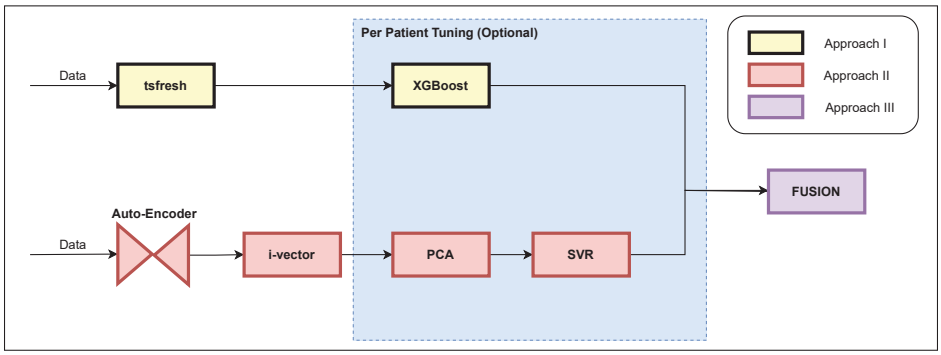


Figure 3.1 General pipeline of the three main approaches

3.1 CIS-PD Database

The database we used for this work is called Clinician Input Study (CIS-PD) and was obtained from the Biomarker & Endpoint assessment to track Parkinson's disease (BEAT-PD) DREAM

Challenge. For up-to-date information on the study, visit the challenge website [1]. The database collected triaxial accelerometer data from mobile sensors from subjects with PD in their daily life, at home. Figure 3.2 shows the system of the axis of the smartwatch. Participants were not asked to do any specific task. They only had to wear an Apple Watch Series 2 on any wrist and for at least 12 hours per day.

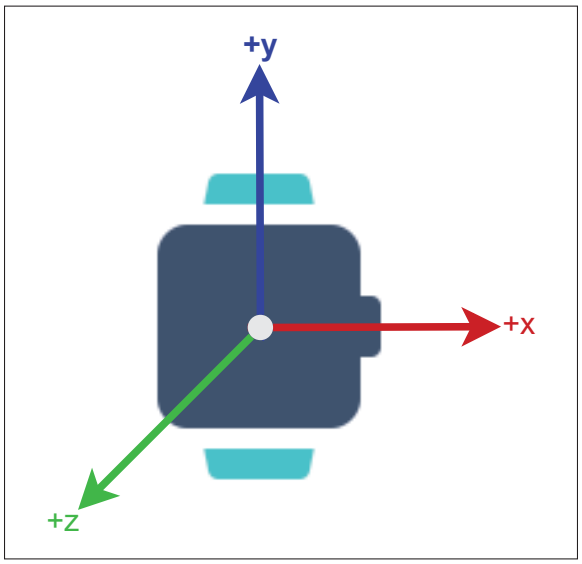


Figure 3.2 Representation of a smartwatch and the system of axis

The CIS-PD database was first proposed in Elm *et al.* (2019). Fifty-one patients were recruited, but only 39 completed the whole six months. The patients were men and women over 18 years old and experiencing severe symptoms of the disease.

There were four clinic visits during the study, following the timeline shown in Figure 3.3. Forty-eight hours before these visits, participants had to report their symptoms every 30 minutes in a diary that was a mobile application called Fox Wearable Companion App Michael J Fox Foundation (2016).

[1] https://www.synapse.org/!Synapse:syn20825169/wiki/600405

At the two weeks visit, the MDS-UPDRS Part III evaluation was performed twice. The first time was OFF medication when the last dose was at least 12 hours before. Then, the participant took their medication, and the evaluation was made ON medication an hour later.

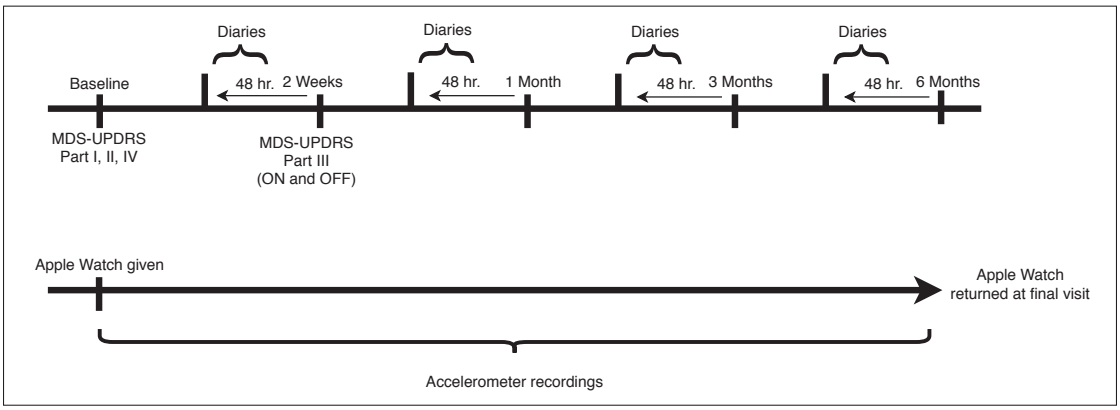


Figure 3.3 Timeline of the collection of data for CIS-PD
Taken from BEAT-PD Challenge Webinar (2020)

The participants had to report their symptoms in diaries every 30 minutes for 48 hours before the clinic visits, except when they were asleep. They could also complete diaries from the past if they did not rate their symptom right away. In Appendix I, Figure I-1 shows an example for dyskinesia. The participants had to rate their symptoms on a scale of 0 to 4, and the corresponding severity is in Table 3.1.

Table 3.1 Required ratings filled
by participants in diaries

Labels	**On/Off**	**Tremor, Dyskinesia**
0	Full ON	None
1		Slight
2		Mild
3		Moderate
4	Full OFF	Severe

For the challenge, the organizers selected 16 subjects to be included in the training and testing set. Five more subjects were determined as unsuitable for the test dataset but were available in an ancillary data subset. The number of recordings per patient is shown in Appendix II-1. It

varies considerably, from 34 files in the training subset to 299. Additionally, not all subjects had data for the three sub-challenges, so on/off had 15 subjects available, tremor had 13 subjects, and dyskinesia, 11.

Some additional clinical data was also provided. Demographics data, age, gender, race, and ethnicity is provided, as shown in Table 3.2. The scores of the MDS-UPDRS Part I, II, and IV (4.1, 4.2, 4.3, 4.4, 4.5, 4.6) are also provided. These were taken at the start of the study as a baseline. MDS-UPDRS Part III scores are also provided at the two-week visit while the subjects are on and off.

Table 3.2 Demographics information for CIS-PD

	CIS-PD
Nb of subject_id	16
Nb of female	5
Nb of male	11
Race	15 White 1 Unknown
Age average (std deviation)	62.8125 (10.857)

3.2 Data Pre-Processing

Before modeling, the first step was to analyze and visualize the signals like in Figure 3.4. Then, we noticed that some measurements had very straight lines. It could mean that the subject had to remove their watch or that they were completely still. Furthermore, accelerometers measure both the orientation of the acceleration and a gravity component (Suzuki *et al.*, 2017). Therefore, we removed the gravity component using a HPF and removed the inactivity segments. More details about the implementation of both techniques are available in section 3.2.1 and 3.2.2.

As a result of those manipulations, four different kinds of variation on the dataset can be used as input, as shown in Figure 3.5:

- **Original Training Data**: No data pre-processing,

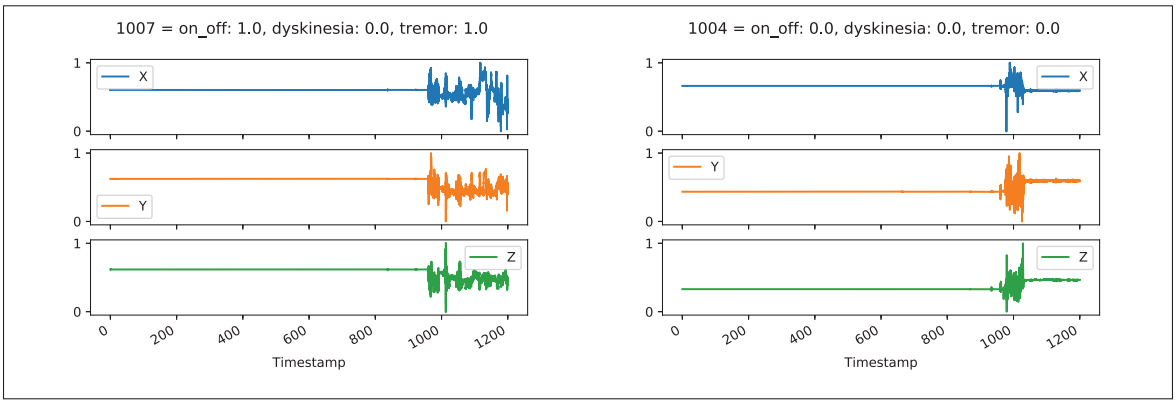


Figure 3.4 Two plots showing accelerometers data from two measurements

- **High-Pass Filter**: Apply the HPF on the original data to remove the offset,
- **High-Pass Filter + Inactivity Removed**: Removes the gravity offset and inactivity,
- **Original + Inactivity Removed**: Remove the inactivity from the original data, without applying offset removal.

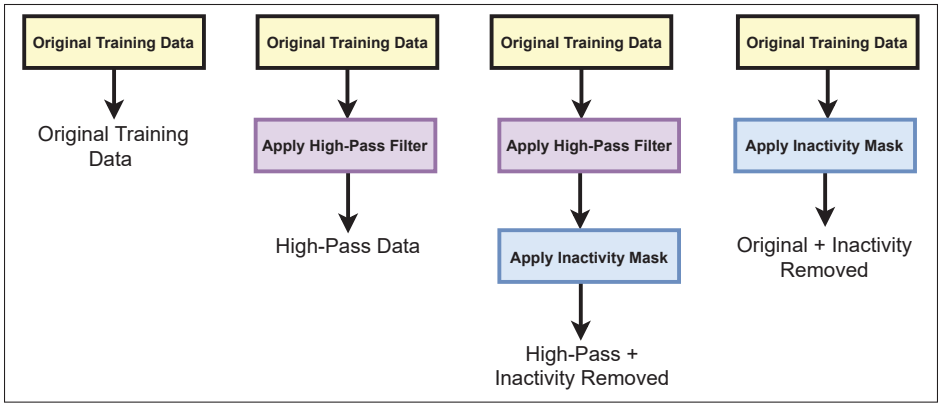


Figure 3.5 The 4 different data pre-processing options

3.2.1 High-Pass Filter

The HPF is used to remove the gravity component (offset) in the time-series as the mean amplitude of the signals was not centered at zero (Figure 3.6a). Since the gravity component has a lower frequency than human movements, a high-pass filter attenuates those lower frequencies

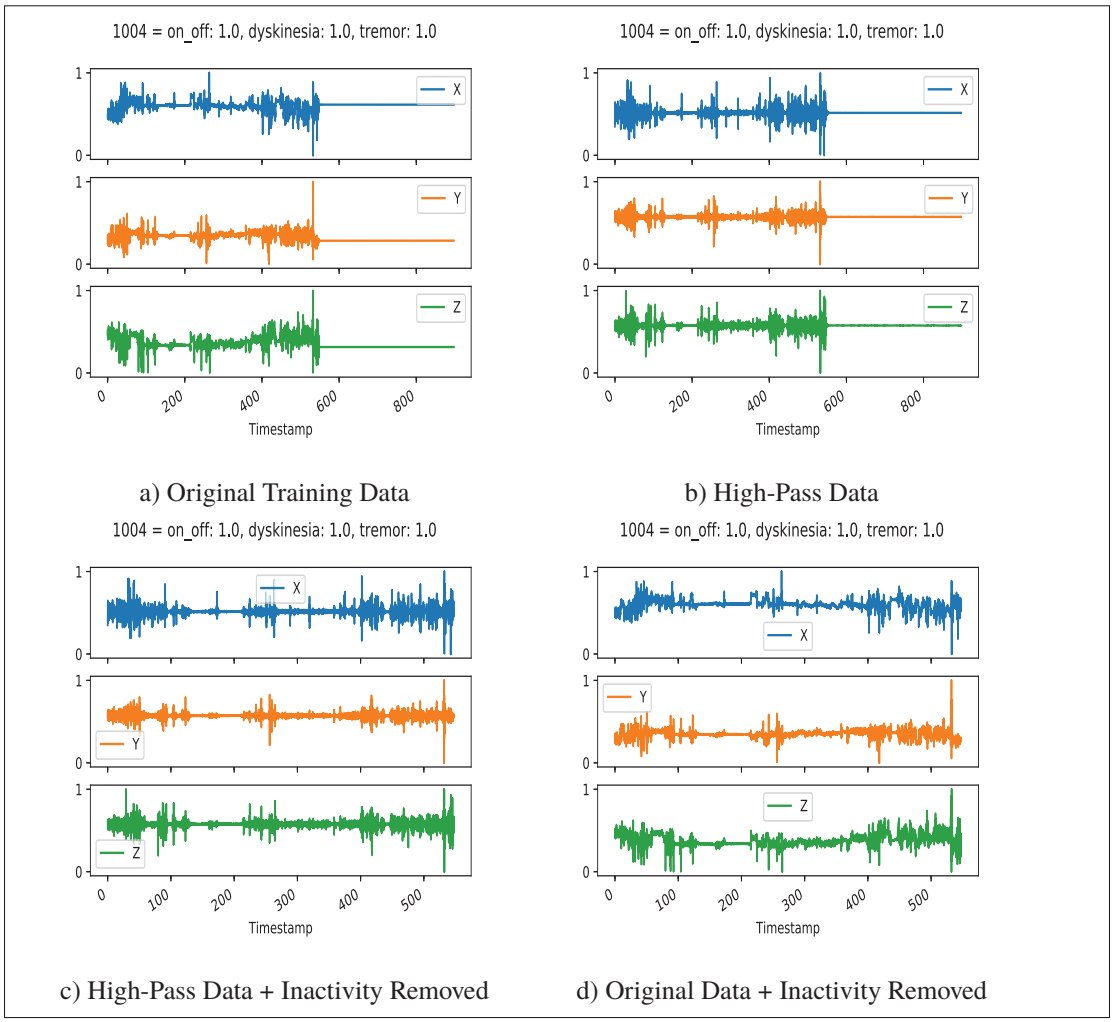


Figure 3.6 Pre-Processing results on the accelerometer data of the CIS-PD database

than a specific cutoff and passes signals with higher frequencies. Additionally, the HPF removes low sinusoids from the signal.

We used a cutoff of 0.5 Hz, with an order of 10 and a sample rate of 50 Hz (one data point every 0.02 seconds). An example of the high-pass frequency response and the filtered signal is in figure 3.7.

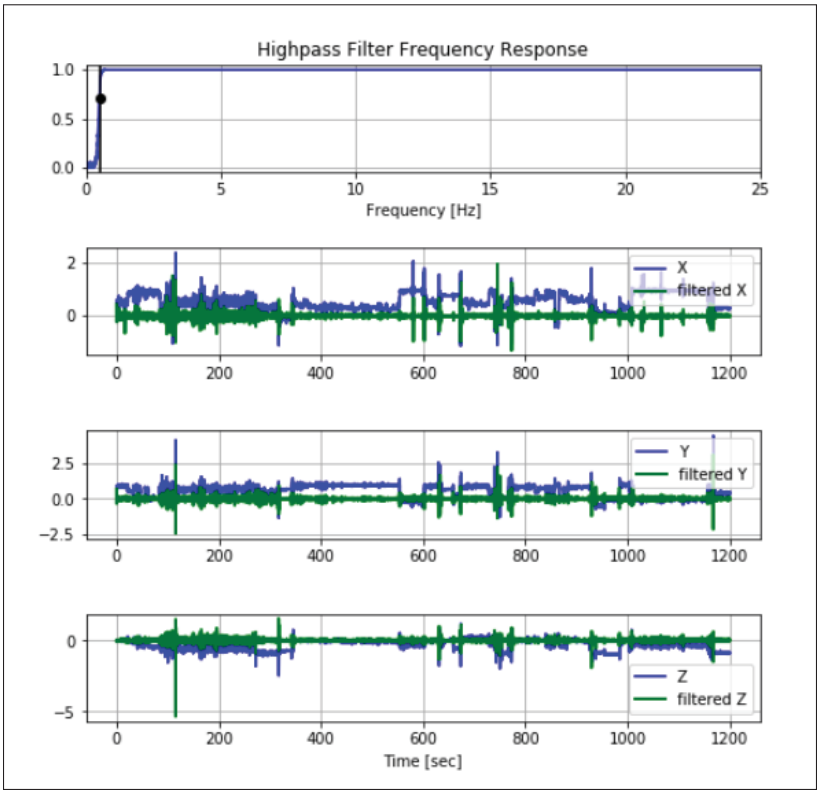


Figure 3.7 High-pass filter frequency response and the result on the three axis of the signals

3.2.2 Inactivity removal

As the model's objective is to predict symptoms where movement is involved, we identified regions of inactivity to remove them. We consider that these regions do not contain any information about movements. First, we apply a hpf to remove the offset. This step was necessary because we use the energy of the signal to detect inactivity. Therefore it had to be centered around zero. Then, we identified and removed inactivity by creating masks. Figure 3.8 explains these steps.

For a segment to be identified as inactivity, two conditions needed to be met:

- Condition 1: $energy\_of\_measurement < energy\_threshold$

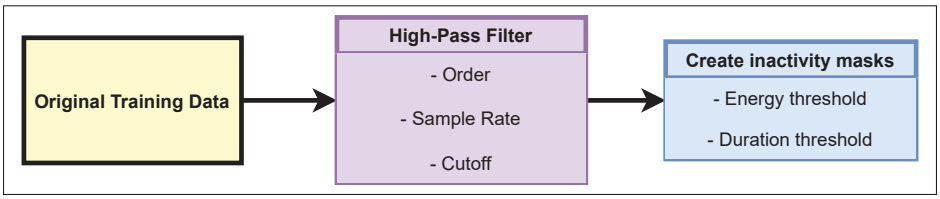


Figure 3.8 Steps and parameters to create inactivity masks that allow inactivity removal

$energy\_threshold$ is the value of energy above which the frame is considered to have activity. It is a percentage of the maximum energy level recorded in the measurement. For example, if $energy\_threshold = 10$, then the threshold for each axis of the accelerometer will be 10% of that max value. Candidates to inactivity removal become every value that's lower than that threshold;

- Condition 2: $duration\_of\_inactivity > duration\_threshold$

  $duration\_threshold$ is the minimum number of consecutive inactivity candidates before the sequence will be indeed removed and confirmed as inactivity. We removed sections that are at least 1 minute long of inactivity detected. If it is shorter than 1 minute, we do not consider it as inactivity as it is not long enough, and we keep the values.

3.3 Cross-validation with 5 folds

We performed a 5-fold cross-validation per patient. We divided all the subjects' measurements into five groups that became the training and development sets for this work. As CIS-PD is a multi-label dataset where one sample belongs to more than one class, it is a challenge to balance equally the groups for all the labels. Therefore, we used on/off as the unique label to balance them. However, there were some exceptions. For example, in CIS-PD, subject 1046 did not have any labels for on/off. In that case, we used the labels for tremor.

3.4 Approach I - Time-series features

The first approach uses the python package Tsfresh (Christ *et al.*, 2018) to extract features from the signal, followed by an XGBoost to do regression. Tsfresh extracts statistical data from the signal that was successfully used before in similar challenges (Wang *et al.*, 2017). We chose XGBoost because it handles well the tabular data coming from tsfresh data extraction. Besides, DT can select important features and combine them to make robust predictions. However, an imbalanced dataset can affect the performance.

3.4.1 Feature extraction

The Tsfresh, NumPy, and scipy libraries were used to extract 43 unique features from the three axes of the original signal (X, Y, Z). We also used them on three additional signals that we computed with the absolute difference between the samples i and noted as $\Delta X, \Delta Y, \Delta Z$ (see equation 3.1). These engineered axes were inspired by Schaff (2017), who used them successfully for research on tremor and dyskinesia.

$$\begin{aligned} \Delta X[i] &= |X[i+1] - X[i]| \\ \Delta Y[i] &= |Y[i+1] - Y[i]| \\ \Delta Z[i] &= |Z[i+1] - Z[i]| \end{aligned} \tag{3.1}$$

From the three axis and computed signals $(X, Y, Z, \Delta X, \Delta Y, \Delta Z)$, we extracted features using Tsfresh (Christ *et al.*, 2018). The name of the library stands for Time Series Feature extraction based on scalable hypothesis tests. It is a python package that automatically extracts features from time series. The following lists the main features we extracted, and an exhaustive list of all features and their parameters is available in Appendix III-1:

- Autocorrelation: Measures the correlation and if there is a linear relationships between values that are k time periods apart;

- c3: Measures the third-order autocovariance. It is a measure of non-linearity in time series; (Schreiber & Schmitz, 1997)
- fft_coefficient: Computes the Fourier coefficients. It can return the real, absolute, imaginary part and the angle in degrees;
- Longest strike below mean: Measures the longest strike of values that are below the mean;
- Number of peaks: Computes the number of peaks where a value x is higher than its n neighbors to both left and right;
- Spkt welch density: Estimates the cross power spectral density of the time series at different frequencies;
- Agg autocorrelation: Calculates the mean, variance, standard deviation, or median of the autocorrelations;
- Max langevin fixed point: Largest fixed point of dynamics;
- Linear trend: Calculate a linear least-squares regression for the values of the time series versus the sequence from 0 to length of the time series minus one. (Christ *et al.*, 2018)

In addition to Tsfresh, we also extracted the following features with the NumPy and scipy libraries:

- Mean: The mean value of the axis,
- Max: The max value observed in the axis,
- Standard deviation: Measures the spread of the distribution,
- Variance: Measures the spread of the distribution,
- Peak to peak (ptp): Range of values along the axis ($maximum - minimum$),
- Percentile: Computes the q-th percentile,

- Skew: Measures the skewness of the axis. If it is normally distributed, the skewness would be close to 0. In other words, if the axis is asymmetric, the value would be high,
- Kurtosis: Measures if the axis has a heavy tail or a light tail when compared to a normal distribution,
- Kstat: Computes the nth k-statistic,
- Moment: Measures the shape of the axis, related to skewness and kurtosis.

