Classification of hyperkinesia in Parkinson patients using mobile sensors

Gustaf von Grothusen

Master's Thesis
Supervised by
Prof. Andreas Jakobsson

LUND UNIVERSITY

Dept. of Mathematical Statistics
Lund University
Sweden
Fall 2022

1 Abstract

In this thesis, we explore the possibility to monitoring hyperkinesia in people who suffer from Parkinson's disease (PD) using sensors in mobile phones. This is done by first collecting data from 25 patients diagnosed with PD by using a sensor-recording smartphone in a bag attached to the stomach, and at the same time let trained professionals make assessments of the degree of which they show signs of hyperkinesia, on the clinical dyskinesia rating scale, CDRS. Given the labels and the sensor data, a set of models has been trained. Both models for binary classification, i.e., predicting the presence of hyperkinesia vs no hyperkinesia, as well as models aiming to estimate the CDRS score were investigated. As the available data is a mere 429 samples, a key part of the work has been to self-engineer features descriptive to signs of hyperkinesia. The proposed models are kernel support vector machines for both the binary classification and for the regression. The proposed method provides results that are in line with what can be expected of an assessment by a trained professional.

1.1 Keywords

Dyskinesia, hyperkinesia, Parkinson's disease, support vector machine, SVM, SVR, SVC, feature engineering, digital phenotyping, CDRS, smartphone, accelerometer

2 Glossary

2.1 Medical terms and abbreviations

Dyskinesia - ”Dys-mobility”: Bad mobility, either too high or too low.
Hyperkinesia - Hyper mobility. Subclass of dyskinesia.
Hypokinesia - Under mobility. Subclass of dyskinesia.
Tremor - Shaking movement
ET - Essential tremor
PD - Parkinson’s disease
PT - Physiological tremor
CDRS - Clinical dyskinesia rating scale, score of 0-4.
Precursor - Chemical that tightly relates to another, target chemical: When a precursor of chemical A is introduced in the body, it quickly reacts so it becomes chemical A with the help of natural processes in the body.
Levodopa - Drug; Precursor to dopamine.

2.2 Technical terms

SVM - Support vector machine
SVC - Support vector classification
SVR - Support vector regression
MSE - Mean squared error
MAE - Mean absolute error
ρ - Pearson correlation
NN - Neural network.
std - Estimate of standard deviation

2.3 Notation

$|| \cdot ||_i$ - i:th vector norm. If i is not stated, 2-norm is assumed.
$\mathbf{v}$ - Vectors are denoted as bold lowercase letters.
A - Matrices are denoted as capital non-bold letters.
Lowercase $\phi_k(\cdot, ...)$ - Self engineered feature k.
Uppercase $\Phi(\cdot, ...)$ - Auto-generated feature.

3 Introduction

In this thesis, we examine the possibility of using mobile phone sensors for classifying hyperkinesia in humans with Parkinson's disease.

In an attempt to justify the subject, the following should be noted: In all forms of healthcare, feedback is essential. In the short term, for determining or tweaking treatments for individuals. In the longer term, to determine what treatments or medications are most effective in general. Feedback is traditionally collected by:

- Assessment by a doctor
- Questionnaire
- Self assessment

All means of collecting feedback have an associated cost, monetary, social or both. To be assessed by a doctor, one would have to go to the hospital and back, being both costly and time consuming. To fill in a questionnaire can be boring, especially if the same questionnaire would have to be filled every day. One might then refrain from doing the questionnaire, or simply forget. Furthermore, the feedback is often granular and rather subjective [1].

Given that smartphones are carried by almost everyone, it would be highly valuable if it was possible to use it to collect feedback passively. It would imply a great cut of costs for collecting feedback, and the data available for decision making could be greatly extended. In this thesis, we examine whether it is possible to evaluate hyperkinesia in Parkinson's disease in this way using a phone.

3.1 Parkinson's disease

The main cause of Parkinson's disease is when neurons in the basal ganglia become impaired or die. The basal ganglia is an area of the brain which controls movement, and a key function of these cells is producing dopamine. When these cells are impaired, a lack of dopamine occurs, and it is this lack of dopamine that causes the main problems with the disease: tremors and hypokinesia (immobility).

For some additional context, a few quick facts are presented below:

- In Sweden, there are about 20000 people diagnosed with PD, most of whom are men[2].
- The disease commonly debutes in the ages 55-60 years [2].

- The former student of Lund University, Prof. Arvid Carlsson was awarded the Nobel prize in medicine in year 2000 for suggesting that lack of dopamine caused the disease [3].

3.1.1 Treatment

Parkinson's disease is most commonly treated with a drug called Levodopa, a precursor to dopamine. This artificially increases the levels of dopamine to counter the lack of dopamine in the patients. The approach, although effective, poses some drawbacks:

- One of the most prominent side effects is hyperkinesia. This is porttrayed in Figure 1.
- Taking the drug over a long period of time causes tolerance. In turn, this is countered by increasing the dose, which also can lead to even more hyperkinesia.
- A lot of the patients get a more or less standardised drug scheme, and because the sensitivity to the drug is individual, some patients get too large doses [1].

Each patient diagnosed with Parkinson's disease follow up with a doctor with an interval of approx 6 months, with a 30 minute session where they have an opportunity to be examined and discuss their medication and overall form. Ergo, the doctor has a quite small sample size to make quite an important decision: deciding the medication scheme. In fact, the time during which the patients are examined is a mere 200 ppm of their time awake ($60/(60 * 16 * 365) = 1/(16 * 365) = 1/5840 \approx 200\text{ppm}$).

Furthermore, there is risk of getting a biased set of samples during the examination. How could the doctor know if the patient is showing symptoms representative for their overall time awake? This problem is hard to address, but is potentially addressable with the type of sensors smartphones posses.

3.2 Clinical study

In on ongoing study lead by Dr. Grigoriou et al., it is investigated whether a combination of Levodopa and ropinirole can reduce the hyperkinesia that sometimes comes with taking Levodopa alone. Levodopa is a replacement for dopamine, while ropinirole instead is targeting the receptors in the synapse, rendering them more sensitive. The goal is to see if they can measure a significant difference in the degree of which the participants are affected by hyperkinesia, by only altering their medication.

The study includes 25 patients diagnosed with PD in ages 48 - 81 and each of whom were participating in measurements on two occasions: one where they

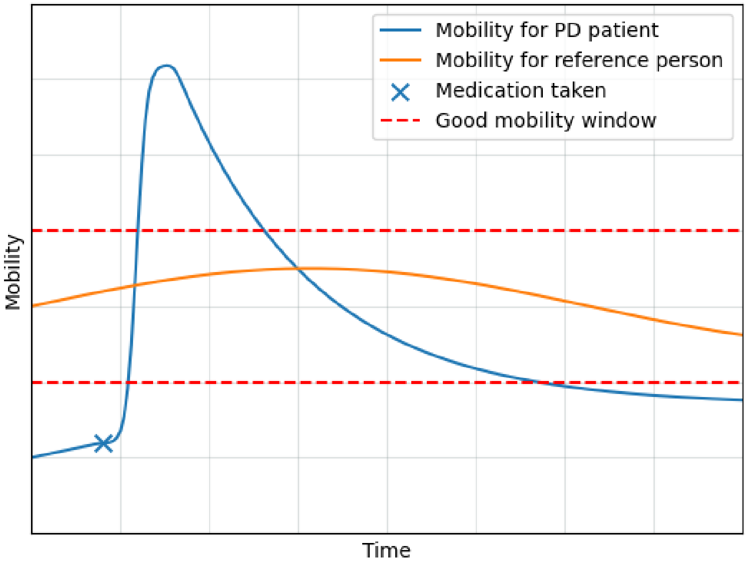

Figure 1: Taking the medication can cause the mobility to spike, and cause hyperkinesia. However, it is often preferable to have hyperkinesia to some degree rather than having the opposite, hypokinesia [1].

got the traditional Levodopa alone, while on the other occasion they received a combination of Levodopa and ropinirole. Each of the patients was scheduled to do 10 measurements on each occasion, but due to feeling ill, some of them did not follow through on all their scheduled measurements. On average, the patients did 17 measurements each, resulting in 429 samples being available in total.

During each measurement, the patients were asked to do a set of 4 tasks. These included sitting still, pretending to drink a glass of water, putting a lab coat on and taking it off again, and walking around 10 meters. This took them a couple of minutes, each time.

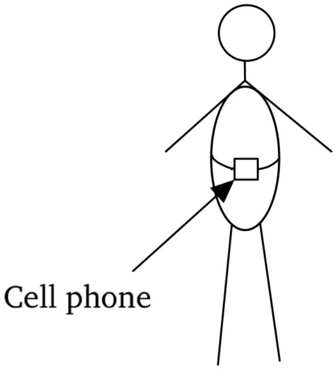

Figure 2: The cell phone's attachment to the participants' stomach.

Each sample was labeled on the spot by a professional, and each occasion was also filmed. The films were labeled by two other experienced physicians, that unlike the doctor present at the measurement did not know about which medication the patient had received. Only the two labels done with video data is considered in this thesis. All labels were quite detailed with different scores of the hyperkinesia over different parts of the body. Among other things, the hyperkinesia in the torso was specifically assessed with a CDRS score. It is this torso hyperkinesia that is further examined in this work.

3.3 Rating of hyperkinesia

To assess to which degree a person has dyskinesia (collective term for hyper- and hypo- kinesia), the clinical dyskinesia rating scale or CDRS is used. It is defined as in Table 1. The definition leaves room for interpretation, which sometimes causes dubious assessments even from trained professionals.

CDRS	Description
0	No dyskinesia
1	Questionable or mild dyskinesia
2	Moderate dyskinesia
3	Severe dyskinesia
4	Incapacitating dyskinesia

Table 1: Definition of CDRS. For the rest of this report, we will only assess hyperkinesia.

3.4 Aim

The aim of this work is to show that it is possible to utilise mobile sensors to passively collect valuable medical feedback. To be able to deploy a system to the real world and relying upon the outputs, one would need extensive testing of it. The scope of this work is therefore not to show a complete system, but instead to shine a light upon the possibility and perhaps encourage further research in the subject area.

Furthermore, there are two levels at which this problem can be addressed:

- Estimate the CDRS, given that the person is sitting down.
- Estimate the CDRS, not knowing at all what the person is doing.

In the work leading up to this report, both these problems were addressed. However, because the usability of a model that can address the problem at the second level is much higher, this has been the main target, and is what will be assessed for the main part in this report, although results for the both problems are presented in the results section.

Level 1	Model trained on data truncated to when participants are sitting down
Level 2	Model trained on all data

3.5 Limitations of this report

In assessing the level of hyperkinesia in the participants, and getting a more complete picture of the problem nature, the recorded videos have been of noTable importance. As access to these videos is restricted to personel working with the study, these had to be omitted in the presented analysis.

4 Previous work

Many studies have investigated similar problems although with slightly different circumstances and goals. A lot of them has approximately the same amount of data available, but the experiment setup and hence the data collection is made in many different varieties. For example, it is common to use more than one sensor [4, 5, 6]. A common method is also to utilise smartwatches [7, 8]. Further, the prior work is most commonly assessing the problem of determining the degree of tremors, not dyskinesia, either in the case of PD or ET. Below follows an overview of the two studies with most impact on this thesis.

4.1 Classification of hyperkinesia in Parkinson's disease

4.1.1 Circumstances

In Erik Liljeroth's thesis [9], a similar problem as in this thesis was addressed, with part of the same data as is available at the time of this thesis. One notable difference between the studies that during the work in [9], there was only one label available per sample, while there were two available in this thesis.

4.1.2 Method

The methods used in [9] were mainly based on artificial neural networks, applied in different constellations, for example a 1D-CNN. The results presented were promising, but a main problem to this approach was the scarcity of data.

4.1.3 Takeaways

Artificial neural networks have been found to outperform other methods under some circumstances. For example, 2D-convolutional networks are really good at evaluating images. One of the reasons why they are so good, is that there is a lot of local patterns that can be exploited. But when it is not certain that these local patterns exist, or are very complex to decipher, another approach might be more suiTable. Perhaps the most important reason as to why neural networks have become so popular is that they scale better with more data than traditional learning methods, such as logistic regression or support vector machines. But in a problem where data is scarce, this scaling property is not utilizable. For example, in a study by Shultz MA et al. [10], they compared the performance of a binary classification problem on brain images, for different models, altering the amount of samples available for training. They found that classical methods (such as logistic regression and support vector machines) outperform ANNs with datasets of sizes comparable to the one used in this study.

4.2 Classification of essential tremor with fuzzy reasoning

In a study by Caro Fuchs et.al. [11], they tried to classify tremor severity on patients suffering from ET. ET is a neurological disease that causes tremors mainly in the extremities of the patient. The main difference in the symptoms is that ET tremors only show up either in postural or transitional poses. Here, postural means having a pose which requires energy to maintain, for example holding out a hand straight in front of your body. By transitional, one implies transitioning between poses, for example moving the hand from being sprawled in front of the body to pointing at ones nose.

A main difference with the problem studied in [11] is that they in this study are assessing tremors, which have distinct features in the frequency domain. In the problem addressed in this thesis, the movement is more random.

4.2.1 Aim

The study intends to ”assess severity of tremor objectively, to be better able to asses improvement in ET patients due to deep brain stimulation or other treatments.”

4.2.2 Circumstances

Smart phones were strapped to the back of both wrists of the patients and sampled accelerometer and gyroscope data at 100Hz. Ground truth was given by a trained professional.

4.2.3 Results

They managed to get a mean absolute error of 1.75 on a scale of 0-84, on a validation set of 25% of the patients.

4.2.4 Takeaways

The main take away from this study is the fact they used quite simple features along with a quite simple model, in comparison with for example a deep neural network, and got better results. The reason they mention as a means of success is the fact that when using such a simple model, it is a lot easier to get a feel for what features are really descriptive and what they are describing, as well as how the model operates under the hood, which can be quite tricky with large models.

5 Method

5.1 Overview

The following will be covered in this section:

- Detailed circumstances under which the work is constrained.
- The embraced workflow.
- Detailed overview over how the final model works.

In the interest of conciseness of the report, the partial results presented in this section are w.r.t. to the "level 2" problem, i.e., not truncating the data to any specific task. The final results for the "level 1" problem is presented in the results section, but unless explicitly stated, any results presented are w.r.t. the harder "level 2" problem.

5.2 Available data

An important aspect of this problem is the fact that the labels from different persons differ, although their correlation is high. To visualize the correlation between the different labels, the confusion matrix is suiTable,

$$C = \begin{pmatrix} 237 & 7 & 1 & 0 \\ 27 & 38 & 7 & 0 \\ 22 & 20 & 39 & 2 \\ 0 & 2 & 17 & 10 \end{pmatrix},$$

where the i:th row corresponds to expert A's labels and the j:th column to expert B's. A visualisation of the confusion matrix can be seen in Figure 3. The confusion matrix testifies the fact that making this classification is hard even for experienced professionals.

5.2.1 Ground truth

As the doctors' assessments differ, we should ask ourselves what should be used as ground truth for the construced model. Is it reasonable to use the mean, or could some other method be applied?

	ρ	mean absolute difference
all samples	0.790	0.303
non zero	0.641	0.677

Table 2: Metrics to display how much the labels from different people differ for all available data. When there is no hyperkinesia going on at all, the assessment is easier, and to get a more complete picture, the metrics for nonzero examples are also displayed.