3.4.2 XGBoost

XGBoost is a DT based algorithm that was introduced by Chen & Guestrin (2016). To better understand the XGBoost, this section will explain the model's evolution from basic DT to the XGBoost, and then we will introduce the architecture and training methods of the algorithm in this work.

Evolution to the XGBoost

First, bagging was the first improvement to a basic DT. Also called bootstrap aggregation, an ensemble method combines predictions from multiple algorithms by doing an average of their predictions or choosing the majority output. As DT have a high variance due to their sensitivity to the specific data they are trained on, bagging aims to reduce the variance by creating several subsets from the initial dataset. Each subset will be used to train a single model that will be later combined with the average. As a result, each DT will have a low bias (trees are deep and not pruned) and high variance (as it is sensitive to the subset). Therefore, when averaging the predictions, it reduces the variance while keeping a low bias.

Random Forest improved bagging: it makes sure the individual trees are different by randomly selecting features a node can use to split. This helps in the case where one feature is good at predicting the class of the samples. In that situation, all the individual trees could choose this feature as the root node, making the trees all similar. It is not an acceptable behavior because it

will reduce the variance of the trees. Therefore, when performing the average, it will not be reduced as much. It is called a RF because it is made of DT that are random in both features and subsets.

The next improvement to RF is boosting. It is another ensembling technique that trains trees one at a time. At each tree (using random subset and feature selection), the goal is to improve the model and reduce the error by increasing the weight of misclassified examples so that the next estimator is more likely to classify it correctly. Consequently, the next improvement to boosting is called Gradient Boosting (GB) and uses gradient descent to minimize the error.

Finally, the XGBoost is based on all these previous improvements, and its strength resides in improving the GB algorithm with six key points:

1. Parallel processing to build trees,
2. Tree pruning: GB used a greedy criteria to stop the training. XGBoost uses a max_depth criteria,
3. Handles missing values: the algorithm automatically learns what is the best value for the data that is missing,
4. Regularization with L1 and L2,
5. Cross-Validation is built-in,
6. Hardware optimization.

The XGBoost has many hyperparameters one can tune to get the best results. The authors have divided them into three categories:

1. **General parameters:** what kind of booster to use, level of verbosity, number of threads, and more;
2. **Booster parameters:** parameters related to the individual booster (tree);

3. **Learning task parameters:** specifies, for example, what kind of regression to perform (with the objective parameter that defines the loss function), random seed number, or which evaluation metric to use for the validation step.

The following non-exhaustive list of booster parameters explains the key hyperparameters that we used to tune the XGBoost in this work:

- max_depth: the maximum depth of a tree;
- learning_rate: after each boosting step, it controls how much the GB weights are updated in the next iteration to converge to the minimum loss;
- subsample: subsample ratio of the training instances. It is the parameter that enables the random subset of training examples at each iteration;
- colsample_bytree: subsample ratio for the features at each tree. This introduces the random selection of features for each tree;
- colsample_bylevel: subsample ratio at for the features at each level in the tree. This creates a random selection of features at each level of the tree and is chosen from the set of features chosen for the current tree;
- min_child_weight: the minimum sum of instance weight needed in a child. If a leaf node has less instance weight than the value of this parameter, the partitioning stops there;
- gamma: minimum loss reduction needed to split a node and go deeper;
- reg_lambda: L2 regularization term on weights;
- n_estimators: the number of decision trees in the XGBoost.

As the RFR is a precursor of the XGBoost, they share many hyperparameters. However, the former has a few different ones, for example:

- min_samples_split: the minimum number of samples required to split an internal node,

- min_samples_leaf: the minimum number of samples in a leaf node.

XGBoost in this work

As a pre-processing step, we centered the training data by removing the subject-specific mean of each feature and the mean label (equation 3.2). Then, after getting predictions from the model, each subject's mean label was added to the output to obtain the final predictions (equation 3.3).

$$\begin{aligned} x_k &= x_k - mean(x_k) \\ y_k &= y_k - mean(y_k) \end{aligned} \tag{3.2}$$

$$\hat{y}_k = \hat{y}_k + mean(y_k) \tag{3.3}$$

where:

x_k = features of a subject k
y_k = labels of a subject k
$\hat{y}_k$ = predictions from the model specific to a subject k

For this approach, we experimented with two different configurations. The first configuration is called **Tsfresh + XGBoost Everyone**. We trained one model for all subjects on the entire training set. We optimized the hyperparameters over the data from all patients with a grid search (the different values we tried are listed in Table 3.3). Therefore, all the subjects shared the same hyperparameters. We added a one-hot encoder identifying each subject to the set of features. Figure 3.9 shows the architecture for this approach as well as the final hyperparameters used.

We also experimented with using a RFR for **Tsfresh + RFR Everyone** instead of an XGBoost as they are less affected by imbalanced datasets and are easier to tune. We also performed a grid search for this algorithm, and the values we tried are in Table 3.4.

Table 3.3 Grid search performed on the hyperparameters for the approach I when using an XGBoost. The best hyperparameters for Tsfresh + XGBoost Everyone are in bold

Hyperparameters	Values
objective	**reg:squarederror**
max_depth	**2**, 3, 4, 5, 6
learning_rate	0.001, 0.01, 0.05, 0.1, 0.2, **0.3**
subsample	0.5, 0.6, 0.7, 0.8, 0.9, **1.0**
colsample_bytree	0.4, 0.5, 0.6, 0.7, **0.8**, 0.9, 1.0
colsample_bylevel	0.4, **0.5**, 0.6, 0.7, 0.8, 0.9, 1.0
min_child_weight	**0.5, 1.0, 3.0, 5.0, 7.0, 10.0**
gamma	0, 0.25, 0.5, **1.0**
reg_lambda	0.1, 1.0, 5.0, 10.0, 50.0, **100.0**
n_estimators	50, **100**, 500, 1000

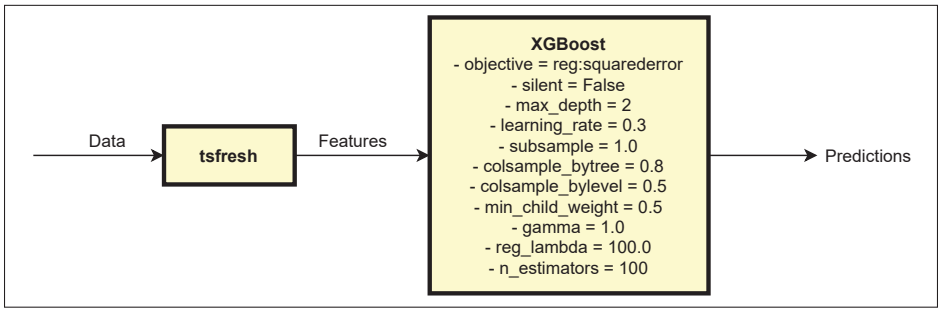


Figure 3.9 Architecture of approach I for Tsfresh + XGBoost Everyone

Table 3.4 Grid search performed on the hyperparameters for approach I when using a RFR. The best hyperparameters for Tsfresh + RFR Everyone are in bold

Hyperparameters	Values
n_estimators	50, 65, 95, 100, 500, **1000**
max_depth	**2**, 3, 4, 5, 6, 8, 15, 25, 30
min_samples_split	2, 5, 10, **15**, 100
min_samples_leaf	**1**, 2, 5, 10

The second configuration was **Tsfresh + XGBoost Per Patient** (Figure 3.10). In this configuration, the XGBoost is trained only on the data specific to each patient. The hyperparameters are optimized for each patient, so they vary from one patient to another. As we are working on disease-related data, we expected that the signals and the labels are very subjective and differ significantly from one patient to another as each subject's condition is unique. Therefore, tuning per patient might lead to modeling the severity of that particular subject's symptoms more accurately.

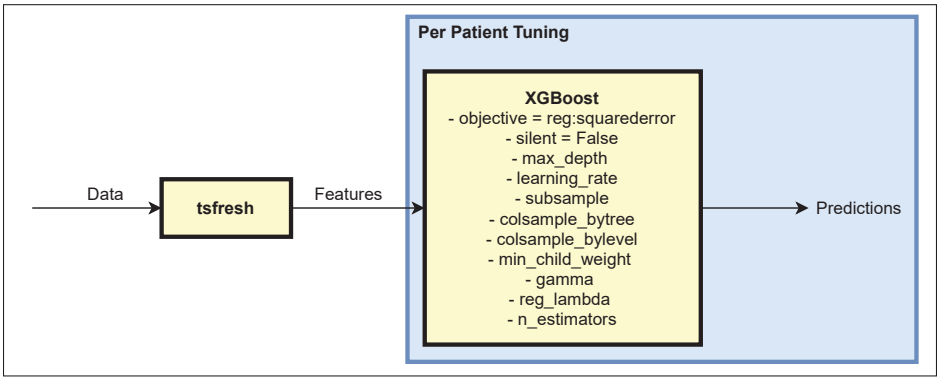


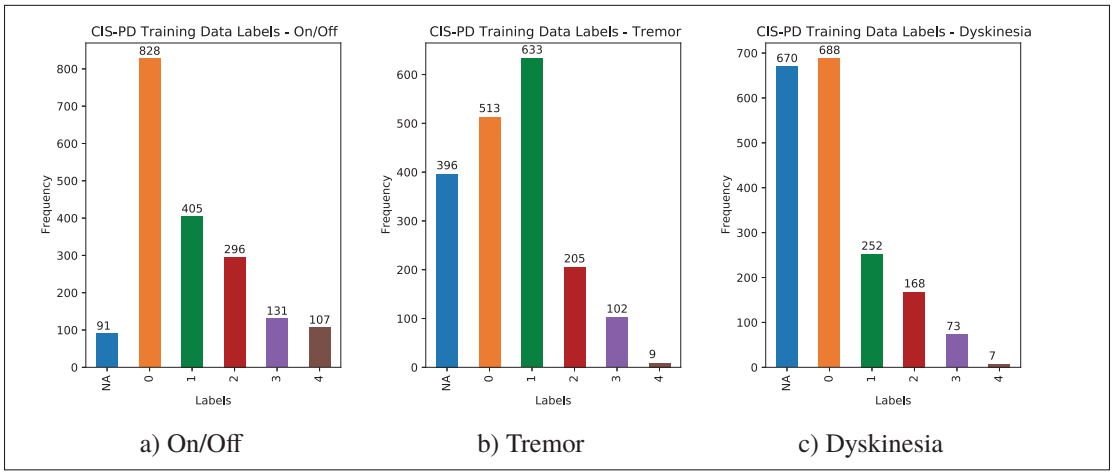
Figure 3.10 Architecture of approach I for Tsfresh + XGBoost Per Patient

We also used an early stop as a regularization method to avoid overfitting the model. When there was no improvement in the root mean square error metric in 100 rounds, the training stopped. We experimented using the same training data for early stop (denoted **train** in the results section) or a development dataset for the early stop (noted as **dev**). The development set was created by splitting the test set in two and doing two rounds of predictions to cover each split.

3.4.3 Data augmentation

The CIS-PD database contained imbalanced data, as some classes were more frequent than others (Figure 3.11). The problem with an imbalanced dataset is that the algorithm may focus on classifying examples according to the label with the most examples. Therefore, we performed four different kinds of data augmentation to augment the size of the training dataset.

The first method we tried, linear combination, was applied to the features. This is because we believed that there might be a linear relationship between the extracted features. For the three other techniques: Gaussian noise, resampling, and rotation, we applied the perturbation directly on the raw signals before extracting the features.



a) On/Off b) Tremor c) Dyskinesia

Figure 3.11 Frequency of observations of each label in the CIS-PD database

3.4.3.1 Linear Combination

The linear combination technique was initially called Mixup and proposed in (Zhang *et al.*, 2017b) to train neural networks. This simple data augmentation technique linearly combines two training examples to create a synthetic sample (equation 3.4).

$$\begin{aligned} \tilde{x} &= \lambda x_i + (1-\lambda)x_j \\ \tilde{y} &= \lambda y_i + (1-\lambda)y_j \end{aligned} \tag{3.4}$$

where:

x_i, x_j = raw input vectors

y_i, y_j = labels of x_i and x_j

λ = scalar between 0 and 1

$\tilde{x}$ = new augmented feature with label $\tilde{y}$

Contrary to Mixup, where examples were randomly combined, we randomly combined feature samples on the condition that they were from the same subject, as the label is very subjective on the patient.

While many data augmentation techniques aim to make slight changes to the data without changing the label, a linear combination is a non-label-preserving technique. We compute a new label $\tilde{y}$. It can also serve as regularization as we introduce some noise in the training data. As the labels are also changed, it can help the model reduce the memorization of corrupt labels. Therefore, this method was promising for this task. Each label is subjective to the subject as they were self-rating their symptoms. The annotation of labels is also susceptible to large intra-class variability.

3.4.3.2 Gaussian Noise (Jittering)

Another standard data augmentation in time series is to inject Gaussian noise in the signal without changing the label. We consider two parameters to generate the noise. The mean (μ) represents the center of the distribution. Second, the standard deviation (σ) characterizes the spread of the points from the mean.

3.4.3.3 Resampling

We resampled the signals while changing as little as possible their speed. As we are working with data collected from wearable sensors, changing the signal's speed too much could affect the label. We experimented with downsampling and upsampling the signal. When downsampling, we applied a low pass filter to smooth the signal and remove high frequencies. The parameters used are an order of 10 and a sample rate of 50 Hz. We determined the cutoff value with the

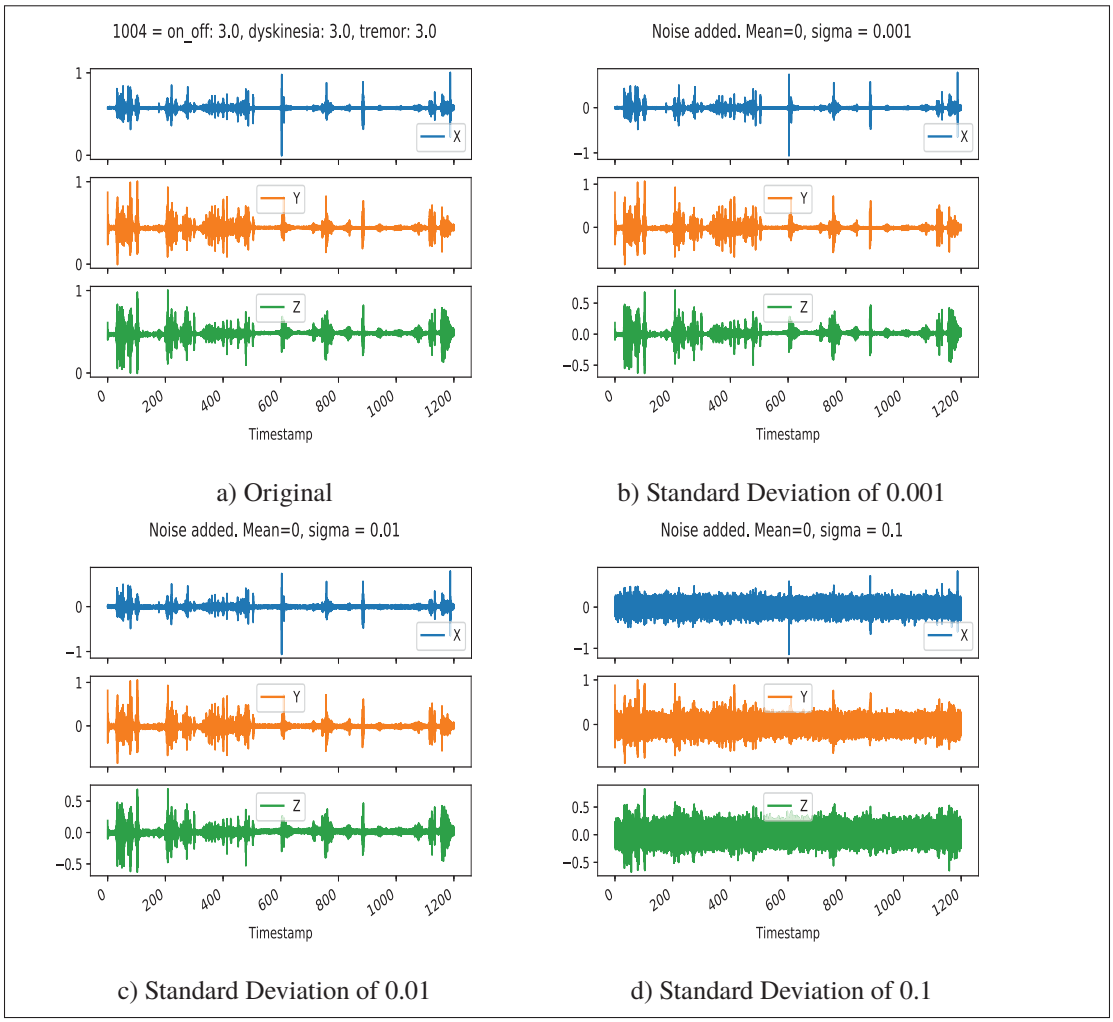
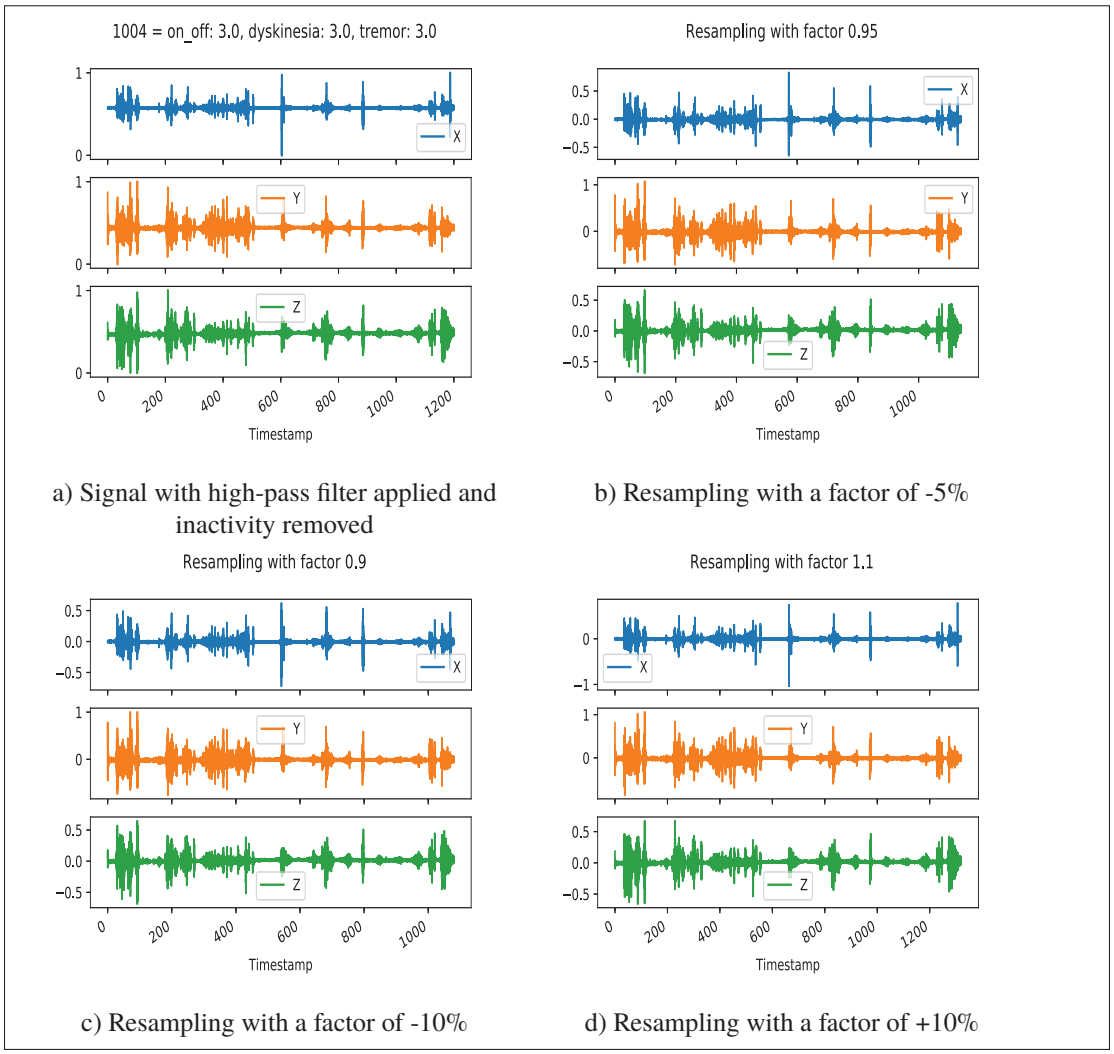


Figure 3.12 Visualization of accelerometers when noise is injected

Nyquist sampling theorem. It states that the sampling rate should be at least twice the maximum frequency component of the signal. The initial sample rate was 50 Hz, which we downsampled by a factor : $cutoff = \frac{50}{2} \times factor$. With this cutoff value, it attenuates frequency that is higher than half the sampling rate.

3.4.3.4 Rotation

Finally, we tried to use rotation as a data augmentation technique. The intuition behind this is that if a subject is experiencing tremor, the tremor's signal would be the same even in another



a) Signal with high-pass filter applied and inactivity removed

b) Resampling with a factor of -5%

c) Resampling with a factor of -10%

d) Resampling with a factor of +10%

Figure 3.13 Visualization of accelerometers when resampling the signals with different factors

direction. Considering this, we can increase the number of examples in our training dataset while keeping the same labels. It is also beneficial as the watch can be worn in different ways on the wrist, from one patient to another, but also from one day to the next.

To apply the rotation on the signals, we used Euler's rotation theorem, which says that in a three-dimensional space, any displacement of a rigid body can be explained with a single rotation matrix (equation 3.8). This single rotation matrix is obtained by doing three successive rotations along the axes z, y, and x in that precise order (equations 3.5, 3.6, and 3.7). Therefore, we chose

a random angle between $[-45°, 45°]$ to rotate the signal and used that angle for all the axes : $\phi = \theta = \psi =$ randomly generated angle, where ϕ, θ and ψ are angles of the rotations around the axes.