As the labels are perturbed, we next evaluate whether using the mean of the two available labels is a reasonable choice of ground truth. Let

$$y_{est} = y_{true} + \epsilon + b, \tag{1}$$

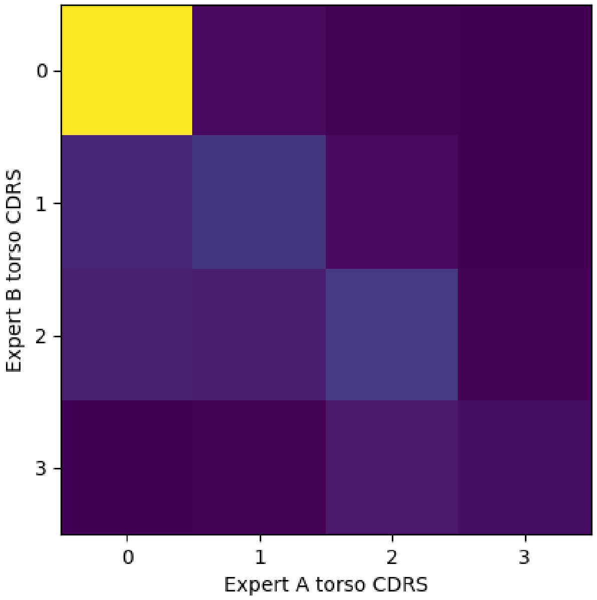

Figure 3: Confusion matrix over all the labels by the different experienced physicians. Although there is strong correlation between the two, it is clear that the correlation is not perfect.

for some random noise variable $\epsilon \in N(0, \sigma)$ and a bias term, $b \in \mathbb{R}$. The model here is that depending on who is making the assessment, the estimation can be biased, and at the same time contain Gaussian noise. The fact that the underlying property is hard to estimate accurately even for trained humans has an interesting implication: If a model can align with each of the different sets of labels as good as or better than they align with each other, the model is of value.

Further assessing the model of the underlying true property, y_{true}, the mean is formed as

$$y_{gt} = \frac{y_{est1} + y_{est2}}{2} = y_{true} + \frac{\epsilon_1 + \epsilon_2 + b_1 + b_2}{2}. \tag{2}$$

. The error is then distributed as (assuming y_{est1} and y_{est2} are independent)

$$\frac{\epsilon_1 + \epsilon_2 + b_1 + b_2}{2} \in N\left(\frac{b_1 + b_2}{2}, \frac{\sqrt{\epsilon_1^2 + \epsilon_2^2}}{2}\right). \tag{3}$$

The biases are hard to address, and there is no reason to believe any bias would be stronger than the other. Further, the new standard deviation, $\sqrt{\epsilon_1^2 + \epsilon_2^2}/2$, in the case of $\epsilon_1 \approx \epsilon_2$ implies a smaller standard deviation by approximately a factor of $\sqrt{2}$. Since there is no reason to believe any b or ϵ is larger than any other, the mean is a reasonable way forward. This will be referred to as the label for the rest of this report.

5.2.2 Splitting up the available data

The data is first split into two chunks. One chunk is for training, which is used for both training and validating the models during development, and one chunk for testing, that is kept separate and untouched, to get an as honest as possible estimation of the model's accuracy. To determine a suitable test set, we should first have a look at the labels.

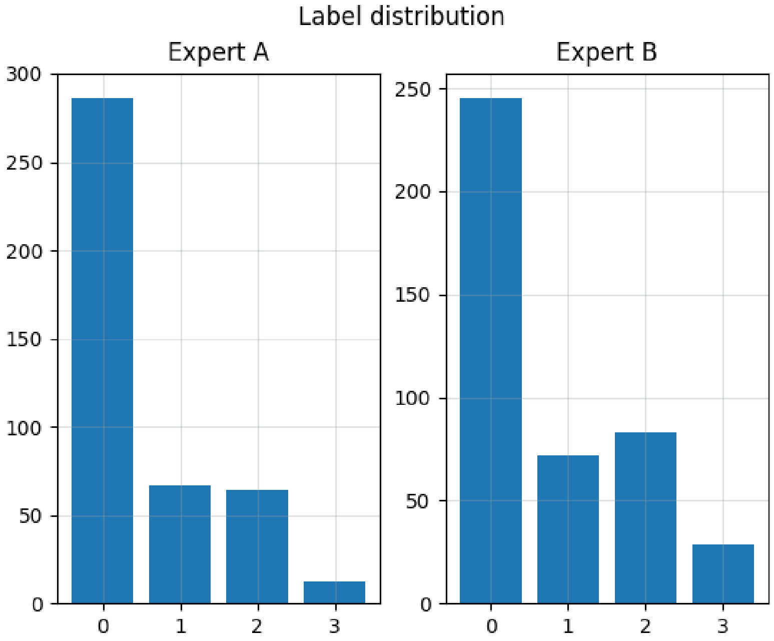

Figure 4: Individual label distribution for the doctors, respectively. We can see that the label 0 by far is the most common one, and that there are no samples for which the CDRS is 4.

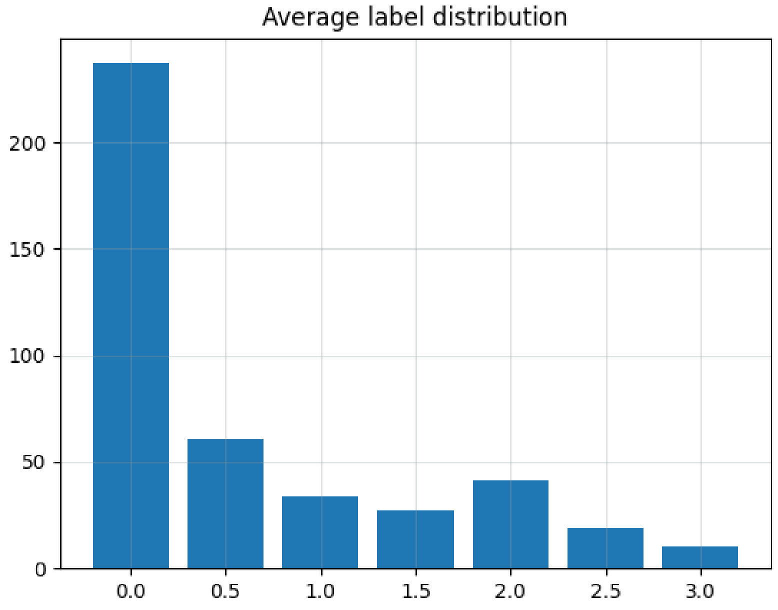

Figure 5: Distribution over the average of the two doctors' labels.

The most prominent observation is that the data set is imbalanced. There are a number of measures that can be taken to counter this, for example SMOTE [17]. The most important aspect is that inaccurate estimates of the model performance can be retrieved, especially since the training and validation splits are chosen at individual level, not at the sample level. For example, the number of individuals with severe hyperkinesia (2 or more) is a mere 5. If the training/ validation split is then chosen such that all individuals with severe hyperkinesia is put into one of the chunks, the estimate of the model performance will become unreliable.
The available data is split in the following way:

1. The data is split into two chunks, for training and testing, with approximate proportions $\frac{3}{4}$ and $\frac{1}{4}$, respectively. The testing data is not used for any validation until the final model is validated in a last step. It is made sure that no patient in the testing data is in the training data, to better be able to make conclusions about generalisation when validating the models. Further, to make sure the training data is somewhat representative of the testing data, a small stratification scheme was employed: When having sorted the patients w.r.t. amount of average labels of 2 or above, traversing this list in a descending order, every fourth patient was chosen for the test set. The remaining patients (that did not have any average labels of 2 or above) were split at random with the given proportions.

2. The training data from the previous step is then used to train a set of models. The split within this data is done in the same manner as the training- and testing data was picked from the entire data set.

With the test set chosen, we can evaluate whether it seems representative of the training set. As seen in Tables 3 and 4, the correlation between the different assessments is lower in the test set than on the training set. This should be taken into account when assessing the models.