A rotation of ψ radians around the x-axis is defined as (Slabaugh, 1999):

$$R_x(\psi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\psi & -sin\psi \\ 0 & sin\psi & cos\psi \end{bmatrix} \tag{3.5}$$

A rotation of θ radians around the y-axis is defined as:

$$R_y(\theta) = \begin{bmatrix} cos\theta & 0 & sin\theta \\ 0 & 1 & 0 \\ -sin\theta & 0 & cos\theta \end{bmatrix} \tag{3.6}$$

A rotation of ϕ radians around the z-axis is defined as:

$$R_z(\phi) = \begin{bmatrix} cos\phi & -sin\phi & 0 \\ sin\phi & cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.7}$$

The general rotation matrix R is obtained by doing a sequence of three rotations around each axis:

$$R = R_z(\phi)R_y(\theta)R_x(\psi) \tag{3.8}$$

Finally, the signal X is rotated using a dot product of the rotation matrix, which gives the rotated signal denoted as $X_{rotated}$:

$$X_{rotated} = X \cdot R \tag{3.9}$$

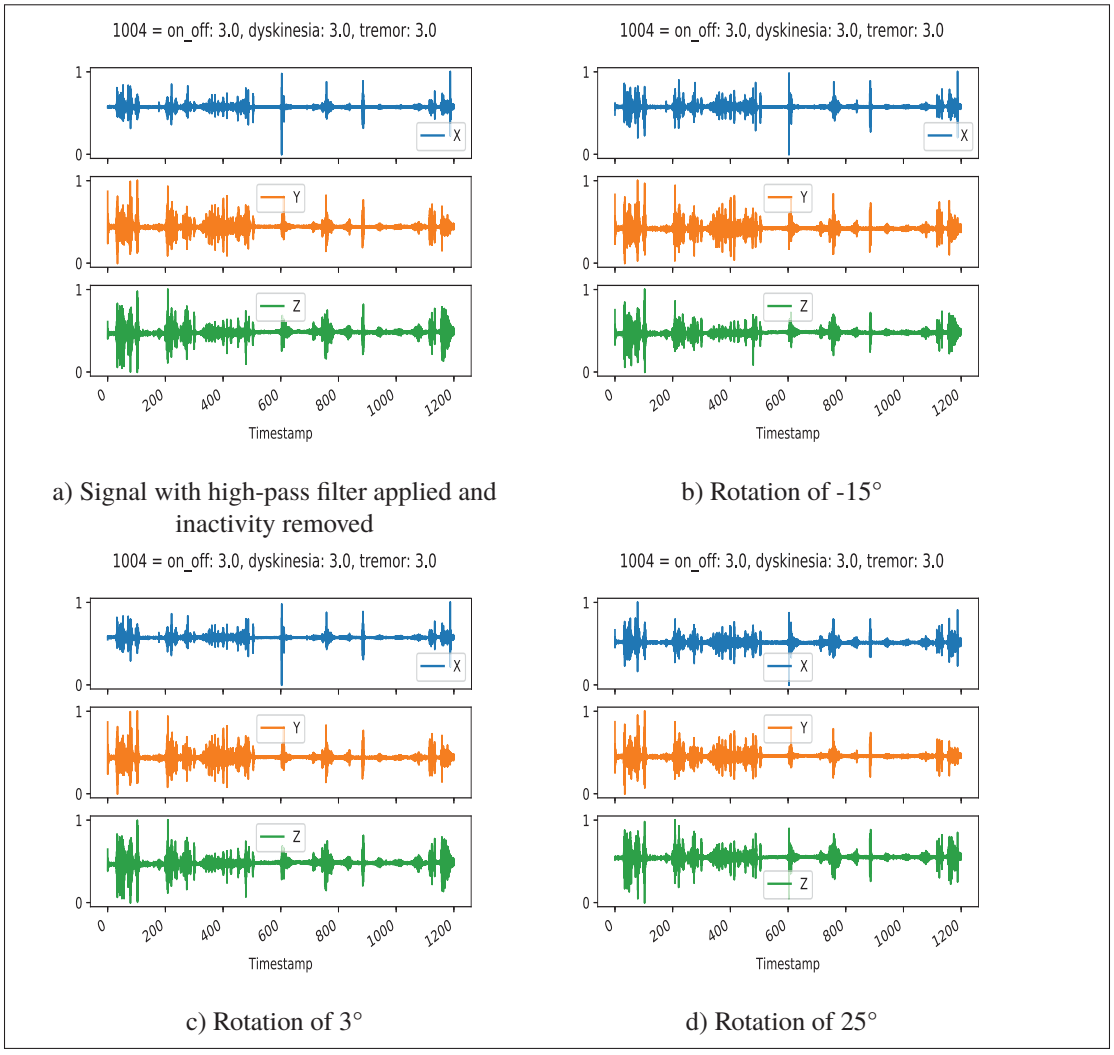


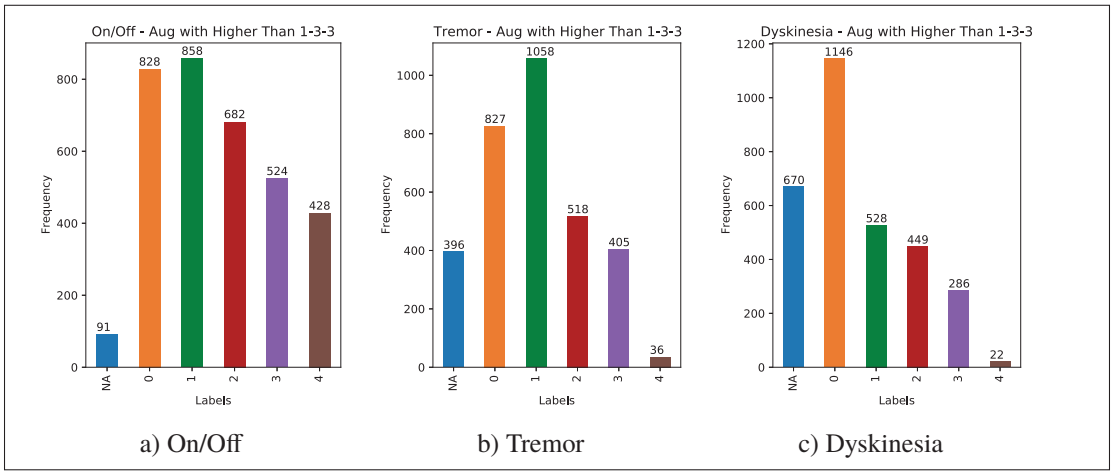
Figure 3.14 Visualization of accelerometers when rotating the complete signal with random angles

3.4.3.5 Oversampling the minority classes

The second part of the data augmentation experiment was only to augment the imbalanced labels. As Figure 3.11 shows, the CIS-PD dataset was severely imbalanced. Using rotation as data augmentation and the signal with the HPF applied and inactivity removed, we tried only to augment some labels to make the dataset more balanced. We chose to use rotation because it is easy to generate new random angles and diversify the data. It was easier to do for on/off, as there were hundreds of examples in each label's training dataset. However, for tremor and

dyskinesia, only 9 and 7 samples were available from the training data for the label 4. Even if we augmented those few examples, we are still far from a balanced dataset.

For the first experiment, we augmented all samples with a label higher or equal to 1. Examples with labels higher or equal to three were also augmented twice. Figure 3.15 shows the resulting distribution of labels. However, this way of augmenting the labels was intensive and changed the distribution.



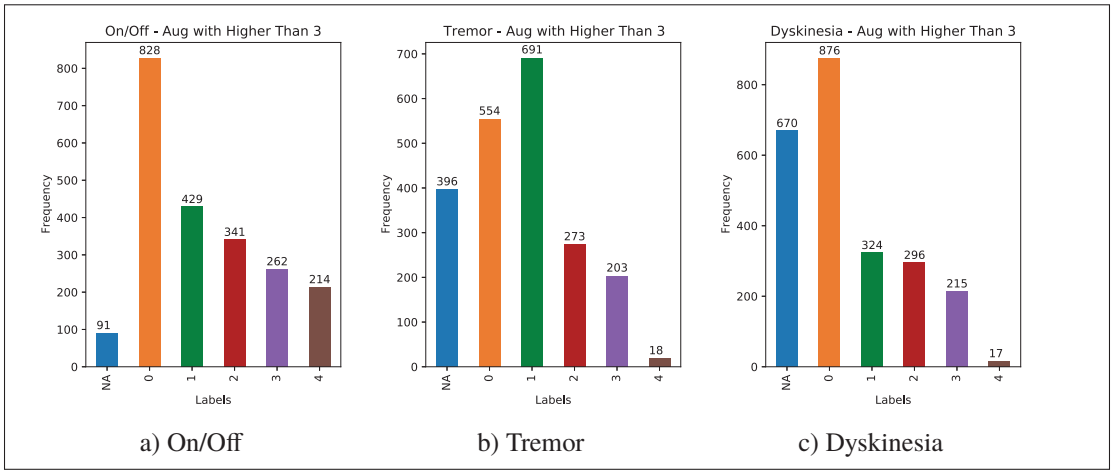
a) On/Off b) Tremor c) Dyskinesia

Figure 3.15 Distribution of the labels for the CIS-PD database when augmenting with samples having labels higher than 1 and 3 (twice)

We did another experiment that was less drastic and only augmented labels higher or equal to 3. The resulting distribution is shown in Figure 3.16.

3.5 Approach II - Embeddings

For the second approach, we experimented by using different methods to extract features and using different backends. Figure 3.17 shows the experiments that we did.



a) On/Off b) Tremor c) Dyskinesia

Figure 3.16 Distribution of the labels for the CIS-PD database when augmenting with samples having labels higher than 3

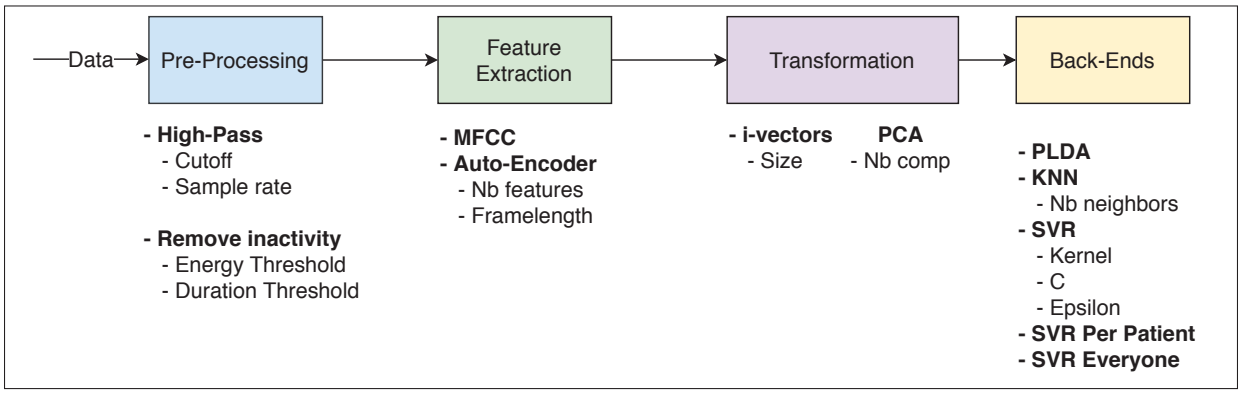


Figure 3.17 Diagram of the experiments for approach II and the hyperparameters to optimize

3.5.1 Feature extraction

This section describes the different methods used to extract features from the signals. We used two techniques for this approach : using MFCCs (section 3.5.1.1), or using an Auto-Encoder (AE) (section 3.5.1.2).

3.5.1.1 MFCC

We used MFCCs to extract features from time-series data. Traditionally used on the speech signal, the intuition behind this tentative is that MFCCs do time-series transformation: frequency analysis, filtering, power transformation. These are features that can extract information from any time series, even if it is not speech. Figure 3.18 shows the steps to extract MFCCs.

Figure 3.18 Steps to extract MFCCs from a signal

1. Windowing: the signal is divided into short frames typically around 20 to 40 ms;

2. Fast Fourier Transform: We calculate, for each frame, the power spectrum using the Fast Fourier Transform (FFT). It also transforms the signal from the time domain to the frequency domain;

3. Mel scale filterbank: We apply the 20-40 triangular filters to the power spectrum and sum each filter's energy. This is because the human hearing contains less information than what was extracted in the previous step. Therefore, we apply triangular filters and sum it up, which loses some information but allows us to know how much energy each band has. The number of filters applied will determine the number of coefficients (features) obtained at the end of the process;

4. Take the logarithm of all the previous filterbank energies: the original scale was linear, and the information in high frequency is more sparse. Changing the scale to the logarithm also makes the task to compute MFCCs easier by changing the operation from multiplication to addition;

5. Apply Discrete Cosine Transform (DCT) : It is a transformation similar to FFT. The mel filterbanks are multiplied by cosines of different frequencies. It helps to decorrelate the features.

3.5.1.2 Auto-Encoder

We used an AE to extract features from the signal. A neural network-based feature extraction method is proven to generate abstract features of high-dimensional data successfully. AE are unsupervised neural networks that are often used for dimensionality reduction. They compress the data while minimizing the reconstruction loss. The first part of the auto-encoder compresses the data at every layer until the bottleneck layer, where we can extract features in a lower dimension.

We did not have many training samples (1856), but the files were long enough to train the AE. To do so, we performed windowing. On average, the files had 60000 data points. We used frame lengths of 400 (8 seconds), with a step of 200 (4 seconds), so we ended up having $1856 \times (60000/400) = 278400$ data points, which is enough for training a deep neural network model.

In Figure 3.19, the AE uses a frame length of 400. Therefore, the input of the first fully-connected layer is of size 1200 (400 frame length $\times$ 3 axis = 1200). Then, there are three fully connected hidden layers of dimension 512. The layers' input size is constant as it is easier to implement, does not require additional tuning of input sizes, and lets the encoder learn what is important in the input data. Residual connections, inspired by He *et al.* (2016), are also implemented between each hidden layer. We add the output of a previous layer to the output of a deeper layer. The skip connections are used in neural networks to avoid facing the vanishing gradient problem when, during backpropagation, the gradient becomes so small that the network's weights are not significantly updated, which can cause the network to stop training. The skip connections, using the identity function, allow the gradient to be multiplied by one, maintaining a higher gradient value. Additionally, another benefit of skip connections is that they allow keeping information extracted in earlier layers so that deeper layers can use that information to learn further. For each residual block, the activation function is ReLU. The loss function of the AE is the MSE.

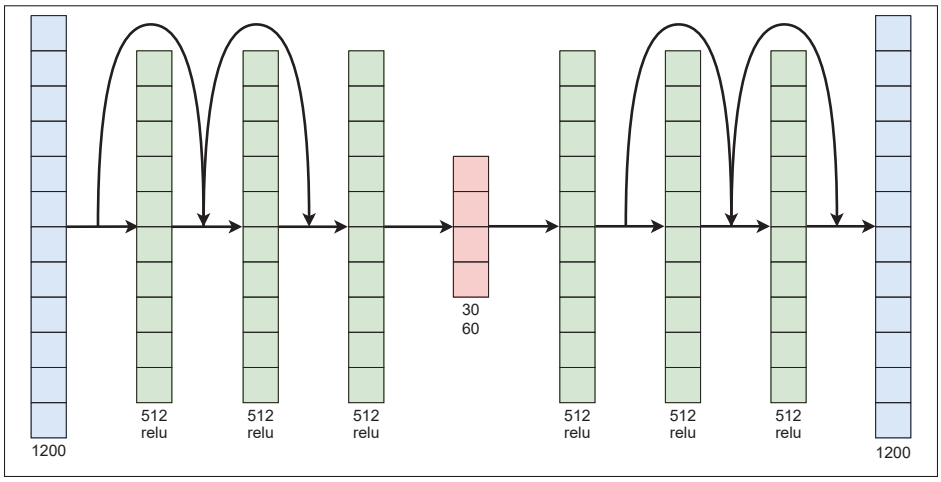


Figure 3.19 Architecture of the auto-encoder

3.5.2 Transformation

3.5.2.1 I-vectors

The goal of i-vectors (Dehak *et al.*, 2010) is to "automatically extract information transmitted in speech signal" (Dehak & Shum, 2011). We typically use them for speech recognition, language recognition, speaker recognition, and emotion recognition. However, we were interested in using i-vectors as we thought it might learn from all subjects' data while still adapting sufficiently to every subject's particular condition. Therefore, we extracted i-vectors from the MFCCs and AE features to convert the features into a fixed-size vector.

To extract i-vectors, we first need to train a GMM-UBM, like so:

1. **Feature Extraction:** The time-series are separated in windows before features like MFCCs are extracted,
2. **Train Universal Background Model (UBM):** A Gaussian Mixture Model (GMM) is trained over a large amount of data representing all the subjects,
3. **Adapt the UBM to the subject:** The target model is adapted from the UBM using data that is specific to that subject during Maximum A-Posteriori (MAP) training (see Figure 3.20),

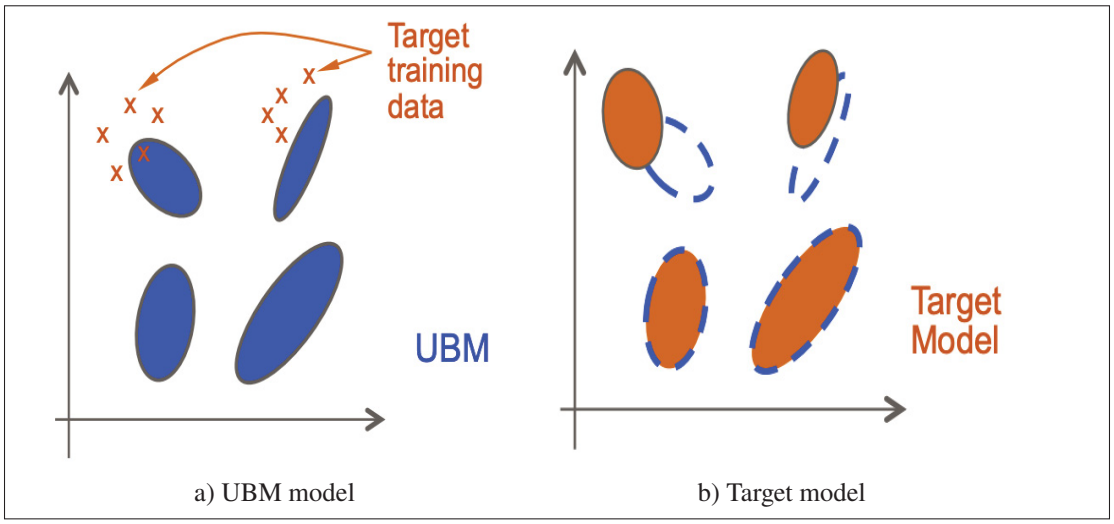


Figure 3.20 Representation of how the UBM model is adapted to the target model
Taken from Dehak & Shum (2011)

After training the GMM-UBM, we use the following equation 3.10 to extract i-vectors:

$$M = m + Tw \tag{3.10}$$

where:

M = the supervector mean of all gaussian in the mixture, obtained by training the UBM. It is a vector containing the mean of all subjects

m = the subject GMM mean supervector that is adapted to the target from the UBM using MAP adaptation

T = a low-rank total variability matrix, modeling the subject and channel variability

w = the i-vector, an intermediate representation that contains information about the subject and also the channel

In conclusion, we used i-vectors to leverage the concept of a UBM that uses all available data from subjects and then adapts a model to a specific subject. As the labels collected in the

database are very subjective to each subject as they annotate their symptoms themselves, it is necessary to adapt every model to the subject.

3.5.2.2 PCA

Principal Component Analysis (PCA) is a technique to reduce the dimensionality of the data while losing as little information as possible. This is useful for machine learning algorithms because of the curse of dimensionality. The more features a model uses, the more complex the model becomes. Furthermore, when too many features are used, the number of training examples needed to have a reliable statistical result grows exponentially.

PCA is an unsupervised approach as it does not use the labels. Going from d dimensions to k while ($d > k$) aims to maximize the variance of the data so that it is spread out in k-dimensions. First, PCA will find the center of the data and will shift it so that the mean becomes at the origin. The first component, called PC1, will then be determined by finding the projection that will maximize the variance by using the constraint that the eigenvector's norm must be equal to one, as eigenvectors indicate the variance.

The second component will be perpendicular to PC1 and will go through the origin. The third component will also be perpendicular to the previous components. Therefore, the hyperparameter of PCA is the number of components. To choose how many components are needed, we usually take the number of components that explain 90% of the variance. We tried different numbers of components, from 50 to 700, with increments of 50.

3.5.3 Backends

3.5.3.1 PLDA

PLDA is used as an i-vector scoring technique. It is a probabilistic approach to Linear Discriminant Analysis (LDA). LDA and PCA are both linear dimensionality reduction techniques. However, LDA is supervised, it uses labels, while PCA is unsupervised. LDA projects the

training data in a linear subspace and, when given a new example, will try to find the smaller distance between the classes and this new example. Therefore, if given a new class that it has not been trained on, it will be difficult to identify that class. This is where PLDA is useful. It is a probabilistic approach to LDA and will try to find the most optimal class.

Step 1: Training

The first step is to train the PLDA model, which will find the values for the parameters μ, F, G, Σ in equation 3.11 using Expectation-Maximization (EM) algorithm. In this work, there was one PLDA model for each subject. Therefore, the classes are the severity labels for the sub-challenges, from 0 to 4.

$$x_{ij} = \mu + \mathbf{Fh_i} + \mathbf{Gw_{ij}} + \Sigma_{ij} \tag{3.11}$$

where:

x_{ij} = j^{th} observation or i-vector of a subject for class i

μ = mean of all observations or i-vectors from all classes

$\mathbf{Fh_i}$ = depends on the identity of the i^{th} class. $\mathbf{F}$ contains the basis for between-class subspace, and $\mathbf{h_i}$ represents the position is that space.

$\mathbf{Gw_{ij}}$ = depends on the specific observation (i and j). $\mathbf{G}$ contains the basis for within-class subspace and $\mathbf{w_{ij}}$ is the position in that space.