	ρ	mean absolute difference
all samples	0.818	0.301
non zero	0.606	0.697

Table 3: The two experts' assessments correlation and mean absolute difference for the training/validation data set.

	ρ	mean absolute difference
all samples	0.686	0.309
non zero	0.605	0.633

Table 4: The two experts' assessments correlation and mean absolute difference for the testing data set.

5.2.3 Data scarcity comments

Lastly, if putting the problem into a machine learning perspective, and comparing it to other problems, it can be noted that the cost of each sample is quite high. For example, sourcing vast amounts of images and annotating them can be done at reasonable cost. An image takes perhaps 30 seconds to label with negligible peripheral costs. In contrast, to collect one data sample in this case takes 20 minutes, and has a lot of peripheral costs associated to it, such as planning and coordinating patient visits, recording the video of the patient, synchronizing the time between the sensor and the video camera, structuring the video data, getting different doctor's assessments of the video, i.e. This is an important aspect to keep in mind while thinking about future prospects of the subject area. Part of the problem nature is that data is highly expensive.

5.3 Modelling methodology

The predictor by Fuchs et al. worked with quite a small set of features and a relatively simple model [11]. Is this true for determining hyperkinesia under our circumstances as well? Hyperkinesia is a measure of how much a person is moving involuntarily. But what is a good abstraction for this property, what are good features for describing it? When this question has been answered, there is also the problem of choosing a good training scheme.

By and large, the work has been conducted in the following way:

1. Find good features.
2. Find a good model to best utilize the features.

To determine what features are good, a bottom up strategy has been employed. By bottom up, it is meant that each feature has been more or less handcrafted and then investigated individually to see if it is descriptive and should be included in the final model. This is in contrast to the most common way of doing things: Throwing in all features available and later removing the less descriptive ones with different regularization techniques. The reason for the bottom-up approach can be motivated by the following:

- Making a model with a large set of features and weights gives room for overfitting. If a model can be made with only a small set of features, the risk of overtraining is made smaller [12].
- The problem is potentially solvable with only a small set of features: the problem of determining hyperkinesia should not be as hard as, for example, predicting the weather.
- In comparison with many deep learning problems, the amount of available data is low. For example, the classic MNIST dataset has 70000 samples [14], while we have a mere 429 in our case. This means that overfitting is a big risk that needs to be assessed in choosing the approach. Recall that overfitting can be done at an architectural or hyperparameter level and not just in the parameter space, so while hunting a small validation error by tuning hyperparameters and tweaking the architecture, there is an induced risk of overtraining.

Given the bottom-up approach, the largest part of the work has been to craft suitable features from the available data. The way this work has been conducted is by using linear regression on the investigated features and examine the loss. A big advantage with such a simple model is that the behaviour of the model can be easily understood by looking at the weights.

Looking at raw accelerometer data is not sufficient. The acceleration data it will be strongly affected by the presence of gravity which will be substantially larger than any of the variations we are interested in, such that

$$\mathbf{a}_{tot} = \mathbf{a}_{lin} + \mathbf{g}, \tag{4}$$

where it is reasonable to assume that $||\mathbf{g}|| >> ||\mathbf{a}_{lin}||$ almost always. The first step is hence to filter out the influence of gravity.

5.3.1 Filtering out gravity

To do this rigorously is quite hard because there is no easy way to acquire ground truth for such data, i.e., which part of the signal is truly gravity and what part is linear acceleration. What can be noticed however, is that Apple does this in their Core Motion package, they utilize the gyroscope and accelerometer to get a good estimate of gravity, and then being able to subtract it from the raw data [15]. Assuming their estimate is good, we can try and mimic the same performance with our own method, since integrating core motion in our own code base would be tedious.

To make use of core motion package, the free iPhone application "Tremor12" [16] was used to collect four test cases. This inherently uses the package and directly outputs a direction of gravity as well as an estimate of the linear acceleration. The key point is that it has split the actually measured accelerometer signal into two parts. Unfortunately, the raw accelerometer signal is not provided, but can be simulated by rescaling the (already normalized) gravity vector and adding the linear acceleration. The goal here is to, given this simulated total signal, separate it again to gravity and linear acceleration.

To employ this approach, a measure of proximity to the Apple solution is needed. Here, the average square distance between the modeled approximation and Apple's approximation of the gravity is used,

$$d = ||\mathbf{g}_{apple} - \mathbf{g}_{approx}||_2^2. \tag{5}$$

As a global loss for a sequence of testing data, the average of the above defined loss is formed, such that

$$L = \frac{1}{N}\sum_{i=1}^{N} d_i. \tag{6}$$

Further, to develop an intuition for the problem, the behaviour of the different methods was animated, and samples of these animations are linked on youtube throughout the text below.

In a first approximation, a simple causal moving average of 20 samples was applied to estimate gravity, forming

$$\mathbf{g} = C(z)\mathbf{x}, \tag{7}$$

where $\mathbf{g}$ is the estimate of gravity, $\mathbf{x}$ is the simulated raw data, and $C(z)$ is given by

$$C(z) = \frac{1}{20}\bigg(1 + z^{-1} + z^{-2} + ... + z^{-19}\bigg). \tag{8}$$

An animation of the approximation can be viewed here: `https://youtu.be/LkgDNcUC75k`

Test case	Causal moving average loss	Non causal moving average loss
1	0.00357	0.00012
2	0.00271	0.00029
3	0.00796	0.00028
4	0.01952	0.00038

Table 5: Losses for different test cases.

In contrast to real time applications, one may for this application also utilize a non causal filter, i.e., making benefit of the hindsight that we have here. So in a second approximation, a non-causal moving average was employed, with the corresponding transfer function,

$$C(z) = \frac{1}{40}\left(z^{-19} + z^{-18} + ... + z^{-1} + 1 + 1 + z + z^2 + ... + z^{19}\right). \tag{9}$$

An animation of this method can be viewed here: `https://youtu.be/kvpjzAunvJI`. As can be seen in the video, this approach works well compared to the previous attempt.

With these results we can move on with confidence that the method of separating gravity from linear acceleration is working well. Worth mentioning is that the gravity vector itself is quite informative: It tells us how the sensor is oriented, and this orientation could be used to craft features.

5.3.2 Evaluation

As we here examine models making binary classification as well as making regressions, we need ways to evaluate the models in these two cases.

For the binary predictor, the metrics accuracy, precision, recall, and f1-score are used. The definition of them are

$$\begin{aligned}
\text{accuracy} =& \frac{\text{number of correct predictions}}{\text{number of incorrect predictions}},\\
\text{precision} =& \frac{\text{true positives}}{\text{true positives + false positives}},\\
\text{recall} =& \frac{\text{true positives}}{\text{true positives + false negatives}},\\
\text{f1-score} =& 2\frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}.
\end{aligned} \tag{10}$$

Precision can be interpreted as an estimate of the probability that a sample predicted as true is really true, and recall can similarly be interpreted as an estimate of the probability that a true sample is detected or ”recalled”. F1-score is the harmonic mean of the two and is a widely used metric for evaluating classification models.

As metrics for evaluating a regression model, mean square error (MSE), the mean absolute error (MAE), max absolute error, and the Pearson correlation (ρ) are commonly used. The Pearson correlation is not such a common choice as a metric for evaluating model performance, but is a measure many medical professionals prefer. These metrics are evaluated for the training and validation data separately. The Pearson correlation coefficient is defined by

$$\rho_{X,Y} = \frac{C(X,Y)}{\sigma_X \sigma_Y}. \tag{11}$$

Due to the fact that a model is a function of both its training data and the architecture, it is vital to choose the training samples with care, ensuring that one does not introduce any biases to seemingly get better results, while in reality one is just losing generalization. A key part of the work has therefore been to evaluate models thoroughly, and partial results have played an important role in deciding the direction of the work. Therefore, partial results will be shown in the method section, as they are an important part of the feedback loop that is machine learning engineering. All results shown in the method section are to be regarded as indications of achievable performance that were received throughout the work, while the final results on the untouched testing data is saved for the results section.