Σ_{ij} = residual noise.

Step 2: Evaluation

After the training, the second part of the PLDA algorithm is to evaluate new i-vectors and predict their class. When we get a new i-vector from the testing data ($x_{i,j}$), the next step is to obtain the likelihood ratio, also called the PLDA scores with the mean i-vector x_{model} and the parameters from the training step ($\mu, \mathbf{F}, \mathbf{G}, \epsilon$). Using equation 3.12, the new i-vector $x_{i,j}$ is compared to all the classes, and the one with the highest log-likelihood (or above a threshold) will be the class assigned to the new i-vector.

$$h_0 = \text{ same class}$$

$$h_1 = \text{ different class}$$

$$Lk(x_{i,j}) = \log\left(\frac{P(x_{ij}, x_{model})}{P(x_{ij})P(x_{model})}\right) \tag{3.12}$$

where:

$Lk(x_{i,j})$ = log likelihood that the i-vector j is from class i
x_{ij} = the i-vector to evaluate
x_{model} = mean i-vector of all the i-vectors of class i in the training data

being

$$P\left(x'\right) = G_{x'}\left[\mu', AA^T + \Sigma'\right] \tag{3.13}$$

Where x' is either the i-vector to evaluate (x_{ij}), the i-vector model for a certain class i (x_{model}), or a matrice with both $\begin{bmatrix} x_{ij} \\ x_{\text{model}} \end{bmatrix}$. $G_a(mean, covariance)$ is a Gaussian in a. A is a matrix containing **F** and **G** from the training step of PLDA and Σ is the residual noise Σ from the training step.

3.5.3.2 KNN

KNN is a simple algorithm that will classify a new data point using the euclidean distance between the new example and the training data points. The new example will be classified according to the majority class of the k nearest neighbors. The hyperparameter of KNN is the number of neighbors, so how many examples to consider to find the class on a new example. In this work, we experimented with $k = 1, 2, 3$.

The Euclidean distance is computed with:

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2} \tag{3.14}$$

where:

p, q = two points
q_i, p_i = Euclidean vectors that start from the origin
n = n-space

3.5.3.3 SVR

SVM is a supervised machine learning algorithm used for classification problems. We can also use this model for regression tasks with a SVR. In this work, we used a SVR to predict the severity of the PD symptoms. The decision surface of a SVM is a hyperplane in an N-dimensional space. The N is equal to the number of features. Data points are used as support to the hyperplane, and the objective is to maximize the margin, which represents the distance between the separation of classes.

SVM can only separate data that is linear. In other words, data where the classes can be separated from a straight line. However, it is possible to transform non-linear data into a new dimension with a kernel trick, making the new projection linearly separable.

We will focus on the linear kernel as it is used in the experiments of this paper, for which the parameters to tune are C and epsilon (ϵ). The C parameter is the penalty factor that controls the amount of influence an error will have on the separation. If C is high, then points that are not linearly separable will be more penalized, leading to overfitting the training data. On the other hand, if C is too low, it can lead to underfitting. Therefore, C is a trade-off between maximizing the margin and classifying data points correctly. The epsilon defines a margin where no penalty is given to data points at an epsilon distance from the separation hyperplane.

The first **SVR** we experimented with was trained on files for a specific subject only, but all subjects shared the same hyperparameters value. The best hyperparameters were found from finding the configuration that provided the lowest final score for the test data.

For the **SVR Per Patient**, the predictions are made by training one SVR per subject. However, the SVR's hyperparameters are tuned for each subject individually, so they have different values for the C parameter and a different number of components when PCA is performed. To find which configuration is the best for each subject, we computed the weighted final score over each subject's five folds. Then, we chose the configuration that has the lowest weighted final score as the best configuration.

Initially, we did some experiments with different kernels and epsilon values. However, the best results were obtained with a linear kernel and an epsilon of 0.1, so we fixed these values to make the hyperparameters optimization faster. The different values we tried are reported in Table 3.5.

The intuition behind using an SVR tuned per patient is that the final score varies greatly depending on the subject. Also, the labels are very subjective to each patient as they were self-reported labels, which is why hyperparameters tuned only on subject-specific training files might obtain better performance. However, this technique also exposes the model to higher possibilities of overfitting.

The **SVR Everyone** is a model trained on all the data for all the subjects. No distinction is made per subject. We used one model for everyone, and the hyperparameters are tuned to optimize the R2 score. It is expected that this model will provide a higher MSE since the data is very subjective to the patient. For this configuration, all combinations of hyperparameters were evaluated. Then, to choose one unique best set of hyperparameters for each i-vector size, we computed the final score from equation 3.16, and the lowest final score was determined the best hyperparameters to be used.

Table 3.5 Grid search performed on the hyperparameters for approach II with the SVR

	Parameter	Values
Input	**Data Pre-processing**	Original training data, HPF, HPF + inactivity removed, original training data + inactivity removed
i-vector	**Dimension**	50, 100, 150, 200, 250, 300, 350, 400, 450, 500, 550, 600, 650, 700
Auto-encoder	**Nb features**	30, 60
	Frame length	240, 320, 400, 480
	Frame shift	200
SVR	**PCA** | **Nb Components**	350, 400, 450, 500, 550, 600, 650, 700
	Kernel	linear
	Epsilon	0.1
	C	2×10^{-13}, 2×10^{-11}, 2×10^{-9}, 2×10^{-7}, 2×10^{-5}, 2×10^{-3}, 2×10^{-1}, 2×10^{1}

3.6 Approach III - Fusion

A fusion of the predictions from Approach I and Approach II was done using a simple average of the predictions from both approaches. See Figure 3.21 for the architecture. Each approach was independent and could be tuned for everyone or per patient or not.

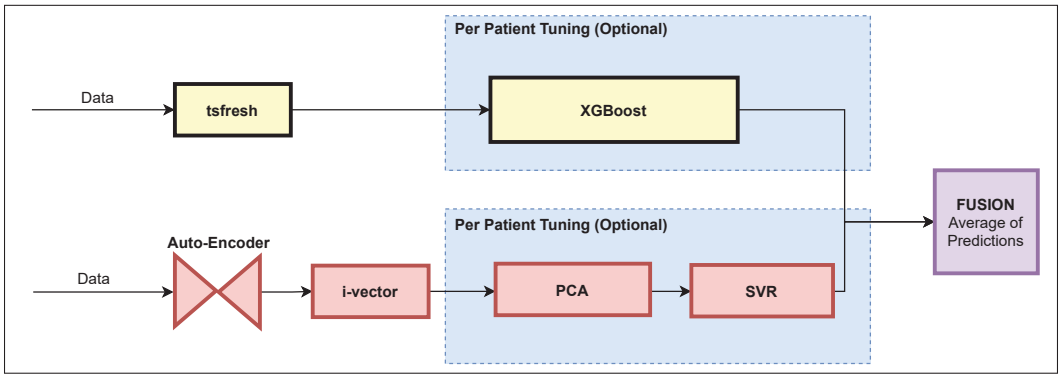


Figure 3.21 Architecture and hyperparameters of approach III

3.7 Evaluation Metric

As the number of files per patient varies greatly (Appendix II-1), the model's accuracy is evaluated with a weighted MSE, so that patients who have more files to train the model are more important. Table II-1 shows the number of files available per patient. The objective is not to create one model for all patients. Instead, it is to have one model per patient. It is expected that a model will not do as well as another if there were more training files available.

The MSE is first assessed for each subject k with equation 3.15:

$$\text{MSE}_k = \frac{1}{n_k} \sum_{i=1}^{n_k} (\hat{y}_{i,k} - y_{i,k})^2 \tag{3.15}$$

where:

n_k = number of test files for subject k
$y_{i,k}$ = i^{th} label for subject k
$\hat{y}_{i,k}$ = i^{th} prediction of the model for subject k

Then, the MSE_k of each subject are combined with equation 3.16:

$$\text{Final Score}_{sqrt} = \frac{\sum_{k=1}^{N} \sqrt{n_k}\, \text{MSE}_k}{\sum_{k=1}^{N} \sqrt{n_k}} \tag{3.16}$$

All the results reported in chapter 4 are from the final score calculated with equation 3.16.

3.8 Null Model

The baseline of this work is called the null model. It consists of the final score that would be obtained without using any accelerometer data. Instead, the null model predicts the average of the labels for each subject (Algorithm 3.1). We then use these predictions to compute the final score for the baseline.

Algorithm 3.1 How to get null model predictions for on/off

```
Input: Dataframe df_labels with columns for measurement_id,
  subject_id and on/off labels
Output: Predictions for the null model
for each subject do
    compute average(df_labels[subject])
end
for each file in df_labels do
    null_model_predictions += predict average of that subject
end
return null_model_predictions
```

3.9 Statistical significance tests

We evaluated our results to assess if the improvements obtained with the various techniques are statistically significant. We used two non-parametric tests: Kruskal-Wallis H and Wilcoxon's test. Our data distribution does not follow a normal distribution, and non-parametric methods do not assume any specific distribution.

The Kruskal-Wallis H test is used to determine if two or more independent samples have a different distribution. It needs at least five observations to be able to perform well. The null hypothesis (H_0) is that all data samples were drawn from the same distribution.

Wilcoxon's test compares only two samples that are related. It needs at least 20 observations to be fair. However, we have fewer observations than required, so that is to take into consideration. The null hypothesis (H_0) is also that the two samples are from the same distribution.

Alpha (α) is the level of significance at which we conclude that the result is due to something other than chance. Therefore the result is statistically significant. We used a typical alpha of 5%, which means that a result would be obtained by chance only 1 time out of 20.

The p-value (p) is the probability of observing the two data samples given the null hypothesis that the two samples were drawn from a population with the same distribution. If $p < \alpha$, then H_0 is rejected, and it means the two samples are not from the same distribution.

To perform the statistical tests, we used the MSE of each subject (the MSE_k from equation 3.15). Depending on the sub-challenge, the number of subjects available fluctuates. Even though there are 16 subjects in the CIS-PD database, some of them have missing data for certain sub-challenges. For on/off, 15 subjects are available. Tremor has 13 subjects and dyskinesia 11. Therefore, it is not optimal to perform statistical tests, as only a few MSE_k are available, but we thought it could provide helpful information.

3.10 Conclusion

In this chapter, we explored the databases used and how the data was collected from smartwatches to predict the medication status (on/off) and the severity of PD symptoms. Then, the four different data pre-processing options were explained.

The architecture of the first approach, which uses Tsfresh and XGBoost, was explained in section 3.4, along with the methodology used to perform data augmentation. Then, the architecture of approach II was covered in section 3.5. The second approach uses different embeddings. The third approach, which performs a fusion of both approaches, was explained in section 3.6. Finally, the evaluation metric, a weighted MSE that will be used to present the results, as explained in section 3.7.

CHAPTER 4

RESULTS

This chapter presents the results of the different approaches used to predict the severity of Parkinson's disease symptoms from accelerometers. First, we evaluate the pre-processing influence in a limited experimental set employing approaches I and II. In these cases, we used fixed hyperparameters for both approaches and only evaluated how results differ based on changing the pre-processing (section 4.1). Then, once we identified the optimal pre-processing conditions, we carry out a grid search fixing this pre-processing and varying the learning hyperparameters in Approach I (section 4.2) and Approach II 4.3. Finally, section 4.4 will cover the fusion of both approaches.

4.1 Data Pre-Processing

In the cross-validation experiments, we analyzed the use of original data, HPF to remove the gravity component and inactivity removal. The results are in the Table 4.1. The null model is the final score that would be achieved when predicting each subject's mean of the training labels (as explained in more details in section 3.8).

For the **approach I** with Tsfresh, using the HPF and inactivity removed data provided no improvement for on/off. For tremor, there was a slight improvement of 0.001. Dyskinesia had the most improvements, with a decrease of 0.004. Therefore, for the rest of this work, we focused on using either the original data or the HPF + inactivity removed data for this approach's experiments.

However, for the **approach II**, the best results were obtained on the HPF data for on/off and tremor. Dyskinesia instead got better results with original and inactivity removed data. The null model performed better than approach II for tremor, indicating that the problem we are trying to solve is difficult as the naive null model is better than our machine learning model. Again, we fixed the pre-processing for the rest of the project and used HPF for on/off and tremor, and

original and inactivity removed data for dyskinesia. The first approach provided better results for two of the three symptoms from the preliminary results, so we focused on improving approach I as it was promising.

In the tables of this chapter, we reported the p-value of Wilcoxon's test. When the p-value was less than 5%, we indicated the significance of the improvement with †. When the p-value was less than 1%, it was reported with ††.

Table 4.1 Final scores for the first two approaches when using different data pre-processing. A frame length of 400 and 30 features was used and i-vectors of dimensionality 500. Wilcoxon's test with respect to the null model: † P-value<0.05, †† P-value<0.01

Model	**Data Input**	**On/Off**	**Tremor**	**Dyskinesia**
Null model	NA	1.187	0.445	0.492
Approach I : Tsfresh + XGBoost Everyone	Original	**1.157**	0.442	0.490
	Original + Inactivity removed	1.182	0.449	0.487
	HPF	1.163	0.444	0.489
	HPF + Inactivity removed	1.178	**0.441**	**0.486**
Approach II : AE + SVR	Original	1.222	0.483††	0.501
	Original + Inactivity removed	1.226	0.495††	**0.495**
	HPF	**1.171**	**0.478**†	0.503
	HPF + Inactivity removed	1.211††	0.488††	0.503

4.2 Approach I - Time series features

The first approach uses Tsfresh to extracts features from the signals and an XGBoost for the regression task. We experimented with two configurations : **Tsfresh + XGBoost Everyone** which trains on the entire training set and shares the same hyperparameters for all subjects and is presented in section 4.2.1. The second configuration we tried, **Tsfresh + XGBoost Per Patient** trains one model per patient and the hyperparameters are tuned for each patient. This configuration is explained in section 4.2.2.

4.2.1 Tsfresh + XGBoost Everyone

Tsfresh + XGBoost Everyone had promising results in the preliminary data pre-processing experiments. Therefore, we decided to use data augmentation to generate artificial samples and increase the training data available. It could help improve the performance of the XGBoost model that might be affected by the imbalanced dataset.

4.2.1.1 Data augmentation

For data augmentation, we focused on two kinds of data input for the experiments: "original" and "HPF + inactivity removed" data. Those provided the best results for Approach I for the three sub-challenges when we analyzed the data pre-processing options in Table 4.1. We first tried to augment all the data with no regard to specific labels. However, we also tried only to augment the under-represented classes with oversampling to synthetically reduce the imbalance (section 4.2.1.1.7). However, this may change the data distribution and may lead to overfitting (Wen *et al.*, 2020).

Table 4.2 shows the preliminary results of the data augmentation techniques. We did not perform any grid search for the hyperparameters of each technique and empirically decided hyperparameters values. For on/off, no improvement was observed when using data augmentation. However, both tremor and dyskinesia were improved when resampling the signal by -10%. Tremor also profited from adding rotation to the signal. Moreover, using Gaussian noise improved the results for dyskinesia. For rotation, double data means that we generated the whole training dataset one time while applying a random rotation. In contrast, triple data means that we generated twice as many examples with rotation as the training dataset contained initially.

Considering that all the techniques contributed to improving results for at least one sub-challenge, we decided to go further and perform a grid search with different hyperparameters values for each technique. The coming sections will present the results.

Table 4.2 Final scores when using data augmentation techniques with approach I: Tsfresh + XGBoost Everyone. The results in bold are the ones that improved the final score in comparison with the result with no data augmentation. Wilcoxon's test with respect to the corresponding data pre-processing model: † P-value<0.05, †† P-value<0.01

Data augmentation technique	**Data Input**		**On/Off**	**Tremor**	**Dysk**
No Augmentation	**Original**		**1.157**	0.443	0.490
	HPF + Inactivity removed		1.178	0.441	0.485
Linear Combination λ = 0.2	**Original**		1.158	0.442	0.498
	HPF + Inactivity removed		1.171	**0.439**	0.489
Gaussian noise σ = 0.1	**Original**		1.175†	0.445	0.493
	HPF + Inactivity removed		1.169†	0.448	**0.484**
Resample (-10%)	**Original**		1.168	0.443	0.493
	HPF + Inactivity removed		1.178	**0.440**†	0.487
Resample (+10%)	**Original**		1.170	0.447	0.491
	HPF + Inactivity removed		1.179	0.445	0.485
Rotation [-45, 45]	**Original**	**Double data**	1.174	0.443	0.491
		Triple data	1.175	0.442	0.494
	HPF + Inactivity removed	**Double data**	1.163††	**0.438**	0.486
		Triple data	1.180	0.441	0.489

4.2.1.1.1 Linear Combination

The first technique we used is a linear combination of the features. The results are in table 4.3. For on/off, the data augmentation did not contribute to improving the final score, although $\lambda = 0.2$ was very close with 1.158. However, there were improvements for tremor with a $\lambda = 0.2$ using HPF + inactivity removed data. For dyskinesia, there was also no improvement with this technique. For the statistical significance test, we compared the augmented results with the corresponding data pre-processing when no augmentation was done. For instance, we compare original data with augmented original data, and do the same for HPF + Inactivity removed, to assess the improvement caused only by the augmentation technique.

Table 4.3 Final scores when using linear combination for data augmentation with approach I: Tsfresh + XGBoost Everyone. Wilcoxon's test with respect to the corresponding data pre-processing model: † P-value<0.05, †† P-value<0.01

Lambda (λ)	**Data Input**	**On/Off**	**Tremor**	**Dysk**
Null Model	**NA**	1.187	0.445	0.492
No Aug	**Original**	**1.157**	0.443	0.490
	HPF + Inactivity removed	1.178	0.441	**0.486**
0.001	**Original**	1.162	0.443	0.493
	HPF + Inactivity removed	1.173	0.447	0.490
0.2	**Original**	1.158	0.442	0.498
	HPF + Inactivity removed	1.171	**0.439**	0.489
0.4	**Original**	1.164††	0.445	0.491
	HPF + Inactivity removed	1.188††	0.446	0.488
0.5	**Original**	1.189††	0.450	0.505†
	HPF + Inactivity removed	1.183††	0.449	0.504

4.2.1.1.2 Gaussian Noise

The next data augmentation technique we tried was adding Gaussian noise to the signals. We explored different values for the hyperparameters of the added noise. The mean stayed the same with $\mu = 0$, but we increased the standard deviation. The results are in Table 4.4. For on/off, when adding noise, there were no improvements. The best result stayed using the original data input. For tremor, the HPF + inactivity removed data had the same final score as when adding a low amount of noise in the signal. However, for dyskinesia, the best results were observed using a high standard deviation of 0.1.

4.2.1.1.3 Resampling

We also experimented with different resampling factors in Table 4.5. This grid search was done using only HPF + inactivity removed since this is the pre-processing which had led to improvements in Table 4.2. On/off did not have any improvement, though -5% and +10% both have the same final score as when using no data augmentation at all (row HPF + Inact Removed).

Table 4.4 Final scores for adding different noise with approach I: Tsfresh + XGBoost Everyone. Wilcoxon's test with respect to the corresponding data pre-processing model: † P-value<0.05, †† P-value<0.01

Std. Dev. (σ)	**Data Input**	**On/Off**	**Tremor**	**Dyskinesia**
Null model	**NA**	1.187	0.445	0.492
No Aug	**Original**	**1.157**	0.443	0.490
	HPF + Inact Removed	1.178	**0.441**	0.486
0.001	**Original**	1.207††	0.449	0.496
	HPF + Inact Removed	1.184†	**0.441**	0.490
0.01	**Original**	1.197††	0.448	0.494
	HPF + Inact Removed	1.177†	0.442	0.490
0.1	**Original**	1.175†	0.445	0.493
	HPF + Inact Removed	1.169 (0.004) †	0.448 (0.002)	**0.484** (0.001)

Tremor and dyskinesia both got better results when using +/- 10%. The final score of tremor also decreased when resampling the signal by -5%.

Table 4.5 Final scores when resampling the signals on HPF + Inactivity Removed data with approach I: Tsfresh + XGBoost Everyone. Wilcoxon's test with respect to the corresponding data pre-processing model: † P-value<0.05, †† P-value<0.01

Resampling Factor	**On/Off**	**Tremor**	**Dysk**
Null model	1.187	0.445	0.492
Original	**1.157**	0.443	0.490
HPF + Inact Removed	1.178	0.441	0.486
-5%	1.178	**0.440**†	0.487
+5%	1.184	0.443	0.487
-10%	1.178	**0.440**†	0.487
+10%	1.179	0.445	**0.485**
-15%	1.179	0.445	0.487
+15%	1.183	0.446†	0.486

4.2.1.1.4 Rotation

The next technique we used was to rotate the whole triaxial signals by a random angle. Initially, we used a wide range of angles of $[-45°, 45°]$. It led to improvements for all the sub-challenges. However, since all the signals were rotated with a different random angle, we thought a more precise range of angles might be more optimal. From the results in Table 4.6, we found that on/off and tremor obtained better results with the initial wide range of angles. However, for dyskinesia, it benefited from using smaller angles of [-30, -25] and [25, 30], and the final score reached 0.483. This experiment was only done on HPF + inactivity removed data since the results were better than on original data from the preliminary search on data augmentation in Table 4.2.

Table 4.6 Final scores when using a smaller range of angles to generate new data. The offset and inactivity were removed from the signal. Wilcoxon's test with respect to the corresponding data pre-processing model: † P-value<0.05, †† P-value<0.01

Angles (°)	**On/Off**	**Tremor**	**Dyskinesia**
Null model	1.187	0.445	0.492
Original	**1.157**	0.443	0.490
HPF + Inact Removed	1.178	0.441	0.486
[-5, 0] [0, 5]	1.182	0.441	0.484
[-10, -5] [5, 10]	1.177	0.444	0.485
[-15, -10] [10, 15]	1.171	0.442	0.484
[-20, -15] [15, 20]	1.164†	0.444	0.484
[-25, -20] [20, 25]	1.169	0.444	0.487
[-30, -25] [25, 30]	1.173	0.445	**0.483**
[-35, -30] [30, 35]	1.181	0.443	0.487
[-40, -35] [35,40]	1.182	0.446	0.486
[-45, -40] [40, 45]	1.171	0.447	0.490
[-45, 45]	1.163††	**0.438**	0.486

4.2.1.1.5 Random Forest Regressor

We also tried to replace the XGBoost with a RFR as their hyperparameters are easier to tune. Additionally, they are less affected than the XGBoost by imbalanced datasets. However, the final scores, presented in Table 4.7, were worst than with the XGBoost even if we performed a grid search using similar values as the XGBoost.

Table 4.7 Final scores when using an XGBoost or a RFR

Model	Data Input	On/Off	Tremor	Dysk
Tsfresh + XGBoost Everyone	**Original**	**1.157**	0.443	0.490
	HPF + Inactivity removed	1.178	**0.441**	**0.486**
Tsfresh + RFR Everyone	**Original**	1.175	0.450	0.491
	HPF + Inactivity removed	1.180	0.451	0.491

Furthermore, we did a preliminary experiment with data augmentation using the RFR to see if that would help improve performance. The results are in Table 4.8. For the data augmentation, the Gaussian noise added had a standard deviation of 0.1, and the rotation angles generated were between $[-45°, 45°]$. Considering how the XGBoost continuously provided better results, we moved on with this algorithm for the rest of this work.

Table 4.8 Final scores on using data augmentations techniques with the RFR

Data augmentation technique	Data Input		On/Off	Tremor	Dysk
No Augmentation	**Original**		1.175	0.450	0.491
	HPF + Inactivity removed		1.180	0.451	0.491
Gaussian noise σ = 0.1	**Original**		1.179	0.450	0.492
	HPF + Inactivity removed		**1.172**	0.455	0.490
Resample (-10%)	**Original**		1.177	0.451	0.494
	HPF + Inactivity removed		1.178	**0.448**	0.493
Resample (+10%)	**Original**		1.179	0.456	**0.489**
	HPF + Inactivity removed		1.186	0.456	0.492
Rotation [-45, 45]	**Original**	**Double data**	1.183	0.451	0.490
		Triple data	1.182	0.450	0.493
	HPF + Inactivity removed	**Double data**	1.179	**0.448**	0.493
		Triple data	1.182	**0.448**	0.495

4.2.1.1.6 Combining techniques

As more than one augmentation method contributed to improving the overall final score, we combined the augmentation techniques to see if that would provide further improvements. The first two rows of Table 4.9 show the final scores obtained without any data augmentation. As always, the HPF + Inactivity removed signals with no data augmentation were combined with the other techniques, doubling the amount of data available for training the models. Similarly, when rotation is check-marked twice, we tripled the amount of training data. The results for dyskinesia did not improve by combining methods. However, it did improve the score for tremor, which went from 0.441 to 0.438, making the combination of data resampling -10% and rotation the best configuration for tremor.

Table 4.9 Final scores when combining data augmentation techniques using HPF + Inactivity removed data as input with approach I: Tsfresh + XGBoost Everyone. Wilcoxon's test with respect to the corresponding data pre-processing model: † P-value<0.05, †† P-value<0.01

Data Input	G. Noise σ = 0.1	Resample (-10%)	Resample (+10%)	Rotation [-45, 45]	On/Off	Tremor	Dysk
Original					**1.157**	0.443	0.490
HPF + Inact Rem.					1.178	0.441	0.486
HPF + Inact Rem.	✓				1.165	0.450	**0.484**
HPF + Inact Rem.		✓		✓	1.172	**0.438**	0.491
HPF + Inact Rem.	✓	✓	✓	✓	1.176	0.439†	0.487
HPF + Inact Rem.	✓	✓	✓	✓✓	1.177	0.442	0.489

4.2.1.1.7 Oversampling minority classes

Finally, we decided to generate samples from the under-represented classes in the dataset artificially. The labels 2, 3, and 4 were significantly under-represented in the dataset, as shown in Figure 3.11. We used rotation to oversample the minority classes. We chose this technique as it is easy to generate more data as it only requires random angles. Rotation was also the best data augmentation for tremor and provided close results for dyskinesia.

The results are in Table 4.10. The first experiment was to augment all labels higher or equal to 1 and 3. The labels higher than 3 were augmented twice by generating another set of random angles. The results are much higher than the null model. Therefore, we thought that we might have augmented the data too much and changed the dataset's distribution (see Figure 3.15). Accordingly, we tried only to augment the labels higher than 3. The results, reported in the last row of Table 4.10, improved from the previous experiment. The distribution of the data was not changed as much, but also fewer samples of the under-represented classes were added to the training dataset (see Figure 3.16). However, all results from these efforts to have a more balanced dataset were worst than the null model. Therefore, we did not try other data augmentations methods like Gaussian noise to oversample the minority classes.

Table 4.10 Final scores when performing data augmentation of only certain labels in an effort to have a more balanced CIS-PD dataset. We are using approach I: Tsfresh + XGBoost Everyone

Data augmentation technique	**Augmented Labels**	**On/Off**	**Tremor**	**Dysk**
Null model	NA	**1.187**	**0.445**	**0.492**
Rotation	$\geq 1, \geq 3, \geq 3$	1.549	0.514	0.544
Rotation	≥ 3	1.241	0.470	0.508

4.2.1.1.8 Summary

Finally, Table 4.11 summarizes the best results we have gotten so far with this approach. For On/Off, none of the data augmentation techniques nor the pre-processing options led to improvement, and the best result is to use the original data directly and achieves a final score of 1.157. For tremor, combining resampling with a factor of -10% and rotation achieved the best final score of 0.438. For dyskinesia, adding Gaussian noise to the HPF + inactivity removed signals led to a final score of 0.484. Even if rotation provided 0.483 for dyskinesia, we consider 0.484 the best final score so far as we repeated Gaussian noise five times, and the improvements were stable.

Table 4.11 Summary of the best final scores obtained for Tsfresh + XGBoost Everyone. Wilcoxon's test with respect to the corresponding data pre-processing model: † P-value<0.05, †† P-value<0.01

Data Input	G. Noise (σ = 0.1)	Resample (-10%)	Rotation [-45,45]	On/Off	Tremor	Dysk
Null Model				1.187	0.445	0.492
Original				**1.157**	0.443	0.490
HPF + Inact Removed				1.178	0.441	0.486
HPF + Inact Removed		✓	✓	1.172	**0.438**	0.491
HPF + Inact Removed	✓			1.169†	0.448	**0.484**

4.2.1.2 Statistical significance tests

Even with all the techniques we tried, we always see minor improvements. Therefore, we evaluated if they are statistically significant, and throughout this chapter, we reported the results of Wilcoxon's tests in the tables. The results' improvements were not significant for the Kruskal-Wallis test, so we did not report them in the tables.

As the statistical tests did not lead us to conclude, we decided to replicate some data augmentation experiments five times to see if the improvements were constant. From Table 4.12, we can see that the results are relatively stable when the experiment is repeated, as the standard deviation is low.

We also repeated the injection of Gaussian noise for $\sigma = 0.1$ and the results are in Table 4.4. The results are very stable for tremor and dyskinesia, and varies a little bit more for on/off.

4.2.1.3 Feature importance analysis

The scikit-learn library provides XGBoost a built-in function to analyze which features might be more critical in the model.

We used it on the best performing models using data augmentation to analyze the essential features. The 15 most important features were selected to appear in the graphs. The information

Table 4.12 Mean final scores and standard deviation (in parenthesis) when repeating the rotation as data augmentation for approach I: Tsfresh + XGBoost Everyone

Angles	On/Off	Tremor	Dyskinesia
[-5, 0] [0, 5]	1.174 (0.007)	0.442 (0.001)	0.486 (0.002)
[-10, -5] [5, 10]	1.174 (0.005)	0.444 (0.001)	0.485 (0.001)
[-15, -10] [10, 15]	1.172 (0.003)	0.442 (0.002)	0.485 (0.002)
[-20, -15] [15, 20]	1.174 (0.006)	0.444 (0.003)	0.485 (0.002)
[-25, -20] [20, 25]	1.174 (0.004)	0.442 (0.001)	0.486 (0.003)
[-30, -25] [25, 30]	1.176 (0.004)	0.443 (0.002)	**0.485 (0.002)**
[-35, -30] [30, 35]	1.176 (0.007)	0.443 (0.001)	0.486 (0.001)
[-40, -35] [35,40]	1.174 (0.006)	0.445 (0.002)	0.486 (0.002)
[-45, -40] [40, 45]	1.175 (0.003)	0.446 (0.002)	0.489 (0.001)
[-45, 45]	**1.171 (0.006)**	**0.441 (0.002)**	0.486 (0.001)

gain is the feature importance score we used. It measures the relative contribution of the feature to the model. A higher value of importance means it is more important to generate predictions.

16 unique features were identified as most important across the 3 sub-challenges (Figure 4.1, 4.2 and 4.3), out of 43 total unique features available. The first letter corresponds to the axis in the plots, then the feature extracted from that axis. When the axis starts with X2, Y2, or Z2, it means that it uses the absolute features $\Delta X, \Delta Y, \Delta Z$ (inspired by Wang *et al.* (2017); Schaff (2017)). It is computed by doing the absolute difference between each sample of that axis (explained in section 3.4.1).

Dyskinesia (Figure 4.3) uses fewer features to do predictions. Only the moment, k-statistic, FFT, time-reversal asymmetry statistic, and the minimum.

4.2.2 Tsfresh + XGBoost Per Patient

Finally, we did one more experiment with the approach I and decided to train one XGBoost model per patient. Therefore, each model was trained only on a specific patient's data and tuned the hyperparameters accordingly. This configuration is called **Tsfresh + XGBoost Per Patient**.

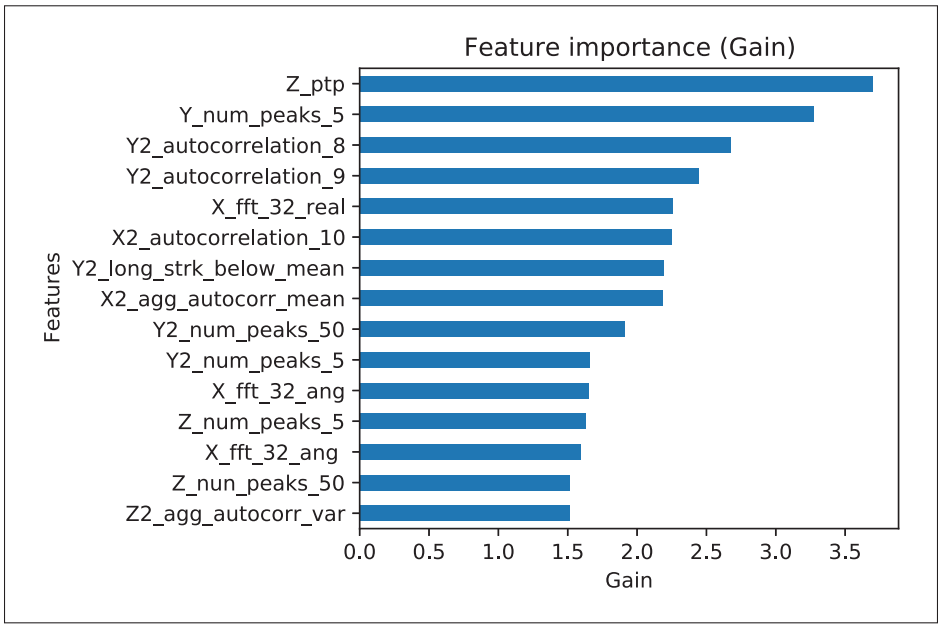


Figure 4.1 Important features for on/off, using original data

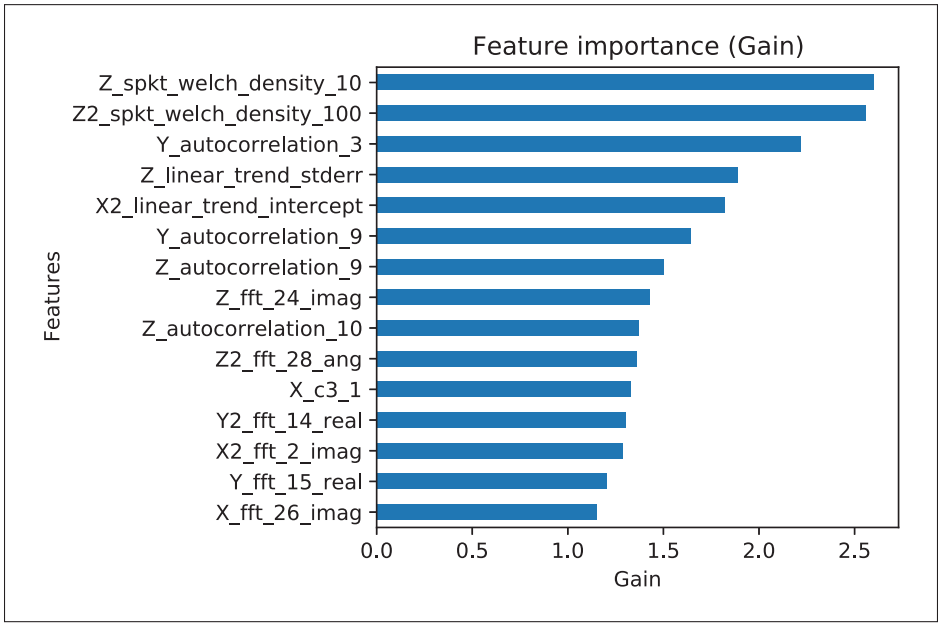


Figure 4.2 Important features for tremor, using HPF + inactivity removed data. We also used rotation and -10% resampling as data augmentation

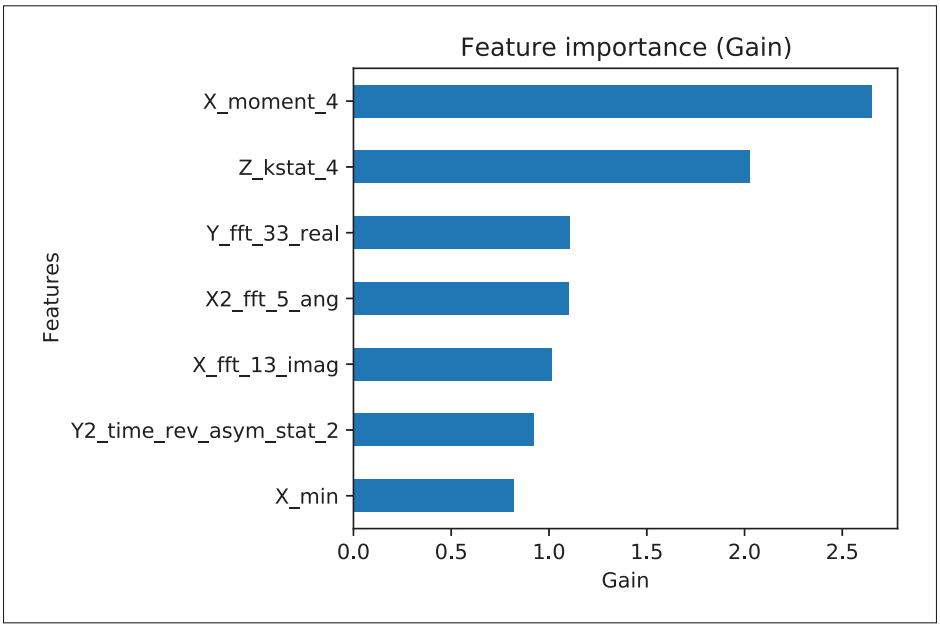


Figure 4.3 Important features for dyskinesia, using HPF + inactivity removed data. We also used Gaussian noise for data augmentation

We also experimented with using different datasets to perform an early stop of the training for this configuration. The results are in Table 4.13. For early stopping, **train** means that we used the whole training dataset to do both the training and early stopping. When noted **dev**, the test set was split in two to create a development set: the first half for early stopping and predicting on the second half. At the next iteration, the test set becomes the initial development set.

We used the original data for this experiment as we focused on improving the results for on/off and tremor, as dyskinesia was performing better with approach II following the data pre-processing experiments. For on/off and dyskinesia, both per patient tuning and using an early stop on the training dataset helped improve the results. However, tremor was also best when tuned per patient, but the early stopping was performed with a development set.

Table 4.13 Final scores for the two configurations of approach I

Model	Early Stop	On/Off	Tremor	Dysk
Tsfresh + XGBoost Per Patient	**Train**	**1.151**	0.454	**0.483**
	Dev	1.237	**0.437**	0.498
Tsfresh + XGBoost Everyone	**Train**	1.157	0.443	0.490
	Without early stop	1.165	0.443	0.491

4.3 Approach II - Embeddings

This section describes the second approach's results to solve the problem of predicting medication status (on/off) and the severity of PD symptoms: tremor and dyskinesia. For this approach, we experimented using different feature extraction methods and backends.

There were many parameters to tune and freeze. We first started by comparing the two feature extraction methods: using MFCCs or an AE followed by various models (PLDA, KNN, and SVR). Tables 4.14, 4.15, and 4.16 shows the results for the three sub-challenges. These preliminary results were obtained on Original + Inactivity removed data. The AE is performing better than the MFCCs, and as the dimensionality of the i-vector increases, the final scores are improving. Additionally, the SVR seems to be the best backend.

Table 4.14 Final scores for on/off with "original + inactivity removed" data. 10 MFCC are used. The auto-encoder had a frame length of 400 (8 seconds) and was extracting 30 features

Feature Extraction	Model	i-vector dimensionality					
		50	**100**	**150**	**200**	**250**	**300**
MFCCs	**PLDA**	2.805	2.617	2.507	2.570	2.604	2.600
	KNN	1.942	1.911	1.912	1.827	1.925	1.992
	SVR	1.357	1.332	1.319	1.301	1.298	**1.291**
Auto-Encoder	**PLDA**	2.305	2.052	2.022	1.999	2.117	2.117
	KNN	1.852	1.943	1.926	1.951	1.855	1.904
	SVR	1.343	1.307	1.287	1.275	1.271	**1.261**

Next, we focused on the AE and experimented with i-vectors of higher dimensionality to see if that would provide further improvements to the final scores. We chose to use HPF data from

Table 4.15 Final scores for tremor with "original + inactivity removed" data. 10 MFCC are used. The auto-encoder had a frame length of 400 (8 seconds) and was extracting 30 features

Feature Extraction	Model	i-vector dimensionality					
		50	**100**	**150**	**200**	**250**	**300**
MFCCs	**PLDA**	1.019	0.866	0.874	0.841	0.831	0.864
	KNN	0.776	0.769	0.763	0.778	0.750	0.781
	SVR	0.544	0.523	0.522	0.512	0.506	**0.500**
Auto-Encoder	**PLDA**	0.974	0.848	0.824	0.880	0.870	0.870
	KNN	0.835	0.878	0.786	0.872	0.822	0.845
	SVR	0.528	0.514	0.500	0.496	**0.490**	0.493

Table 4.16 Final scores for dyskinesia with "original + inactivity removed" data. 10 MFCC are used. The auto-encoder had a frame length of 400 (8 seconds) and was extracting 30 features

Feature Extraction	Model	i-vector dimensionality					
		50	**100**	**150**	**200**	**250**	**300**
MFCCs	**PLDA**	1.010	0.989	0.943	0.974	0.929	0.946
	KNN	0.824	0.809	0.826	0.780	0.803	0.835
	SVR	0.522	0.519	0.513	0.513	**0.511**	0.517
Auto-Encoder	**PLDA**	0.871	0.870	0.875	0.845	0.799	0.799
	KNN	0.979	0.891	0.962	0.891	0.893	0.871
	SVR	0.515	0.508	0.510	0.508	0.506	**0.503**

now on as it provided the best data pre-processing results for on/off and tremor (see Table 4.1), and continued to use original, and inactivity removed data for dyskinesia. Tables 4.17, 4.18, and 4.19 shows the results for the different symptoms. Therefore, we confirmed that the SVR was the best model to use with the AE. Different dimensionalities of i-vectors were optimal for each sub-challenge, ranging from 500 for on/off to 700 for dyskinesia.

Then, we experimented with different kinds of SVR : **SVR**, **SVR Per Patient**, and **SVR Everyone** and the results are in Tables 4.20, 4.21, and 4.22. The descriptions of the configuration used for the three SVRs was explained in the methodology chapter, section 3.5.3.3. We found that the best dimensionality for i-vectors is either 600 and 700, and that the best configuration was **SVR Per Patient**.

Table 4.17 Final scores for on/off when using an auto-encoder to extract features. The input was the HPF data. The auto-encoder was using a frame length of 400 and is extracting 30 features

Feature Extraction	**Model**	**i-vector dimensionality**							
		350	**400**	**450**	**500**	**550**	**600**	**650**	**700**
Auto-Encoder	**PLDA**	2.109	2.048	1.973	1.982	1.956	1.967	1.935	2.025
	KNN	1.941	1.903	1.919	1.893	1.929	1.925	1.897	1.866
	SVR	1.225	1.207	1.196	**1.171**	1.177	1.187	1.182	1.191

Table 4.18 Final scores for tremor when using an auto-encoder to extract features. The input was the HPF data. The auto-encoder was using a frame length of 400 and is extracting 30 features

Feature Extraction	**Model**	**i-vector dimensionality**							
		350	**400**	**450**	**500**	**550**	**600**	**650**	**700**
Auto-Encoder	**PLDA**	0.875	0.853	0.819	0.838	0.835	0.905	0.770	0.796
	KNN	0.895	0.878	0.852	0.868	0.833	0.865	0.818	0.860
	SVR	0.483	0.483	0.478	0.478	0.478	0.477	**0.474**	0.477

As the **SVR Per Patient** provided better results than the other SVRs, we moved on with this configuration. We assumed that the test dataset would also be close to the training dataset as it will only contain the same subjects (the model was not expected to generalize to unknown subjects). The distribution should be the same without too many outliers as PD progresses slowly.