5.3.3 Naive model

To determine what features are good or not, it is good to have a baseline of what a naive model is capable of. As a naive model, a simple mean of all the labels is taken as prediction, regardless of the input data. This is the baseline that all models, both simple and complex models should beat.

	MSE training	MSE validataion	MAE validataion	Max AE val
mean	0.64	0.64	0.618	2.511
std	0.108	0.108	0.073	0.163

Table 6: Compact form of how the evaluation metrics are distributed over different training / validation splits. Because the correlation coefficient is not defined for the naive model, whose output has standard deviation 0, it is left out.

5.4 Feature extraction and selection

In this section, it will follow a detailed overview of the methodology that has been applied in the search for descriptive features. The main objective is to find a model that both performs well, and with high probability is not overtrained to our small dataset. As a means of minimizing the risk of overtraining, a lot of energy has been put into finding a small set of features that are still descriptive enough to solve the problem. An empirical trial and error approach is used, where the only thing we alter are the features, and the models are evaluated with high regard to signs of overtraining. The hope is that the reader after this discussion is able to either extrapolate the methodology to other problems, or provide critique and feedback on how it may be improved.

To determine whether a feature or a set of features are efficient in describing the observed data, a linear model is applied,

$$y = \mathbf{w}^T \mathbf{x} + b, \tag{12}$$

where $\mathbf{x}$ denotes the features for a sample. To be clear, $\mathbf{x}$ is not the raw data, but the raw data for a sample, mapped to a feature (or set of features) by some function that is to be determined.

For completeness and thanks to the fact that this model is easily optimized in a least squares sense, the solution to obtaining a least-squares optimal solution will be detailed.

The LS estimate is formed by minimizing the MSE, i.e.,

$$\min_{\mathbf{w},b} \frac{1}{2} \sum_{i=1}^{N} \left(\mathbf{x}_i^T \mathbf{w} + b - y_i \right)^2, \tag{13}$$

where $\mathbf{x}_i$ is the feature vector for sample i and y_i is the label for sample i.

This function is strongly convex under weak conditions and may therefore be minimized efficiently. Let

$$f(\mathbf{w}, b) = \frac{1}{2} \sum_{i=1}^{N} \left(\mathbf{x}_i^T \mathbf{w} + b - y_i \right)^2. \tag{14}$$

Then, differentiate w.r.t. b:

$$0 = \frac{\partial f}{\partial b} = Nb + \sum_{i=1}^{N} \left(\mathbf{x}_i^T \mathbf{w} - y_i \right) \implies b = \bar{y} - \mathbf{w}^T \bar{\mathbf{x}}, \tag{15}$$

where $\bar{y}$ is the mean of all the elements in the label vector and $\bar{\mathbf{x}}$ is the mean of all the feature vectors in the training set. Inserting b in to (14) yields

$$f(\mathbf{w}, b) = \frac{1}{2} \sum_{i=1}^{N} \left(\mathbf{x}_i^T \mathbf{w} + b - y_i \right)^2 = \frac{1}{2} \sum_{i=1}^{N} \left(\mathbf{w}^T (\mathbf{x}_i - \bar{\mathbf{x}}) - (y_i - \bar{y}) \right)^2. \tag{16}$$

Now, let $\tilde{\mathbf{x}}_i = \mathbf{x}_i - \bar{\mathbf{x}}$ and $\tilde{y}_i = y - \bar{y}$ and reformulate the problem as

$$\min_{w} \frac{1}{2} \sum_{i=1}^{N} \left(\mathbf{w}^T \tilde{\mathbf{x}}_i - \tilde{y} \right)^2. \tag{17}$$

Form the matrix $\mathbf{X}$, where the i:th row contains $\tilde{\mathbf{x}}_i$, and also form the vector $\mathbf{Y}$ where the i:th element contains $\tilde{y}_i$. Then, the objective function can expressed as

$$f(\mathbf{w}) = \frac{1}{2}||\mathbf{X}\mathbf{w} - \mathbf{Y}||_2^2. \tag{18}$$

Taking the gradient and setting it to zero yields, assuming X has full column rank,

$$\frac{\partial f}{\partial w} = \mathbf{X}^T\Big(\mathbf{X}\mathbf{w} - \mathbf{Y}\Big) = \mathbf{0} \implies \mathbf{w} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}. \tag{19}$$

Assuming X has full column rank is reasonable, as we have much more samples than features.

Using this optimization scheme, one may quickly get a sense for what features are descriptive, both by looking at the loss and also by looking at the weights that correspond to each feature.

5.4.1 Magnitude of linear acceleration

The first feature that was investigated was the average magnitude of the linear acceleration:

$$\phi_1(\mathbf{A}) = \frac{1}{N}\sum_{t=1}^{N}||\mathbf{a}_t||_2^2, \tag{20}$$

where A can be viewed as a matrix of size $3 \times N_{samples}$, and $\mathbf{a}_t$ as the three-dimensional linear acceleration vector at time t, or in other words the t:th column of $\mathbf{A}$.
If this feature is large, it should mean a lot of involuntary movement, although it may be so for other reasons as well. Nevertheless, the least squares method described above was applied and the results can be found in Table **??**.

	training MSE	validation MSE	ρ training	ρ val	MAE	Max AE
mean	0.617	0.644	0.165	0.145	0.612	2.565
std	0.098	0.028	0.089	0.075	0.023	0.076

Sadly, the results are not significantly better than the naive model.

5.4.2 Mean of absolute values of the linear acceleration

If one instead keeps each dimension separate and extract three features, namely the mean of the absolute value of the linear acceleration in each direction, such that

$$\boldsymbol{\phi}_2(\mathbf{A}) = \frac{1}{N}\Bigg(\sum_{t=1}^{N}|a_x^{(t)}|, \sum_{t=1}^{N}|a_y^{(t)}|, \sum_{t=1}^{N}|a_z^{(t)}|\Bigg)^T = \frac{1}{N}\Bigg(||\mathbf{a}_x||_1, ||\mathbf{a}_y||_1, ||\mathbf{a}_z||_1\Bigg)^T, \tag{21}$$

might we improve our results? Here, more features are introduced, which gives the model more freedom to potentially overtrain, but also potential to find somewhat more complex relationships in the data.

	training MSE	validation MSE	ρ training	ρ val	MAE	Max AE
mean	0.461	0.51	0.519	0.482	0.523	2.374
std	0.074	0.028	0.066	0.024	0.022	0.105

Table 7: Summary of the distribution over the evaluation metrics. A large improvement is observed in terms of correlation with the ground truth.

5.4.3 Moving average of linear acceleration

A next natural step is to instead look at moving averages of the linear acceleration. A possible feature is:

$$\phi_3(\mathbf{x}, n_C) = ||C(z)\mathbf{x}||_1, \tag{22}$$

where $C(z)$ is the filter $\frac{1}{n_C}(z^{-n_C+1} + z^{-n_C+2} + ... + z^{-1} + 1)$. This feature applied to the linear acceleration in the three directions, with n_C of 10, 20 and 30 yields the results in Table 8.

	training MSE	validation MSE	ρ training	ρ val	MAE	Max AE
mean	0.207	0.276	0.818	0.769	0.391	2.196
std	0.04	0.026	0.041	0.02	0.02	0.265

Table 8: Evaluation metrics for $\phi_3(\cdot, n_C)$ applied to linear acceleration in three dimensions, for all $n_C \in \{10, 20, 30\}$, which makes for 9 features in total. The results are quite promising, beating all previous attempts.