We tested different hyperparameters to extract the features from the signal with the AE: using different frame lengths, number of features, and i-vectors dimensionalities. Again, we used HPF

Table 4.19 Final scores for dyskinesia when using an auto-encoder to extract features. The input was original + inactivity removed. The auto-encoder was using a frame length of 400 and is extracting 30 features

Feature Extraction	**Model**	**i-vector dimensionality**							
		350	**400**	**450**	**500**	**550**	**600**	**650**	**700**
Auto-Encoder	**PLDA**	0.881	0.826	0.868	0.893	0.910	0.905	0.884	0.853
	KNN	0.895	0.826	0.887	0.884	0.828	0.923	0.877	0.923
	SVR	0.504	0.502	0.502	0.495	0.498	0.498	0.498	**0.491**

Table 4.20 Final scores for on/off when using different configurations of SVRs. The input was the HPF data. The auto-encoder is using a frame length of 400 and is extracting 30 features

Model	i-vector dimensionality				
	500	**550**	**600**	**650**	**700**
SVR	1.171	1.177	1.187	1.182	1.191
SVR Per Patient	1.140	1.150	**1.137**	1.143	1.145
SVR Everyone	1.262	1.240	1.253	1.262	1.263

Table 4.21 Final scores for tremor when using different configurations of SVRs. The input was the HPF data. The auto-encoder is using a frame length of 400 and is extracting 30 features

Model	i-vector dimensionality				
	500	**550**	**600**	**650**	**700**
SVR	0.478	0.478	0.477	0.474	0.477
SVR Per Patient	0.457	0.460	**0.451**	0.459	**0.451**
SVR Everyone	0.509	0.500	0.492	0.484	0.497

data for on/off and tremor, and original with inactivity removed for dyskinesia as it provided the best results. The results are in Tables 4.23, 4.24, and 4.25.

For on/off, extracting 30 features with a frame length of 8 seconds obtained the best final score. For tremor, different configurations obtained the final score of 0.451. Considering this, we

Table 4.22 Final scores for dyskinesia when using different configurations of SVRs. The input was the HPF data. The auto-encoder is using a frame length of 400 and is extracting 30 features

Model	i-vector dimensionality				
	500	**550**	**600**	**650**	**700**
SVR	0.495	0.498	0.498	0.498	0.491
SVR Per Patient	0.481	0.481	0.478	0.480	**0.473**
SVR Everyone	0.507	0.520	0.526	0.532	0.537

decided to use the same configuration as on/off for this sub-challenge. For dyskinesia, a frame length of 8 seconds with 30 features was beneficial and decreased the final score to 0.473.

Table 4.23 Final scores for on/off when the AE extracts features from the HPF signal. We are using the SVR Per Patient for the regression

Nb Features	**Frame Length**	**i-vector dimensionality**				
		500	**550**	**600**	**650**	**700**
30	**240 (4.8 sec)**	1.183	1.160	1.178	1.173	1.153
30	**320 (6.4 sec)**	1.192	1.190	1.187	1.176	1.176
30	**400 (8 sec)**	1.140	1.150	**1.137**	1.143	1.145
60	**320 (6.4 sec)**	1.178	1.170	1.167	1.168	1.166
60	**400 (8 sec)**	1.216	1.190	1.171	1.176	1.173
60	**480 (9.6 sec)**	1.178	1.167	1.156	1.171	1.159

Table 4.24 Final scores for tremor when the AE extracts features from the HPF signal. We are using the SVR Per Patient for the regression

Nb Features	**Frame Length**	**i-vector dimensionality**				
		500	**550**	**600**	**650**	**700**
30	**240 (4.8 sec)**	0.467	0.461	0.467	0.467	0.465
30	**320 (6.4 sec)**	0.458	0.456	0.466	0.454	0.460
30	**400 (8 sec)**	0.457	0.460	**0.451**	0.459	**0.451**
60	**320 (6.4 sec)**	0.457	0.458	0.456	**0.451**	0.460
60	**480 (9.6 sec)**	0.456	0.459	0.459	0.457	**0.451**
60	**480 (9.6 sec)**	0.458	0.464	0.463	0.461	0.460

Table 4.25 Final scores for dyskinesia when the AE extracts features from the original + inactivity removed data for dyskinesia. We are using the SVR Per Patient for the regression

Nb Features	**Frame Length**	**i-vector dimensionality**				
		500	**550**	**600**	**650**	**700**
30	**240 (4.8 sec)**	0.490	0.490	0.490	0.485	0.491
30	**320 (6.4 sec)**	0.486	0.482	0.481	0.484	0.484
30	**400 (8 sec)**	0.481	0.481	0.478	0.480	**0.473**
60	**320 (6.4 sec)**	0.487	0.490	0.489	0.488	0.483
60	**400 (8 sec)**	0.487	0.487	0.482	0.484	0.481
60	**480 (9.6 sec)**	0.490	0.491	0.490	0.489	0.489

To summarize, the architecture in Figure 4.4 obtained the best results. First, the features were extracted from the signal using an AE. Then, we used a representational learning method called i-vector to convert the features into a fixed size vector, regardless of the signal's length. This way, we used a combination of trained AE and i-vector extractor to obtain a single (fixed-sized) vector per signal. We used PCA on the extracted i-vectors features. Then, a SVR with a linear kernel predicted the labels. The number of components for PCA and C's value for the SVR were tuned per patient. The details of the best hyperparameters found for each patient are available in Appendix V.

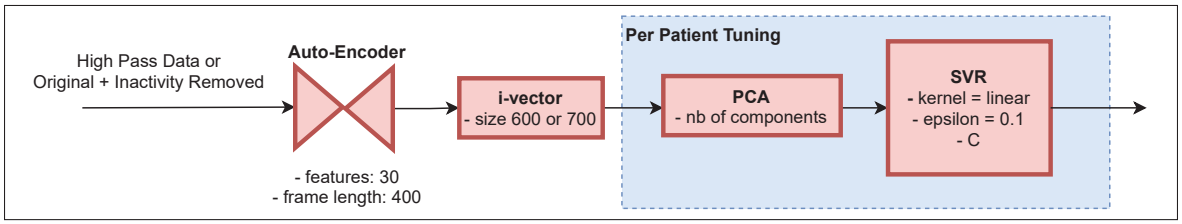


Figure 4.4 Architecture of approach II with the hyperparameters that provided the best results

4.3.1 Data augmentation

As tuning one by one the architecture's hyperparameters significantly improved the results from approach II, we performed preliminary data augmentation experiments (Table 4.26). The "no augmentation" row in the table shows the best results from approach II without data augmentation. On/off did not have any improvement from data augmentation. However, tremor had a small improvement when adding Gaussian noise and the final score decreased slightly from 0.451 with no augmentation to 0.450, but still not as good as the null model. However, the improvement is noticeable for dyskinesia, which goes from 0.473 to 0.462 with Gaussian noise. For the evaluation of whether the improvements are statistically significant or not, in Table 4.26, we compared the row with "no augmentation" with respect to the null model. For the rows with data augmentation, we compared them with the "no augmentation" results to see if the data augmentation techniques we used were indeed helpful.

Table 4.26 Final scores for data augmentation experiments with the approach II. Wilcoxon's test with respect to the corresponding data pre-processing model: † P-value<0.05, †† P-value<0.01

Data augmentation technique	Data Input		On/Off	Tremor	Dysk
Null Model	**NA**		1.187	**0.445**	0.492
No Aug	**Orig + Inactivity Removed**		1.159††	0.461	0.473
	HPF		**1.137**††	0.451	0.477†
Gaussian noise σ = 0.1	**Orig + Inactivity Removed**		1.170	0.456	0.469†
	HPF		1.182	0.450	**0.462**
Resample (-10%)	**Orig + Inactivity Removed**		1.179†	0.467	0.481
	HPF		1.158	0.453	0.469
Resample (+10%)	**Orig + Inactivity Removed**		1.172	0.453	0.475
	HPF		1.175	0.458	0.471
Rotation [-45, 45]	**Orig + Inactivity Removed**	**Double data**	1.154	0.451	0.480
	HPF	**Double data**	1.172	0.454	0.469

4.4 Approach III - Fusion

For the fusion, we perform the average of the predictions from the approach I and II. The following list is a summary of which approach and architecture provided the best final score that we aim to improve with the fusion:

- **On/Off:** 1.137 from Approach II,
- **Tremor:** 0.437 from Approach I - **Tsfresh + XGBoost Per Patient**, early stop on dev set,
- **Dyskinesia:** 0.462 from Approach II.

We used two different methods for the fusion of approach I. First, we used **Tsfresh + XGBoost Per Patient** with early stopping on the development set (section 4.4.1) and secondly, we used the best results from **Tsfresh + XGBoost Everyone** (section 4.4.2). In both scenarios, approach II uses the best configuration of the AE and i-vectors for all sub-challenges. The best configuration for each sub-challenge of Approach II is:

- **On/Off:** HPF, extracting 30 features with a frame length of 400, and i-vectors of size 600, SVR Per Patient,
- **Tremor:** HPF, data augmentation with Gaussian noise, extracting 30 features with a frame length of 400, and i-vectors size of 600, SVR Per Patient,
- **Dyskinesia:** Original + Inactivity removed data, extracting 30 features with a frame length of 400, and i-vectors of dimensionality 700, SVR Per Patient.

4.4.1 Fusion with Tsfresh + XGBoost Per Patient

We first used **Tsfresh + XGBoost Per Patient** with early stopping on the development set to perform fusion, and the results are in Table 4.27. We did not use early stopping on the training set because as the models are already tuned per patient, we wanted to avoid overfitting by using a development dataset instead of the training data for both training and evaluation. This fusion helped improve the final score for tremor and dyskinesia, but not for on/off, as the best final score to improve is 1.137 obtained with approach II.

Table 4.27 Final score when doing Fusion. For approach I, each sub-challenge used the predictions from per patient tuning with an early stop on the dev set

Description	On/Off	Tremor	Dyskinesia
Approach I : Tsfresh + XGBoost Per Patient with early stop on development set	1.237	0.437	0.498
Approach II: AE + SVR Per Patient	1.137	0.450	**0.462**
Approach III: Fusion with average	1.155	**0.429**	**0.462**

4.4.2 Fusion with Tsfresh + XGBoost Everyone

We tried one more method for the fusion fusion because on/off did not get the best results when using **Tsfresh + XGBoost Per Patient** like tremor and dyskinesia. We used **Tsfresh + XGBoost Everyone** with the configuration that was providing the best results for all sub-challenges. The results are presented in Table 4.28. This time, we improved the results for on/off as the final score decreased from 1.137 to 1.129.

As a reminder, here are the best configuration for each sub-challenge of Approach I using **Tsfresh + XGBoost Everyone**:

- **On/Off:** Original data,
- **Tremor:** HPF + Inactivity Removed, combination of basic features, resample -10% and rotation between $[-45°, 45°]$,
- **Dyskinesia:** HPF + Inactivity Removed, combination of basic features, and noise data augmentation with $\mu = 0$, and $\sigma = 0.1$.

Table 4.28 Final scores when doing the fusion with average. For approach I, each sub-challenge uses the configuration that provided the best results using the "everyone" configuration

Description	On/Off	Tremor	Dyskinesia
Approach I : Tsfresh + XGBoost Everyone	1.157	0.438	0.482
Approach II: AE + SVR Per Patient	1.137	0.450	**0.462**
Approach III: Fusion with average	**1.129**	0.438	0.466

The results presented in Table 4.27 and 4.28 are the best results we were able to achieve with the different experiments. Therefore, using fusion allowed us to get the best results for all sub-challenges.

4.5 Conclusion

In conclusion, throughout this chapter, we presented the various experiments we used to solve the research questions of this thesis:

- Can we predict the medication status (on/off) using accelerometers from a smartwatch during passive monitoring?
- Can we predict tremor severity using accelerometers from a smartwatch during passive monitoring?

- Can we predict dyskinesia severity using accelerometers from a smartwatch during passive monitoring?

We used four different data pre-processing options, and three of them proved to improve results on different sub-challenges and in the two approaches. The first approach used Tsfresh to extract features from signals followed by an XGBoost regressor. Different data augmentation techniques were used on the features (linear combination) and the signals (Gaussian noise, resampling, rotation).

The second approach, where an AE extracts features and is followed by a SVR, also showed promising results. Finally, approach III led to the best results for on/off, tremor and dyskinesia. For on/off, an average of the predictions between approach I using **Tsfresh + XGBoost Everyone** and the best configuration of approach II achieved a final score of 1.129. For tremor and dyskinesia, the lowest final score was obtained when using the approach I **Tsfresh + XGBoost Per Patient** with an early stop on the development dataset, where the final scores were 0.429 and 0.462. For all the symptoms, approach II was tuned per patient, and various data augmentation techniques helped improve the results.

CHAPTER 5

DISCUSSION

In this work, we presented three approaches to assess the severity of the PD symptoms: on/off, tremor and dyskinesia. One is based on extracting time series features, while the second approach uses embeddings to represents the accelerometers signals. Finally, a third approach aims to combine those two approaches to see if an ensemble method could provide better results. In this section, we will analyze those results.

5.1 Used data

One of the limitations of this research is the methodology used for the collection of the data. Although self-reported diaries are well accepted to label the severity of symptoms experienced by subjects, their use also introduces an intra-subject and inter-subject bias by an incorrect disease state recognition (Goetz *et al.*, 1997). This bias can further down the machine learning pipeline, affect the accuracy of the results. Goetz *et al.* (1997) reported that patient and clinician agreement was significantly higher after using a training tape to educate patients about motor fluctuations, even two months after watching the educational videos. Therefore, for future data collection, inclusion criteria could be that the participants are trained to assess their symptoms consistently with a couple of clinical visits. They could rate their symptoms under the supervision of clinicians who could validate their assessment. Considering the CIS-PD database contains this bias, there is only so much we can do to improve the predictions of the machine learning algorithms if the labels are not consistent.

The accelerometers in devices like the Apple Watch are not of medical grade, nor Food and Drug Administration (FDA) approved. Although they are of good quality and might give results as precise as medical-grade devices, their use can be questionable. Making health decisions based on devices that are not of medical grade would be a challenge.

Furthermore, it is essential to keep in mind the limitations of this research. We do not believe it is feasible to assess tremor in the lower limbs while using only a smartwatch on a wrist.

Another interesting question to consider is if the subjects changed their daily activities because of the sensors or removed the watches before or after filling the self-reported diaries as many files contained straight lines. That would alter the results of the research as patients would change their habits.

Also, one can raise the question if a scale of 0 to 4 is the best rating score to use. This scale follows the same scale widely used in the MDS-UPDRS, but other studies have used the Likert Scale, from 1 (none) to 7 (severe symptom) (Heijmans *et al.*, 2019). Another study that used smartphones and scripted activities to assess the severity of the disease, Zhan *et al.* (2018) developed an entirely new scoring method called the Mobile Parkinson Disease Score (mPDS). The mPDS "objectively weights features derived from each smartphone activity [...] and is scaled from 0 to 100 (where higher scores indicate greater severity)." (Zhan *et al.*, 2018) They used weak supervision to assign labels to the recordings. Therefore, this scoring method removes the need for subjects to label their symptoms, thus reducing bias. They showed that the mPDS correlates with current standards PD measures. Considering the MDS-UPDRS is still widely used and uses only a few options (1 to 4) for subjects to rate the severity of their symptoms, it still seems like the best option. However, future research could also explore further weak supervision techniques, which could greatly help in reducing the bias in labels collected from surveys.

5.2 Data Pre-Processing

For the first approach, the data pre-processing we applied worked well and offered improvements, except for on/off. It is difficult to assess why on/off might be having different results, especially as looking at the visualization of time series does not provide any insights. For example, a time series that has a label of 0, when compared with another recording labeled as very severe, the

signals look similar to human eyes. Using only a HPF without removing the inactivity provided better results for on/off and tremor for the second approach.

We did not experiment with the hyperparameters and thresholds of the pre-processing. We used an order of 10 for the HPF. In comparison, Van Hees *et al.* (2013) used an order of 4 to separate movements from gravity for human daily physical activities applications.

5.3 Approach I - Time series features

The following sections will analyze the results from the first approach, which uses time series features extracted with Tsfresh along with an XGBoost regressor to predict the severity of the PD symptoms.

5.3.1 Features

Tsfresh is a python package to extract many time-series features from signals. For on/off, we found that the range of values (ptp), the number of peaks of support 50 ($number\_peaks$), and the correlation between values that are 8 and 9 points apart ($autocorrelation$) were the three best features.

We also found that extracting power spectral analysis from the frequency domain was a useful feature for tremor ($spkt\_welch\_density$), and it was the two best features for the XGBoost. We used a power spectrum in the range of 1 to 10 Hz and trying with 100 Hz, and found that 10 and 100 were the most useful for our model, as opposed to Giuffrida *et al.* (2009) who found better results with a 3 to 7 Hz range.

For dyskinesia, the $moment$ and $kstat$ of 4 and the fast Fourier transform (fft) for the real, imaginary part (img) and the angle in degrees (angl) at different frequencies were most important.

5.3.2 Data augmentation

We experimented with various data augmentation techniques: linear combination, Gaussian noise, resampling, and rotation. All the techniques helped to achieve better results: they reduced the final score, a weighted MSE which is the average squared error (explained in section 3.7). Considering that the database has a scale of 0 to 4, the worst possible final score would be 16, and the best final score would be no error at all (0). For tremor, a combination of resampling with -10% and rotation using a wide range of angles $[-45°, 45°]$ is the lowest final score obtained in the first approach for **Tsfresh + XGBoost Everyone** with 0.438. For dyskinesia, injecting Gaussian noise also decreased the final score to 0.484. This means that the augmented data was useful for the XGBoost, and validates our hypothesis that rotating the accelerometers' axis still represents tremor, no matter in which axis it is detected. However, for on/off, data augmentation did not help to improve the final score, with only using the original data providing a final score of 1.157 for Approach I. Therefore, on/off has a higher squared error than the other symptoms, meaning it is more difficult to predict.

One of the limitations of this work is that we might be working with inconsistent and subjective labels due to how the data was collected. Using data augmentation and potentially altering the signals slightly without changing the labels, we introduced a small bias and noise that might help make the model more robust to changes in the labels and inconsistency.

Oversampling the under-represented classes did not improve the results for any symptom. It was challenging because there were very few labels were available in the minority classes. Undersampling would have been easier as it would have only required removing some samples, although it might lead to underfitting as we are removing examples. However, the majority classes are so outnumbered, removing a few samples is not likely to make any significant improvement.

5.4 Approach II - Embeddings

For the second approach, we tried both MFCCs and an AE to extract features. We then compared those two techniques with three different backends: PLDA, as it is common when using i-vectors. KNN because it was a simple classification algorithm to experiment with. Finally, we tried a SVR to keep the algorithm simple still while doing a regression instead of classification with KNN.

The best feature extraction technique was using an AE. MFCCs were sometimes better, for example, when using a KNN. However, the best results were obtained with the SVR, which was slightly better using the AE. Although MFCCs are usually used in speech, they still were competitive results with the AE, which means that the extracted features represent the time series extracted from accelerometers. After experimenting with different frame lengths and the number of features to extract, we found a 400 (8 seconds) window, and extracting 30 features worked best. The overlap stayed constant at 200 (4 seconds) shifts.

KNN provided similar or worst performances than PLDA, as the model was probably too simple and could not learn the more complicated fluctuations of the symptoms. Finally, the SVR was the best of the three, and we decided to push further and try to optimize the SVR model going forward.

We experimented with three different configurations of SVRs. The first one, **SVR** was finding the best hyperparameters that would be shared across all subjects, while **SVR Per Patient** was optimizing each hyperparameter to the training data of every subject. Without surprise, this configuration provided the best results. It might be prone to overfitting, as it is learning exactly from the data it will be tested on with no noise. However, this work aimed to create a model for each subject. The goal was not to create a model that would generalize to subjects never seen before in the training phase. Therefore, it is acceptable to tune the parameters on more focused data. The recruited participants were also in different stages of the disease (Hoehn and Yahr stages 1 to 3). We also tried the opposite to use all the training data available to train one model

for all subjects. This model was called **SVR Everyone**. It did not perform as well as **SVR** or **SVR Per Patient**, but we expect it can generalize better to out-of-domain subjects.

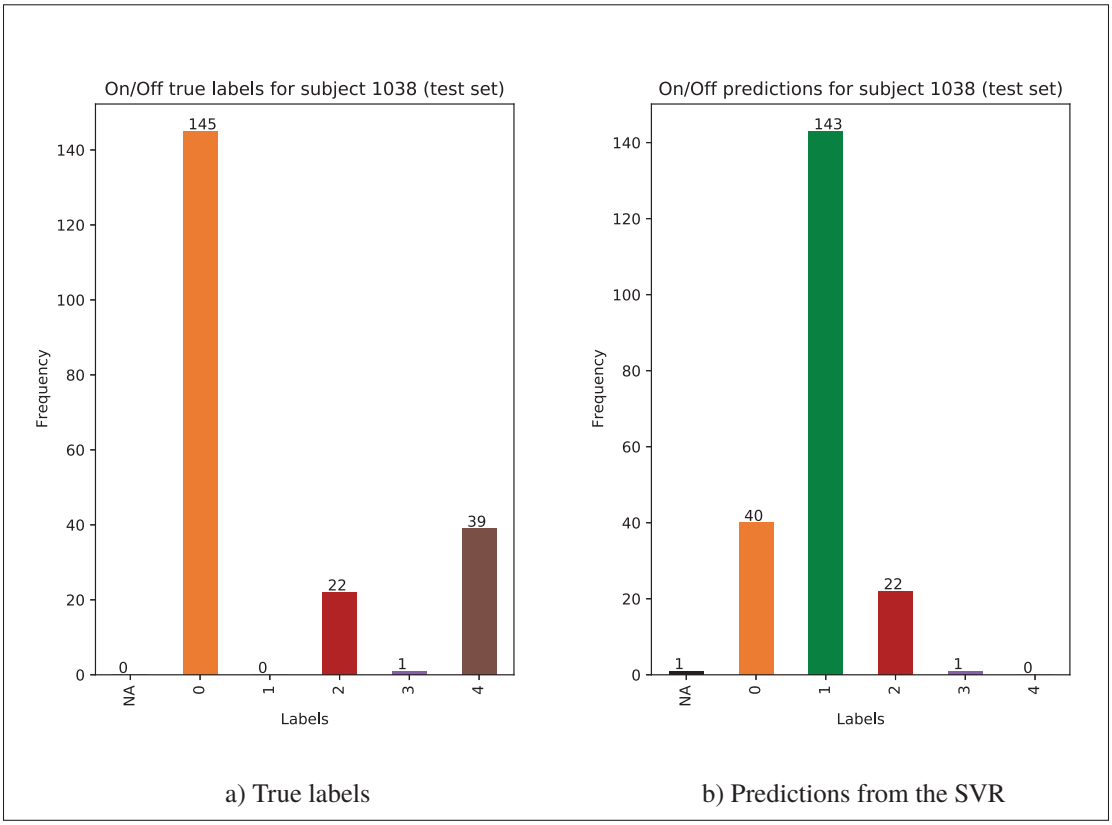
As the dataset is unbalanced, and the MSE per subject fluctuates greatly (see Appendix IV), we looked at a subject (1038) with a very high MSE and compared their true labels in comparison to the predictions of the SVR. The results are in Figure 5.1. In the true labels, we can see that we have many on/off files with a label of 0 (145). However, we have no labels at 1 and only 22 with a label of 2 and 39 with 4. When we look at the predictions in Figure 5.1b, we can see that the model predicts in majority labels of 1 for the test folds. Therefore, it raised the question of whether the model is only learning to predict the labels' mean. However, the SVR result is better than the null model. If we compare with another subject with a lower MSE, subject 1004 (Figure 5.2), we can see that this subject has a better distribution of the labels, and the average of the true labels is 1.40. The model predicts in majority labels of 2 and 1. Therefore, it might help in having a lower MSE for that particular subject.

5.4.1 Data augmentation

The preliminary data augmentation experiment for approach II helped to improve the final score for dyskinesia. Using Gaussian noise with a standard deviation of 0.1 led to a decrease of 0.09 compared with the final score without any augmentation. For tremor, the null model still is better than our machine learning approach. However, adding Gaussian noise slightly improved the final score compared to the score with no data augmentation. For on/off, augmenting the training data did not help at all.

5.5 Approach III - Fusion

The fusion of both approaches provided the best results obtained for all three sub-challenges. The simple average we used helped get the best of both approaches and improve the final score. The predictions of both models were very close to each other.



a) True labels b) Predictions from the SVR

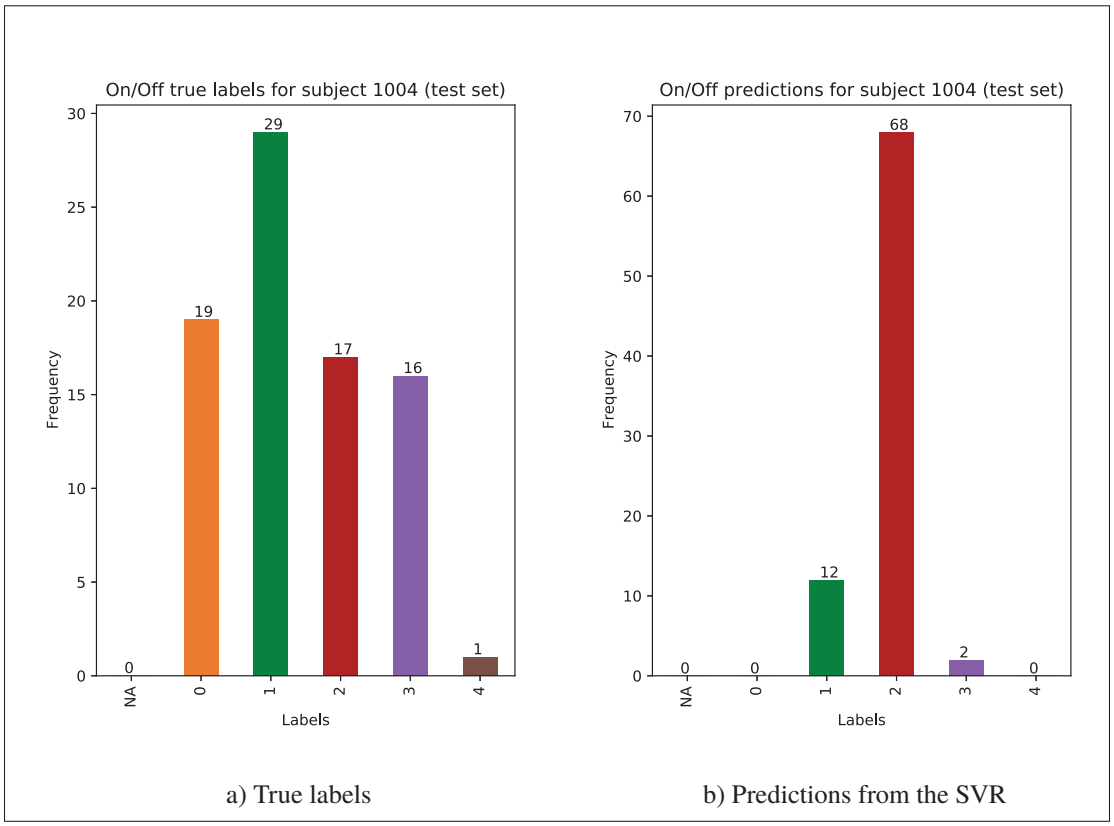
Figure 5.1 Comparing the true labels and the predictions obtained with the SVR for subject 1038

5.6 Analysis of the approaches of other teams in the challenge

The following sections will analyze the approaches of other teams who participated in the BEAT-PD challenge.

5.6.1 Sub-challenge 1: Predict On/Off medication status

The **dbmi** team (Huang *et al.*, 2020) did well for the on/off and dyskinesia sub-challenges. They also used Tsfresh to extract features but performed different data pre-processing steps. They performed mean resampling at a sample rate of 10 Hz, which reduced their dataset dimension. They also performed a linear interpolation to smooth along missing values. Additionally, they removed the gravity in the accelerometers with a value of 1, as for Apple, an acceleration of



a) True labels b) Predictions from the SVR

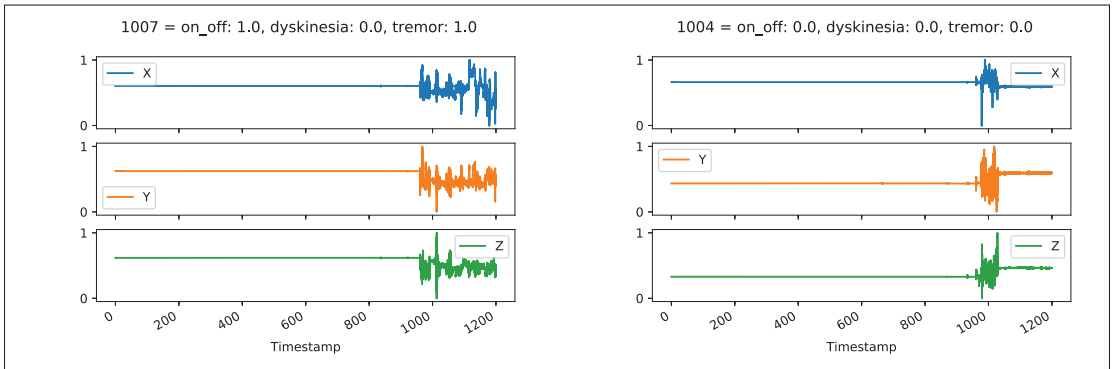
Figure 5.2 Comparing the true labels and the predictions obtained with the SVR for subject 1004

9.8 meters per second is represented with a value of 1.0. This is done because accelerometers recordings contain both the contribution of gravity and the linear acceleration (the movement), which are very difficult to separate (Van Hees *et al.*, 2013). It is one of the drawbacks when using accelerometry (Suzuki *et al.*, 2017).

They used a RFR, and found that the hyperparameters used significantly influenced their model's performance. This might explain why we did not get results as good with the RFR as we did with the XGBoost. We spent more time optimizing the XGBoost instead, even if the latter model is harder to tune. dbmi divided the time series into a window of 10 seconds, used 5 seconds of overlapping, and then extracted features from Tsfresh. This is different from what we did. We did not perform windowing before extracting features with Tsfresh.

Another team, **HaProzdor** (Matzner *et al.*, 2020), also used a RFR as well as 10 seconds windows with 5 seconds of overlap for on/off. Their approach is similar to our second approach. Interestingly, they extracted features similar to MFCCs (temporal and spectral features), and used PCA followed with an AE. However, instead of using a SVR, they used a RFR. They constructed 32 models with different hyperparameters values and different data given as input and performed an average of all the models' predictions to give a final prediction. There are many similarities between their approach and ours. The main difference might be that we are not performing windowing for the i-vectors extraction. They also did random undersampling, while we only experimented with oversampling.

They also experimented using the full session of the 20 minutes recording (the subject rated their symptom in the middle of this window) or using only the first half session before the subject's scoring. However, without data pre-processing to remove the inactivity present in many accelerometers like in Figure 5.3, using only the first half of a recording might not provide satisfactory results as many contain inactivity.



a) Accelerometer data with a lot of inactivity at the beginning

b) Another example of a recording with a lot of inactivity

Figure 5.3 Original accelerometers

5.6.2 Sub-challenge 2: Predict tremor severity

Another team, **yuanfang.guan**, (Guan, 2020) used a completely different approach than the ones reviewed until now, and used a 1-dimensional CNN with spatial and time augmentation.

The 20-minute recordings were segmented into 512 time points. Then, ten epochs were trained for each subject.

It is the only team that used a deep neural network method successfully. We briefly experimented with a CNN, but the performance on this model was not even close to the results from our approach I and II, so we did not continue to work in that direction.

5.6.3 Sub-challenge 3: Predict dyskinesia severity

Another team successful in predicting dyskinesia, **ROC BEAT-PD** (Page *et al.*, 2020) used a single RFR model per sub-challenge for all subjects. They extracted 16 features based on the accelerometers and used 30 seconds windows, so their windows are much longer than what we used. Using different time windows to assess different symptoms might improve as they do not have the same duration. Therefore, maybe longer windows of 30 seconds are useful for dyskinesia and not others. For example, tremor happens at rest. Therefore, selecting a window of more than 10 seconds might lead to capturing a period of rest tremor, but also the next moment when the individual starts moving, suppressing the tremor, which can reduce tremor detection (Salarian *et al.*, 2007). The team also differentiate itself from the other teams because they used the clinical data in their model: they added the age, gender, and MDS-UPDRS scores as input to the models. As we are using a scale of 0-4 to predict the severity, it is directly correlated to the MDS-UPDRS scores. Therefore it can be beneficial for a model to leverage this additional information. They also filled the missing values by using the participant's mean when available, otherwise, they used the population's mean. This might have allowed for more data to be accessible in the sub-challenges and lead to improvements in the performance of the models. To get a final prediction from all the extracted windows, they performed a simple average.

Our approach with SVR Everyone decreased our model's performance significantly, so, interestingly, this team was able to do that while getting good results. Is it only because they added more information about the demographics of the subjects? It is possible that the model was able

to learn from the age, gender, and MDS-UPDRS score so much that it enabled to train a single model on all the data.

CONCLUSION AND RECOMMENDATIONS

This research aimed to assess the severity of three symptoms of PD: the on/off phenomenon, tremor, and dyskinesia using the accelerometers from smartwatches. Furthermore, the challenge in this task resides in the fact that the symptoms are evaluated in a home environment and passive monitoring. The subjects are not required to perform any specific tasks. The research questions were the following:

- Can we predict the medication status (on/off) using accelerometers from a smartwatch during passive monitoring?
- Can we predict tremor severity using accelerometers from a smartwatch during passive monitoring?
- Can we predict dyskinesia severity using accelerometers from a smartwatch during passive monitoring?

The final score (weighted MSE) for each sub-challenge was respectively 1.129, 0.429 and 0.462. Therefore, it is possible to a certain extent to predict the medication status and the severity of tremor and dyskinesia from smartwatches. Each symptom had its own approach that led to the lowest final score:

- **On/Off**: Merging approach I (**Tsfresh + XGBoost Everyone**) with approach II (**AE + SVR Per Patient**),
- **Tremor**: Merging approach I (**Tsfresh + XGBoost Per Patient**) with approach II (**AE + SVR Per Patient**),
- **Dyskinesia**: Approach II (**AE + SVR Per Patient**).

Our four data pre-processing methods were all helpful for one symptom in one of our approaches. Additionally, using Gaussian noise, resampling, rotation, and linear combination are all data

augmentation techniques that helped improve our approach I or II results. Therefore, these techniques can generate more data, which is often a challenge in medical applications.

There are a few limitations to this study. We will discuss them in the next section.

Limitations & suggestions to overcome them

Limiting the number of mobile sensors used to monitor subjects during their daily life is understandable to make sure they are comfortable. However, it also raises the limitation that we cannot evaluate lower limb symptoms. For example, wearing only one smartwatch on one wrist might only detect the tremor on that hand and not on the other. For more comprehensive monitoring of the disease, it could be interesting for subjects to wear one smartwatch on each wrist, along with small sensors on each ankle. It could be a good compromise between enough sensors while still being comfortable for subjects.

Another limitation is that this work does not generalize to new subjects it has not been previously trained on. In that case, domain mismatch could happen when a subject is left out, especially as subjects might accomplish different daily activities, do not live in the same environments, and rate their symptoms differently, considering each symptom's rating is subjective. However, educating the participants with educational videos explaining how to rate symptoms might improve the quality of the labels. Additionally, using data augmentation can also help mitigate the inter-subject inconsistency of the labels by introducing a small noise.

Furthermore, self-reported labels might be inconsistent across subjects. Therefore, it could be highly beneficial to provide training to participants before collecting data to ensure that subjects can rate the severity of their symptoms consistently in accordance with clinicians' observations.

Using smartwatches like an Apple Watch to collect data is convenient as it is affordable and accessible. However, it is not a medical-grade device yet. It might not become one in the short

term as it would require extensive research. It would be interesting to compare the recordings of a medical-grade accelerometer to the results collected by an Apple Watch.

Future research

Future research could explore the use of on/off labels to predict dyskinesia. As these two symptoms are highly correlated, having information about one could help make better predictions. For example, in this work, better results were made on dyskinesia, so these predictions could be added to the on/off sub-challenge features and could provide some improvements.

We could also experiment in using larger windows for the AE. We currently use 10 seconds windows with a small overlap, but other studies have shown promising results with 30 seconds window for on/off. We might find that different size of windows is better for specific symptoms. The objective is always to capture the movement without dilating it.

Finally, we did not use the clinical or any demographics data. The age, the gender, and some MDS-UPDRS score Part III were available and could be added to the features of our approaches and might help in improving the predictions, especially for a SVR or an XGBoost model trained on the data of all patients.

APPENDIX I

FOX WEARABLE APP SCREENSHOTS

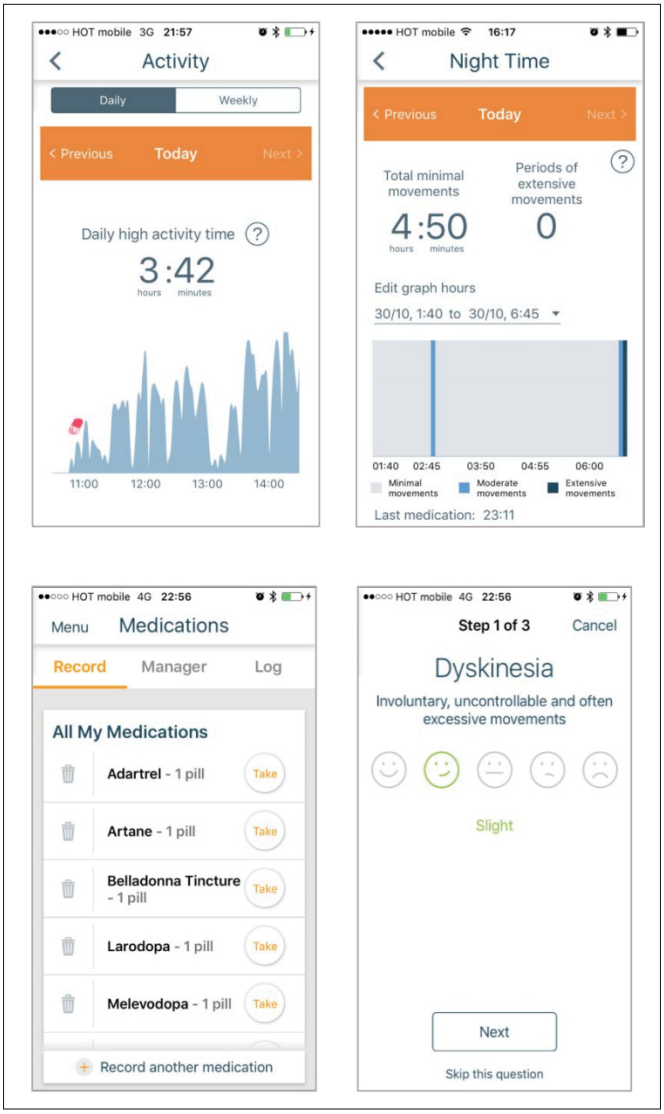


Figure-A I-1 Fox Wearable App Screenshots shared by Elm *et al.* (2019) and used for the diaries in CIS-PD. The content is a under Commons Attribution 4.0 International License[1]

[1] http://creativecommons.org/licenses/by/4.0/

APPENDIX II

NUMBER OF FILES FOR EACH SUBJECT IN THE DATABASE

Table-A II-1 Number of files for each subject_id (participant) in the CIS-PD database

Subject ID	Training Count	Testing Count	Ancillary Dataset Count
1004	82	27	
1006	37	12	
1007	299	99	
1019	45	15	
1020	195	65	
1023	106	35	
1032	177	59	
1034	40	14	
1038	207	70	
1039	130	43	
1043	34	12	
1044	72	24	
1046	67	22	
1048	91	30	
1049	82	27	
1051	194	64	
1000			20
1016			24
1018			139
1030			152
1041			17
Total	**1856**	**620**	**352**

APPENDIX III

APPROACH I: EXHAUSTIVE LIST OF FEATURES EXTRACTED

Table-A III-1 Features extracted using the tsfresh library for Approach I

Function	**Value of parameters**
abs_energy	NA
abs_sum_of_changes	NA
count_above_mean	NA
count_below_mean	NA
mean_abs_change	NA
mean_change	NA
variance_larger_than_standard_deviation	NA
range_count	[-∞, -4000[, [-3000, -2000[, [-2000, -1000[, [-1000,0[, [0, 1000[, [1000, 2000[, [3000, 4000[, [4000, ∞[
ratio_unique_values	NA
first_loc_min	NA
first_loc_max	NA
time_reversal_asymmetry_statistic	10, 100, 1000
autocorrelation	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 50, 100, 1000
c3	1, 2, 3, 4, 5, 10, 100
fft_coefficient_real	1 to 34
fft_coefficient_imag	1 to 34
fft_coefficient_angle	1 to 34
longest_strike_above_mean	NA
longest_strike_below_mean	NA
cid_ce	0, 1
binned_entropy	5, 10, 20, 50, 80, 100
number_crossing_m	0
number_peaks	1, 3, 5, 10, 50, 100, 500
spkt_welch_density	1, 2, 5, 8, 10, 50, 100
time_reversal_asymmetry_statistic	1, 2, 3, 4, 10, 100
symmetry_looking	0 to 19
large_standard_deviation	1 to 19
quantile	1 to 10
agg_autocorrelation	mean, median, variance
index_mass_quantile_	1 to 10
max_langevin_fixed_point	NA
linear_trend_	pvalue, rvalue, intercept, slope, stderr

Table-A III-2 Features extracted using the NumPy and scipy libraries for Approach I

Package	**Function**	**Value of parameters**
Numpy	Mean	NA
Numpy	Maximum	NA
Numpy	Minimum	NA
Numpy	Standard Deviation	NA
Numpy	Variance	NA
Numpy	Peak to Peak (ptp)	NA
Numpy	Percentile	10, 20, 30, 40, 50, 60, 70, 80, 90
Scipy	Skew	NA
Scipy	Kurtosis	NA
Scipy	k-Statistic	1, 2, 3, 4
Scipy	Moment	1, 2, 3, 4

APPENDIX IV

MSE PER SUBJECT

Table-A IV-1 MSE per subject for the on/off sub-challenge

Subject	Nb files	Approach I	Approach II	Null Model
1004	82	1.082	1.604	1.167
1006	37	0.285	0.281	0.263
1007	276	1.409	1.548	1.452
1019	45	2.089	2.098	2.117
1020	195	0.857	0.868	0.861
1023	106	1.197	1.229	1.191
1032	177	0.781	0.801	0.779
1034	40	1.057	1.186	1.227
1038	207	2.481	2.680	2.521
1039	130	1.111	1.051	1.131
1043	34	1.473	1.664	1.581
1044	72	0.264	0.240	0.243
1048	91	0.620	0.613	0.656
1049	82	0.811	0.885	0.858
1051	193	1.189	1.250	1.184

Table-A IV-2 MSE per subject for the tremor sub-challenge

Subject	Nb files	Approach I	Approach II	Null Model
1004	82	1.437	1.504	1.415
1006	37	0.501	0.569	0.495
1007	299	0.277	0.307	0.281
1019	45	0.433	0.461	0.442
1020	195	0.196	0.201	0.198
1023	106	0.343	0.352	0.340
1032	177	0.266	0.312	0.266
1034	40	0.298	0.350	0.299
1038	207	0.247	0.254	0.246
1043	34	1.270	1.280	1.312
1046	67	0.240	0.253	0.246
1048	91	0.414	0.501	0.343
1049	82	0.654	0.656	0.662

Table-A IV-3 MSE per subject for the dyskinesia sub-challenge

Subject	Nb files	Approach I	Approach II	Null Model
1004	82	1.162	1.202	1.194
1007	299	0.089	0.099	0.092
1019	45	0.765	0.691	0.754
1023	106	0.847	0.850	0.849
1034	40	0.815	0.870	0.828
1038	207	0.366	0.367	0.365
1039	130	0.403	0.377	0.406
1043	34	0.536	0.576	0.543
1044	72	0.102	0.116	0.099
1048	91	0.451	0.461	0.443
1049	82	0.421	0.413	0.420

APPENDIX V

APPROACH II: BEST HYPERPARAMETERS FOR SVR PER PATIENT

Table-A V-1 Best hyperparameters for each subject found for the best configuration of on/off using Approach II

Subject	**PCA Nb of components**	**SVR C**
1004	300	0.002
1006	50	20.0
1007	450	0.2
1019	200	20.0
1020	550	0.002
1023	350	0.2
1032	400	20.0
1034	250	0.2
1038	450	20.0
1039	350	0.2
1043	300	0.2
1044	600	0.002
1048	600	20.0
1049	350	0.002
1051	350	0.2

Table-A V-2 Best hyperparameters for each subject found for the best configuration of tremor using Approach II

Subject	**PCA Nb of components**	**SVR C**
1004	150	0.2
1006	150	0.2
1007	500	0.002
1019	250	2×10^{-5}
1020	50	0.2
1023	200	20.0
1032	450	20.0
1034	100	0.2
1038	300	0.2
1043	300	0.2
1046	100	0.2
1048	600	2×10^{-7}
1049	150	0.2

Table-A V-3 Best hyperparameters for each subject found for the best configuration of dyskinesia using Approach II

Subject	**PCA Nb of components**	**SVR C**
1004	250	0.002
1007	50	0.002
1019	150	20.0
1023	150	20.0
1034	650	0.002
1038	450	0.002
1039	650	2×10^{-5}
1043	250	0.2
1044	50	2×10^{-7}
1048	600	0.002
1049	50	0.002

BIBLIOGRAPHY

A. LeWitt, P. (2018, May, 4). Good Days and Bad Days with Parkinson's Disease [Online]. Consulted at https://www.worldpdcongress.org/home/2018/5/4/good-days-and-bad-days-with-parkinsons-disease.