5.4.4 Direction of gravity

Proceeding, we examine a feature that examines the direction of gravity changes, is proposed. Because the direction of gravity at any time can be parameterized by two angles, instead of three cartesian components, the coordinates are mapped by first normalizing it to length one and then applying the function

$$f(\mathbf{g}_t) = (\arccos(g_2)\ \arctan(\frac{g_1}{g_3}))^T. \tag{23}$$

With the way the order of the components has been chosen, the two angles, for the rest of this report are named θ and φ, and interpreted as follows: θ is the latitudinal and φ is the longitudinal angle in a spherical coordinate frame of reference. Hence, θ is a measure of the pitch of the torso, and φ is a measure of roll. An example of this is shown in Figure 6. The linear acceleration is also

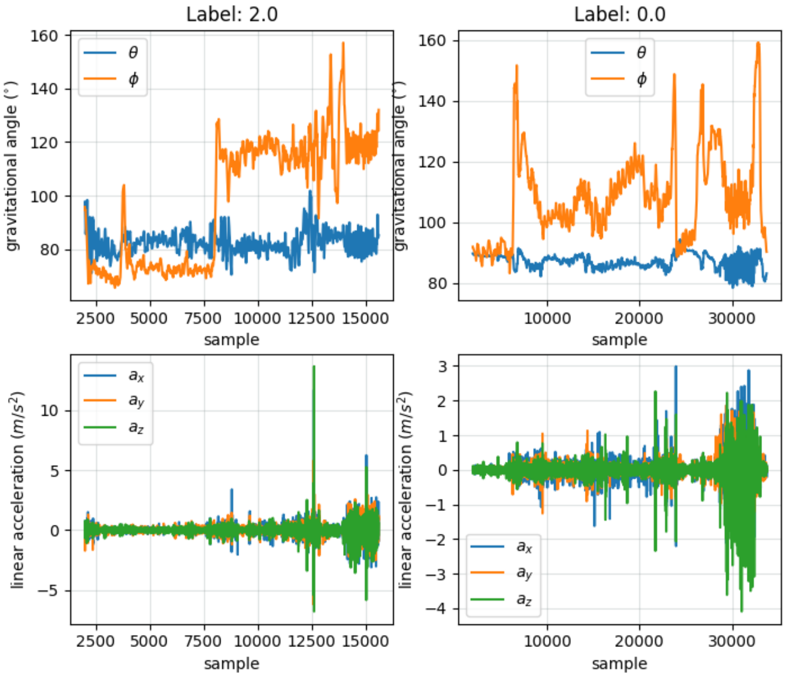

Figure 6: As seen in the Figure, there are no clear patterns in this example. What can be observed, is that even though both the low and high label samples have a lot of high frequency movement over all, the degree of movement in the θ angle is much lower in the lower-labeled case than in the high-labeled example.

plotted for comparison.

A further refined feature may be formed by dividing the sample in segments of size L, with stride S and for each of those segments calculate the variance of the signal within the window. The stride here is the step size used when segments are chosen. For example, having a sequence of 500 samples, subsample length $L = 250$, and a stride of $S = 50$, one might create $(500 - L)/S + 1 = 6$ subsamples. The proportion of segments that have a variance above a predetermined threshold may then be used as a feature:

$$\phi_4(\mathbf{x}, T, L, S) = \frac{\text{number of subsamples with variance below threshold}}{\text{number of subsamples}}, \quad (24)$$

where $\mathbf{x}$ is a one-dimensional series of data, T is the threshold, L is the subsample length, and S is the stride used to create the subsamples. Extracting this feature from θ alone yields a model with the performance summarized Table 9.

	training MSE	Validation MSE	ρ training	ρ val	MAE	Max AE
mean	0.301	0.319	0.723	0.722	0.442	1.739
std	0.044	0.016	0.034	0.008	0.013	0.134

Table 9: Evaluation metrics when using only the feature $\phi_4(\boldsymbol{x}, T = 0.1, L = 200, S = 100)$ applied to θ.

	training MSE	validation MSE	ρ training	ρ validation	MAE	max AE
mean	0.286	0.306	0.74	0.736	0.43	1.701
std	0.042	0.015	0.032	0.008	0.013	0.129

Table 10: Evaluation metrics for the case where the feature $\phi_4(\boldsymbol{x}, T = 0.1, L = 200, S = 100)$ is applied to θ and φ. MAE and max AE metrics are for the validation data.

As the performance using ϕ_4 seems promising, the following features on the same theme were investigated:

$$\begin{aligned} \phi_5(\mathbf{x}, L, S) &= \text{standard deviation of subsample variances}, \\ \phi_6(\mathbf{x}, L, S) &= \text{mean of subsample variances} \end{aligned} \tag{25}$$

The results using this set of features on the angles of gravity are displayed below

	training MSE	validation MSE	ρ training	ρ validation	MAE	max AE
mean	0.198	0.255	0.827	0.789	0.371	2.382
std	0.034	0.061	0.035	0.04	0.016	1.132

Table 11: Evaluation metrics for the case where features $\phi_4(\boldsymbol{x}, T, L, S)$, $\phi_5(\boldsymbol{x}, L, S)$, and $\phi_6(\boldsymbol{x}, L, S)$ applied to θ, for $T = 0.1$, $L = 200$ and $S = 100$. Here, we see that we start getting quite good results but also that we get more sensitive to overtraining as we add more features.

Because in this linear regression scheme, every feature has an associated weight, one may examine the weights to see how influential each feature is. This step may be regarded as a sanity check of the system, and a way to gain further understanding of the problem nature. Firstly, it can be noted that the parameters are expected to be positive or negative depending on how the features are engineered.

ϕ_4	negative
ϕ_5	unclear
ϕ_6	positive

Table 12: Because the feature ϕ_4 is the proportion of subsamples with variance below a threshold, a high output of ϕ_4 should correspond to a low estimate of CDRS. Hence, this parameter should be negative. The feature ϕ_5 is harder to interpret, and a high value is unclear to indicate a higher or lower CDRS. The feature ϕ_6 is a measure of how much the person is moving, with a positive value being expected.

w_1	corresponds to $\phi_4(\cdot, 0.1, 200, 100)$	-0.38
w_2	corresponds to $\phi_5(\cdot, 200, 100)$	-0.27
w_3	corresponds to $\phi_6(\cdot, 200, 100)$	0.61

Table 13: Average of the model parameters from the same model as in Table 11, trained with N=500 different training/validation splits. The parameters are what is to be expected.

Finally the following results were found using only a subset of 2 features.

	training MSE	validation MSE	ρ training	ρ validation	MAE	Max AE
mean	0.215	0.265	0.853	0.848	0.409	1.33
std	0.021	0.06	0.027	0.067	0.049	0.197

Table 14: Evaluation metrics for linear regression when using features $\phi_4(\cdot, 1, 200, 100)$ and $\phi_4(\cdot, 0.3, 200, 100)$ applied to θ alone. Although the validation MSE is about the same, the max absolute error is greatly reduced.

w_1	corresponds to $\phi_4(\cdot, 1, 200, 100)$	-0.69
w_2	corresponds to $\phi_4(\cdot, 0.3, 200, 100)$	-0.10

Table 15: The observations are in line with what is to be expected: Parameters corresponding to ϕ_4 are negative. We can now conclude with more confidence that the features are suitable for further use.

For further reference of the most promising sets of features above, the following feature maps below are introduced.

	ϕ_4	ϕ_5	ϕ_6
T	0.1	-	-
L	200	200	200
S	100	100	100

Table 16: Feature map 1.

	ϕ_4	ϕ_4	ϕ_4	ϕ_5	ϕ_5	ϕ_5	ϕ_6	ϕ_6	ϕ_6
T	0.1	0.1	3	-	-	-	-	-	-
L	100	200	400	100	200	400	100	200	400
S	50	100	200	50	100	200	50	100	200

Table 17: Feature map 2.

	ϕ_3	ϕ_3	ϕ_3
n_C	10	20	30

Table 18: Feature map 3.