Abrami, A., Heisig, S., Ramos, V., Thomas, K. C., Ho, B. K. & Caggiano, V. (2020). Using an unbiased symbolic movement representation to characterize Parkinson's disease states. *Scientific Reports*, 10(1), 1–12.

American Parkinson Disease Association. [American Parkinson Disease Association]. (2015, 07, 22). Good Days, Bad Days. Consulted at https://www.apdaparkinson.org/article/good-days-bad-days/#:~:text=One%20of%20the%20many%20issues,%E2%80%9D%20%E2%80%9CToday's%20a%20bad%20day.

American Parkinson Disease Association. [American Parkinson Disease Association]. (2020, 06, 29). Common symptoms of Parkinson's disease. Consulted at https://www.apdaparkinson.org/what-is-parkinsons/symptoms/#motor.

Ascherio, A. & Schwarzschild, M. A. (2016). The epidemiology of Parkinson's disease: risk factors and prevention. *The Lancet Neurology*, 15(12), 1257–1272.

Bacher, M., Scholz, E. & Diener, H. (1989). 24 hour continuous tremor quantification based on EMG recording. *Electroencephalography and clinical neurophysiology*, 72(2), 176–183.

Baldereschi, M., Di Carlo, A., Rocca, W. A., Vanni, P., Maggi, S., Perissinotto, E., Grigoletto, F., Amaducci, L., Inzitari, D. et al. (2000). Parkinson's disease and parkinsonism in a longitudinal study: two-fold higher incidence in men. *Neurology*, 55(9), 1358–1363.

BEAT-PD Challenge Webinar. (2020, 02, 04). BEAT-PD Challenge Webinar. Consulted at https://drive.google.com/file/d/1yLBu-EvQVgZpklThQ15coZ6oD6W361qp/view.

Bower, J. H., Maraganore, D. M., McDonnell, S. K. & Rocca, W. A. (1999). Incidence and distribution of parkinsonism in Olmsted County, Minnesota, 1976–1990. *Neurology*, 52(6), 1214–1214.

Chen, T. & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794.

Christ, M., Braun, N., Neuffer, J. & Kempa-Liehr, A. W. (2018). Time series feature extraction on basis of scalable hypothesis tests (tsfresh–a python package). *Neurocomputing*, 307, 72–77.

De Lau, L. M. & Breteler, M. M. (2006). Epidemiology of Parkinson's disease. *The Lancet Neurology*, 5(6), 525–535.

De Lima, A. L. S., Hahn, T., Evers, L. J., De Vries, N. M., Cohen, E., Afek, M., Bataille, L., Daeschler, M., Claes, K., Boroojerdi, B. et al. (2017). Feasibility of large-scale deployment of multiple wearable sensors in Parkinson's disease. *PLoS One*, 12(12).

Dehak, N. & Shum, S. (2011, 08, 27). Low-dimensional speech representation based on Factor Analysis and its applications. Consulted at https://people.csail.mit.edu/sshum/talks/ivector_tutorial_interspeech_27Aug2011.pdf.

Dehak, N., Kenny, P. J., Dehak, R., Dumouchel, P. & Ouellet, P. (2010). Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4), 788–798.

Del Din, S., Godfrey, A., Mazzà, C., Lord, S. & Rochester, L. (2016). Free-living monitoring of Parkinson's disease: Lessons from the field. *Movement Disorders*, 31(9), 1293–1313.

Deuschl, G., Schade-Brittinger, C., Krack, P., Volkmann, J., Schäfer, H., Bötzel, K., Daniels, C., Deutschländer, A., Dillmann, U., Eisner, W. et al. (2006). A randomized trial of deep-brain stimulation for Parkinson's disease. *New England Journal of Medicine*, 355(9), 896–908.

Elm, J. J., Daeschler, M., Bataille, L., Schneider, R., Amara, A., Espay, A. J., Afek, M., Admati, C., Teklehaimanot, A. & Simuni, T. (2019). Feasibility and utility of a clinician dashboard from wearable and mobile application Parkinson's disease data. *NPJ digital medicine*, 2(1), 1–6.

Espay, A. J., Bonato, P., Nahab, F. B., Maetzler, W., Dean, J. M., Klucken, J., Eskofier, B. M., Merola, A., Horak, F., Lang, A. E. et al. (2016). Technology in Parkinson's disease: challenges and opportunities. *Movement Disorders*, 31(9), 1272–1282.

European Parkinson's Disease Association. Wearing off and motor fluctuations [Online]. Consulted at https://www.epda.eu.com/about-parkinsons/symptoms/motor-symptoms/wearing-off-and-motor-fluctuations/.

Fahn, S. (1987). Unified Parkinson's disease rating scale. *Recent development in Parkinson's disease*.

Fahn, S. et al. (2003). Description of Parkinson's disease as a clinical syndrome. *ANNALS-NEW YORK ACADEMY OF SCIENCES*, 991, 1–14.

Fisher, J. M., Hammerla, N. Y., Ploetz, T., Andras, P., Rochester, L. & Walker, R. W. (2016a). Unsupervised home monitoring of Parkinson's disease motor symptoms using body-worn accelerometers. *Parkinsonism & related disorders*, 33, 44–50.

Fisher, J. M., Hammerla, N. Y., Rochester, L., Andras, P. & Walker, R. W. (2016b). Body-worn sensors in Parkinson's disease: Evaluating their acceptability to patients. *Telemedicine and e-Health*, 22(1), 63–69.

Foerster, F. & Smeja, M. (1999). Joint amplitude and frequency analysis of tremor activity. *Electromyography and clinical neurophysiology*, 39(1), 11.

Foschini, L., Alonso, D., Frasier, M., Keefe, J., Smolensky, L., Jayaraman, A., Shawen, N., Evers, L., Mariakakis, A., Omberg, L., Sieberts, S. & Snyder, P. (2020, January, 13). BEAT-PD DREAM Challenge [Synapse]. Consulted at https://www.synapse.org/#!Synapse:syn20825169.

Gallego, J. A., Rocon, E., Roa, J. O., Moreno, J. C. & Pons, J. L. (2010). Real-time estimation of pathological tremor parameters from gyroscope data. *Sensors*, 10(3), 2129–2149.

Giuffrida, J. P., Riley, D. E., Maddux, B. N. & Heldman, D. A. (2009). Clinically deployable Kinesia™ technology for automated tremor assessment. *Movement disorders: official journal of the Movement Disorder Society*, 24(5), 723–730.

Goetz, C. G., Stebbins, G. T., Blasucci, L. M. & Grobman, M. S. (1997). Efficacy of a patient-training videotape on motor fluctuations for on-off diaries in Parkinson's disease. *Movement disorders*, 12(6), 1039–1041.

Goetz, C. G., Tilley, B. C., Shaftman, S. R., Stebbins, G. T., Fahn, S., Martinez-Martin, P., Poewe, W., Sampaio, C., Stern, M. B., Dodel, R. et al. (2008). Movement Disorder Society-sponsored revision of the Unified Parkinson's Disease Rating Scale (MDS-UPDRS): scale presentation and clinimetric testing results. *Movement disorders: official journal of the Movement Disorder Society*, 23(15), 2129–2170.

Goetz, C. G., Stebbins, G. T., Wolff, D., DeLeeuw, W., Bronte-Stewart, H., Elble, R., Hallett, M., Nutt, J., Ramig, L., Sanger, T. et al. (2009). Testing objective measures of motor impairment in early Parkinson's disease: Feasibility study of an at-home testing device. *Movement Disorders*, 24(4), 551–556.

Guan, Y. (2020, May, 13). 2020 BEAT PD DREAM Challenge (Yuanfang Guan Solution 1st Place for tremor prediction) [Synapse]. Consulted at https://www.synapse.org/#!Synapse:syn21784439/wiki/603050.

He, K., Zhang, X., Ren, S. & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

Heijmans, M., Habets, J. G., Herff, C., Aarts, J., Stevens, A., Kuijf, M. L. & Kubben, P. L. (2019). Monitoring Parkinson's disease symptoms during daily life: a feasibility study. *npj Parkinson's Disease*, 5(1), 1–6.

Hoehn, M. M., Yahr, M. D. et al. (1998). Parkinsonism: onset, progression, and mortality. *Neurology*, 50(2), 318–318.

Hssayeni, M. D., Jimenez-Shahed, J., Burack, M. A. & Ghoraani, B. (2019). Wearable sensors for estimation of parkinsonian tremor severity during free body movements. *Sensors*, 19(19), 4215.

Huang, Y., Keller, M. & Saqib, M. (2020). Predicting Parkinson's Symptoms using Time Series Features from Sensor Data. Consulted at https://www.synapse.org/#!Synapse:syn21902357/wiki/603079.

Joundi, R. A., Brittain, J.-S., Jenkinson, N., Green, A. L. & Aziz, T. (2011). Rapid tremor frequency assessment with the iPhone accelerometer. *Parkinsonism & related disorders*, 17(4), 288–290.

Keijsers, N. L., Horstink, M. W. & Gielen, S. C. (2003). Automatic assessment of levodopa-induced dyskinesias in daily life by neural networks. *Movement disorders: official journal of the Movement Disorder Society*, 18(1), 70–80.

Keijsers, N. L., Horstink, M. W. & Gielen, S. C. (2006). Ambulatory motor assessment in Parkinson's disease. *Movement disorders: official journal of the Movement Disorder Society*, 21(1), 34–44.

Kim, H. B., Lee, W. W., Kim, A., Lee, H. J., Park, H. Y., Jeon, H. S., Kim, S. K., Jeon, B. & Park, K. S. (2018). Wrist sensor-based tremor severity quantification in Parkinson's disease using convolutional neural network. *Computers in biology and medicine*, 95, 140–146.

Lee, D. J., Dallapiazza, R. F., De Vloo, P. & Lozano, A. M. (2018). Current surgical treatments for Parkinson's disease and potential therapeutic targets. *Neural regeneration research*, 13(8), 1342.

López-Blanco, R., Velasco, M. A., Méndez-Guerrero, A., Romero, J. P., Del Castillo, M. D., Serrano, J. I., Rocon, E. & Benito-León, J. (2019). Smartwatch for the analysis of rest tremor in patients with Parkinson's disease. *Journal of the Neurological Sciences*, 401, 37–42.

Matzner, A., El-Hanany, Y. & Bar-Gad, I. (2020, May, 13). BEAT-PD DREAM Challenge HaProzdor [Synapse]. Consulted at https://www.synapse.org/#!Synapse:syn22041605.

Mayo Clinic. (2020, December, 8). Parkinson's disease - Diagnosis treatment [Online]. Consulted at https://www.mayoclinic.org/diseases-conditions/parkinsons-disease/diagnosis-treatment/drc-20376062.

Mera, T. O., Heldman, D. A., Espay, A. J., Payne, M. & Giuffrida, J. P. (2012). Feasibility of home-based automated Parkinson's disease motor assessment. *Journal of neuroscience methods*, 203(1), 152–156.

Michael J Fox Foundation. (2016). Fox Wearable Companion App by Michael J Fox Foundation. Consulted at https://appadvice.com/app/fox-wearable-companion-app/1072311255.

Movement Disorder Society Task Force on Rating Scales for Parkinson's Disease. (2003). The unified Parkinson's disease rating scale (UPDRS): status and recommendations. *Movement Disorders*, 18(7), 738–750.

Myin-Germeys, I., Oorschot, M., Collip, D., Lataster, J., Delespaul, P. & Van Os, J. (2009). Experience sampling research in psychopathology: opening the black box of daily life. *Psychological medicine*, 39(9), 1533–1547.

Olanow, C. W., Watts, R. L. & Koller, W. C. (2001). An algorithm (decision tree) for the management of Parkinson's disease (2001):: Treatment Guidelines. *Neurology*, 56(suppl 5), S1–S88.

Page, A., Javidnia, M., Smith, G., Zielinski, R. & Venuto, C. (2020, May, 13). BEAT-PD DREAM Challenge Roc-PD [Synapse]. Consulted at https://www.synapse.org/#!Synapse:syn21781334/wiki/602481.

Papadopoulos, A., Kyritsis, K., Klingelhoefer, L., Bostanjopoulou, S., Chaudhuri, K. R. & Delopoulos, A. (2019). Detecting Parkinsonian Tremor from IMU Data Collected In-The-Wild using Deep Multiple-Instance Learning. *IEEE Journal of Biomedical and Health Informatics*.

Parkinson's Foundation. (2018). Statistics. Consulted at https://www.parkinson.org/Understanding-Parkinsons/Statistics.

Patel, S., Lorincz, K., Hughes, R., Huggins, N., Growdon, J., Standaert, D., Akay, M., Dy, J., Welsh, M. & Bonato, P. (2009). Monitoring motor fluctuations in patients with Parkinson's disease using wearable sensors. *IEEE transactions on information technology in biomedicine*, 13(6), 864–873.

Post, B., Merkus, M. P., de Bie, R. M., de Haan, R. J. & Speelman, J. D. (2005). Unified Parkinson's disease rating scale motor examination: are ratings of nurses, residents in neurology, and movement disorders specialists interchangeable? *Movement disorders: official journal of the Movement Disorder Society*, 20(12), 1577–1584.

Quinn, N., Critchley, P. & Marsden, C. D. (1987). Young onset Parkinson's disease. *Movement disorders: official journal of the Movement Disorder Society*, 2(2), 73–91.

Ramji, V., Hssayeni, M., Burack, M. A. & Ghoraani, B. (2017). Parkinson's disease medication state management using data fusion of wearable sensors. *2017 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*, pp. 193–196.

Reeve, A., Simcox, E. & Turnbull, D. (2014). Ageing and Parkinson's disease: why is advancing age the biggest risk factor? *Ageing research reviews*, 14, 19–30.

Rigas, G., Tzallas, A. T., Tsalikakis, D. G., Konitsiotis, S. & Fotiadis, D. I. (2009). Real-time quantification of resting tremor in the Parkinson's disease. *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 1306–1309.

Rigas, G., Gatsios, D., Fotiadis, D. I., Chondrogiorgi, M., Tsironis, C., Konitsiotis, S., Gentile, G., Marcante, A. & Antonini, A. (2016). Tremor UPDRS estimation in home environment. *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 3642–3645.

Salarian, A., Russmann, H., Wider, C., Burkhard, P. R., Vingerhoets, F. J. & Aminian, K. (2007). Quantification of tremor and bradykinesia in Parkinson's disease using a novel ambulatory monitoring system. *IEEE Transactions on Biomedical Engineering*, 54(2), 313–322.

San-Segundo, R., Montero, J. M., Barra-Chicote, R., Fernández, F. & Pardo, J. M. (2016). Feature extraction from smartphone inertial signals for human activity segmentation. *Signal Processing*, 120, 359–372.

San-Segundo, R., Zhang, A., Cebulla, A., Panev, S., Tabor, G., Stebbins, K., Massa, R. E., Whitford, A., de la Torre, F. & Hodgins, J. (2020). Parkinson's Disease Tremor Detection in the Wild Using Wearable Accelerometers. *Sensors*, 20(20), 5817.

Schaff, J. (2017, November, 14). Feature Generation to predict Parkinson's Bradykinesia, Dyskinesia, and Tremor Symptoms [Synapse]. Consulted at https://www.synapse.org/#!Synapse:syn10611666/wiki/493491.

Schreiber, T. & Schmitz, A. (1997). Discrimination power of measures for nonlinearity in a time series. *Physical Review E*, 55(5), 5443.

Sherman, H. & Palisoc, J. (2020, October, 30). World first: focused ultrasound opens blood-brain barrier for delivery of therapeutic in Parkinson's disease [Format]. Consulted at https://sunnybrook.ca/research/media/item.asp?c=2&i=2204&f=world-first-focused-ultrasound-parkinsons-disease.

Sieberts, S. K., Schaff, J., Duda, M., Pataki, B. Á., Sun, M., Snyder, P., Daneault, J.-F., Parisi, F., Costante, G., Rubin, U. et al. (2020). Crowdsourcing digital health measures to predict Parkinson's disease severity: the Parkinson's Disease Digital Biomarker DREAM Challenge. *bioRxiv*.

Slabaugh, G. G. (1999). Computing Euler angles from a rotation matrix. *Retrieved on August*, 6(2000), 39–63.

Spieker, S., Boose, A., Breit, S. & Dichgans, J. (1998). Long-term measurement of tremor. *Movement disorders*, 13(S3), 81–84.

Stott, S. (2019, 03, 06). We need to talk about the UPDRS. Consulted at https://scienceofparkinsons.com/2019/03/06/updrs/.

Suzuki, M., Mitoma, H. & Yoneyama, M. (2017). Quantitative analysis of motor status in Parkinson's disease using wearable devices: From methodological considerations to problems in clinical applications. *Parkinson's Disease*, 2017.

Tsiouris, K. M., Gatsios, D., Rigas, G., Miljkovic, D., Seljak, B. K., Bohanec, M., Arredondo, M. T., Antonini, A., Konitsiotis, S., Koutsouris, D. D. et al. (2017). PD_Manager: an mHealth platform for Parkinson's disease patient management. *Healthcare technology letters*, 4(3), 102–108.

Tsipouras, M. G., Tzallas, A. T., Rigas, G., Bougia, P., Fotiadis, D. I. & Konitsiotis, S. (2010). Automated Levodopa-induced dyskinesia assessment. *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*, pp. 2411–2414.

Tsipouras, M. G., Tzallas, A. T., Fotiadis, D. I. & Konitsiotis, S. (2011). On automated assessment of Levodopa-induced dyskinesia in Parkinson's disease. *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 2679–2682.

Tsipouras, M. G., Tzallas, A. T., Rigas, G., Tsouli, S., Fotiadis, D. I. & Konitsiotis, S. (2012). An automated methodology for levodopa-induced dyskinesia: assessment based on gyroscope and accelerometer signals. *Artificial intelligence in medicine*, 55(2), 127–135.

Tzallas, A. T., Tsipouras, M. G., Rigas, G., Tsalikakis, D. G., Karvounis, E. C., Chondrogiorgi, M., Psomadellis, F., Cancela, J., Pastorino, M., Waldmeyer, M. T. A. et al. (2014). PERFORM: a system for monitoring, assessment and management of patients with Parkinson's disease. *Sensors*, 14(11), 21329–21357.

Um, T. T., Pfister, F. M., Pichler, D., Endo, S., Lang, M., Hirche, S., Fietzek, U. & Kulić, D. (2017). Data augmentation of wearable sensor data for parkinson's disease monitoring using convolutional neural networks. *Proceedings of the 19th ACM International Conference on Multimodal Interaction*, pp. 216–220.

Van Hees, V. T., Gorzelniak, L., Leon, E. C. D., Eder, M., Pias, M., Taherian, S., Ekelund, U., Renström, F., Franks, P. W., Horsch, A. et al. (2013). Separating movement and gravity components in an acceleration signal and implications for the assessment of human daily physical activity. *PloS one*, 8(4), e61691.

Vanrell, S. R., Milone, D. H. & Rufiner, H. L. (2017). Assessment of homomorphic analysis for human activity recognition from acceleration signals. *IEEE journal of biomedical and health informatics*, 22(4), 1001–1010.

Wang, J., Song, X. & Farahi, A. (2017). Parkinson's Disease Digital Biomarker DREAM Challenge. 1.

We Move. (2006). Unified Parkinson's Disease Rating Scale. Consulted at https://img.medscape.com/fullsize/701/816/58977_UPDRS.pdf.

Wen, Q., Sun, L., Song, X., Gao, J., Wang, X. & Xu, H. (2020). Time Series Data Augmentation for Deep Learning: A Survey. *arXiv preprint arXiv:2002.12478*.

Wooten, G., Currie, L., Bovbjerg, V., Lee, J. & Patrie, J. (2004). Are men at greater risk for Parkinson's disease than women? *Journal of Neurology, Neurosurgery & Psychiatry*, 75(4), 637–639.

Zhan, A., Mohan, S., Tarolli, C., Schneider, R. B., Adams, J. L., Sharma, S., Elson, M. J., Spear, K. L., Glidden, A. M., Little, M. A. et al. (2018). Using smartphones and machine learning to quantify Parkinson disease severity: the mobile Parkinson disease score. *JAMA neurology*, 75(7), 876–880.

Zhang, A., Cebulla, A., Panev, S., Hodgins, J. & De la Torre, F. (2017a). Weakly-supervised learning for Parkinson's disease tremor detection. *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 143–147.

Zhang, A., De la Torre, F. & Hodgins, J. (2020). Comparing laboratory and in-the-wild data for continuous Parkinson's Disease tremor detection. *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pp. 5436–5441.

Zhang, H., Cisse, M., Dauphin, Y. N. & Lopez-Paz, D. (2017b). mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.