	ϕ_4	ϕ_4
T	1	0.3
L	200	200
S	100	100

Table 19: Feature map 4.

5.4.5 Features that did not work well

As previously stressed, a lot of the work has consisted of investigating different features of choice. In the sections above, only a small selection of the features that did not work out are shown, while all that are used in the final model are presented. For example, there are a lot of frequency domain features that could potentially be of value. While considering those, and engineering features around for example the estimate of the power spectral density, it became clear that these exhibit no evident pattern. While looking at the recordings of the patients, this is the conclusion as well: The movement we are looking for is quite random, and as far as could be concluded, there could be no distinct frequency observed.

5.5 Choosing a model architecture

Armed with understanding of what features are descriptive to address the problem, it is time *t*o choose model architecture. From Figure 5, it can be seen that approximately half of the labels are 0 while the other half is higher than 0. This makes up a balanced data set for a binary classification problem that can be utilized. The proposed inference pipeline is therefore:

1. Make binary prediction, whether the sample contains hyperkinesia at all or not.

2. Estimate CDRS score.

The idea is also that an ensemble of the two models above can help build confidence in the system. If the models disagree, the result should perhaps not be trusted. If they disagree a lot of the time, a doctor can easily reject the measurements. Presented with just an estimation of CDRS, it is harder to make any conclusion of the quality of the measurement.

A natural first approach is to use a support vector classification, SVC, scheme for the binary classification and its regression sibling, support vector regression, to make a prediction of the CDRS score. Support vector machines are suitable here because they provide easily tuneable hyperparameters to counter overtraining. Next, the main concepts for the variations of support vector machines will be covered.

5.5.1 Kernel support vector machines

The support vector machines in this work are ones that use a kernel to implicitly define a mapping of the self engineered input features to a higher dimensional feature space. The point is that the primal objective has a dual form with a convex objective function, from which it is possible to extract an optimal solution to the problem in its primal form. This enables us to define a kernel in its dual formulation, which corresponds to a feature mapping in the primal form. This feature mapping, which in this report is denoted Φ, is usually not retrievable, and we have only an implicit notion of it, to connect the primal to the dual form.

The kernel used in this report is a Gaussian one,

$$K(x, y) = e^{-\frac{||x-y||_2^2}{2\sigma^2}}, \tag{26}$$

which is widely used. Further details on kernel support vector machines can be found in the work of Schollköpf et.al. [18].

5.5.2 Support vector machine for binary classification

The objective for an SVC in its primal form is given by

$$\begin{gathered} \min_{\mathbf{w},\zeta} ||\mathbf{w}||_2^2 + C\sum_{i=1}^{N} \zeta_i, \\ \text{subject to } y_i(\mathbf{w}^T\Phi(\mathbf{x}_i) + b) \geq 1 - \zeta_i \\ \zeta_i \geq \ 0 \ \forall \ i \ \in \{1, ..., N\} \end{gathered} \tag{27}$$

The goal is to find a separating hyperplane in the feature space defined by the kernel, and maximise the margin to any data points. Ideally, $y_i(\mathbf{w}^T\Phi(\mathbf{x}_i)+b) \geq 1$ for all samples, which would imply full separability, but since this most times is not possible (at least while keeping $||\mathbf{w}||_2^2$ reasonably small), we allow for points to be at a distance ζ_i on the wrong side of the plane, while we at the same time minimise the total distance of data points from their corresponding ”right” side of the hyper plane. This optimisation is solved in its Fenchel-dual form, and given a solution, we are able to extract a model that solves the problem in its primal form. This optimisation scheme and recovery of a model from the dual form is left out.

The here used hyper parameter C is such that a large C imposes a greater loss for data points on the wrong side of the hyper plane. This means that we fit better to the training data, at the cost of getting a large $||\mathbf{w}||_2^2$. A large C, can therefore affect regularization properties of the model, and is therefore important to investigate.

Results on validation data; Fixed hyperparameters

Here, the most important part-results are showcased. First, the dependence of the feature selection is showcased for a fixed C=2. Second, dependence of C is displayed for a few examples. All Tables presented in this section is a representation of the distribution retrieved when training the model on N=500 training/ validation splits.

	training accuracy	val accuracy	training precision	val precision	training recall	val recall	val f1
mean	0.953	0.934	0.978	0.946	0.85	0.811	0.872
std	0.018	0.014	0.031	0.031	0.055	0.052	0.032

Table 20: ”Val” is short for validation. Results for SVC using feature map 1, applied to θ alone, with $C = 2$.

	training accuracy	val accuracy	training precision	val precision	training recall	val recall	val f1
mean	0.955	0.932	0.979	0.948	0.852	0.802	0.868
std	0.018	0.009	0.03	0.039	0.054	0.025	0.016

Table 21: Results for SVC using feature map 1, applied to both θ and φ. According to these results, φ is better to leave out.

	training accuracy	val accuracy	training precision	val precision	training recall	val recall	val f1
mean	0.955	0.936	0.982	0.946	0.854	0.819	0.877
std	0.02	0.014	0.024	0.044	0.07	0.034	0.026

Table 22: Results for SVC using feature map 4 applied to θ and feature map 3 applied to linear acceleration. We observe small improvements in the validation metrics.

Evaluating hyperparameters
As can be seen in figures 7 and 8, there is not a large dependence upon C for $C \geq 1$.

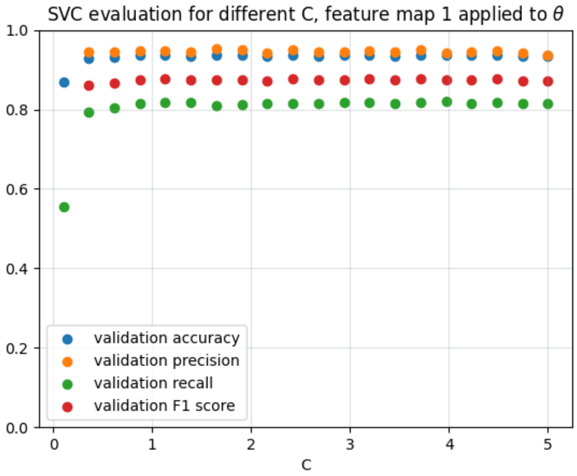

Figure 7: Dependence on C for feature map 1 applied to θ alone.

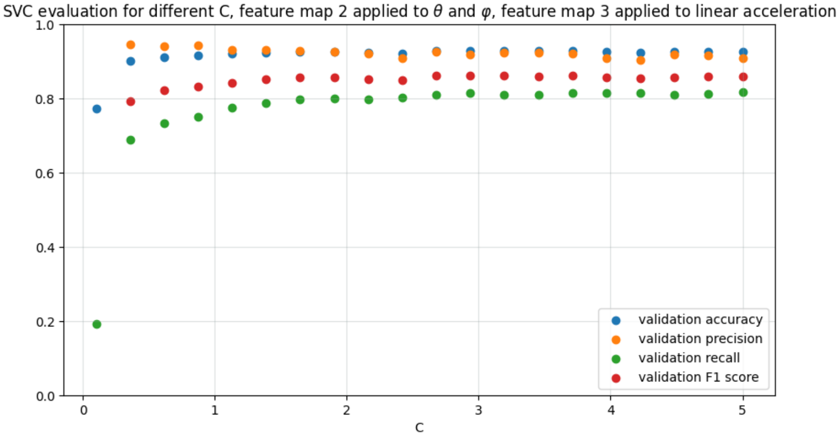

Figure 8: Dependence on C for feature map 2 applied to θ and φ and feature map 3 applied to linear acceleration.

5.5.3 Support vector regression, SVR

A widely used support vector machine used for regression is what is called epsilon-SVR, or ϵ-SVR. Epsilon is here interpreted as the margin which the predictions can be within without contributing to the loss. All the samples that lie outside a distance of ϵ from the target value penalizes the objective function by a value of ζ_i for samples above and ζ_i^* for samples below the target interval.

The primal problem is formed as in equation (28), and Φ is again the feature mapping that is implicitly defined by the kernel.

$$\begin{aligned} &\min_{\mathbf{w},\zeta} ||\mathbf{w}||_2^2 + C\sum_{i=1}^{N}(\zeta_i + \zeta_i^*), \\ \text{subject to } &y_i - \mathbf{w}^T\Phi(\mathbf{x}_i) - b \leq \epsilon - \zeta_i \\ &\mathbf{w}^T\Phi(\mathbf{x}_i) + b - y_i \leq \epsilon - \zeta_i^*, \\ &\zeta_i, \zeta_i^* \geq \ 0 \ \forall \ i \ \in \{1, ..., N\} \end{aligned} \tag{28}$$

Here, there is yet another hyper parameter to determine, namely ϵ. Choosing this, one would have to consider the problem nature and what prediction error is accepTable. But it is also possible that a tighter or looser ϵ than actually needed might generate better results.

Results on validation data; Fixed hyperparameters

	ρ	ρ nonzero	MSE	MSE monzero	MAE	MAE nonzero	worst
mean	0.837	0.566	0.26	0.533	0.316	0.599	1.686
std	0.08	0.228	0.079	0.169	0.057	0.107	0.313

Table 23: Evaluation metrics, feature map 1 applied to θ, for a set of N=500 training/ validation data splits.

	ρ	ρ nonzero	MSE	MSE monzero	MAE	MAE nonzero	worst
mean	0.86	0.645	0.227	0.465	0.305	0.583	1.47
std	0.07	0.17	0.056	0.11	0.047	0.075	0.265

Table 24: Evaluation metrics, feature map 4 applied to θ, for a set of N=500 training/ validation data splits.

Evaluation of hyperparameters

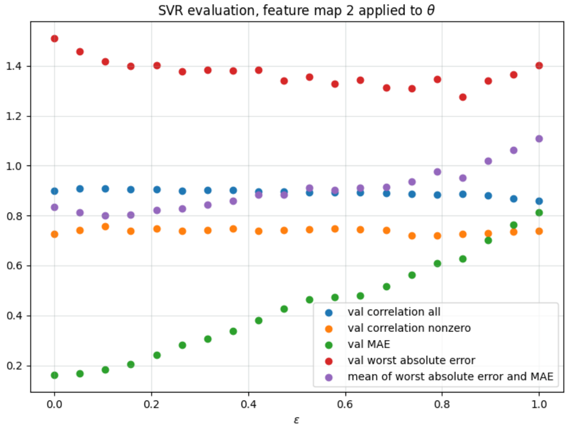

Figure 9: Feature map 2 applied to θ for C = 2 and different $\epsilon \in [0,1]$*. We see that the mean absolute error is heavily dependent on the choice of* ϵ*. Further, it can be seen, that while optimising for a low mean between worst case absolute error and MAE, an* ϵ *of 0.1 is optimal.*

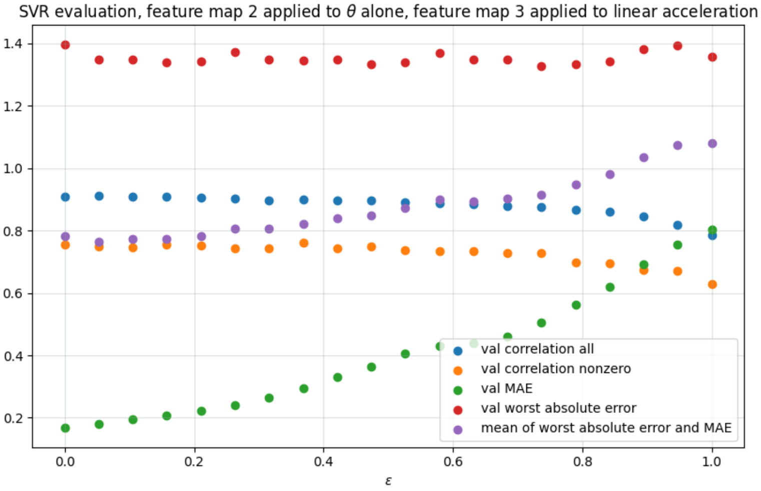

Figure 10: Feature map 2 applied to θ alone, while feature map 3 is applied to linear acceleration in their three dimensions, for C = 2 and different $\epsilon \in [0,1]$*.*

As seen in figures 9 and 10, the correlation with the ground truth is close to 0.9 regardless of the choice of ϵ. Because the mean absolute error is the lowest with a small value of ϵ, everything else same, this is a natural choice for the hyperparameter.

5.5.4 Final models

The final models used for binary classification and regression, backed up with the experiments described in this section, and with a more complete picture with the Tables and pictures in appendix A, are the following:

Binary classification
The model with best f1-score is the one using feature map 2 applied to θ alone, with C = 2.

Regression on CDRS
Although the best overall performance was reached while using feature map 4 on θ (leaving out φ) along with feature map 3 on the linear acceleration, with $\epsilon = 0.05$ and $C = 2$, it might be a good idea to value the simplicity of the model quite high. The model using feature map 4 on θ is therefore also highly interesting, as it is a significantly simpler model. Both these models will be tested on the testing data set.

- Proposal 1: Feature map 2 on θ and feature map 3 on linear acceleration. $\epsilon = 0.05$ and $C = 2$.
- Proposal 2: Feature map 4 on θ. $\epsilon = 0.05$ and $C = 2$.

5.6 Code remarks

The code used for all the work in this report is written in python with a set of packages, the most prominent being *Sklearn* [13]. The code is available on github: `https://github.com/Cryhouse/hyperkinesia`

6 Results and discussion

In this section, the results of the models on the unseen testing data is presented, as well as some discussion around them.

6.1 Binary predictor: Support vector classification

Because the level 2 predictor is the most important part of the results, it will be presented first, followed by the level 1 predictor. Recall that the level 2 predictor is the one that uses the entire signal, whereas the level 1 predictor is only fed data that is truncated to when the person is sitting. As this section is only evaluating model performance, training loss metrics are left out.

6.1.1 Level 2 predictor

accuracy	precision	recall	f1 score
0.79	0.97	0.58	0.73

Table 25: Results for proposed model on the testing data set of 6 patients with a total of 123 samples.

The results are to be compared with those in Table 22, where an accuracy of 0.932, precision of 0.946, recall of 0.802 and F1-score of 0.877 were reached for the validation data. The only metric that improved over experiments is the precision, with only one false positive in the entire testing data set.

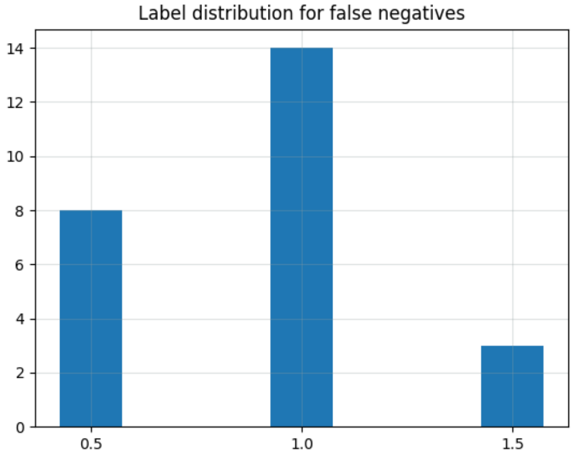

Figure 11: Distribution of the samples that were falsely predicted as not containing any hyperkinesia on the original testing data set.

Because these results are dependent on how the testing data is chosen, a set of tests were also conducted, to see how much the choice of testing data can affect the results. To test this sensitivity, the same method as when splitting up the train data to a training/ validation data was applied to the entire data set. When doing the above described resampling of training/ testing data, a number of N=500 times, the following results were achieved:

	accuracy	precision	recall	f1_score
mean	0.873	0.886	0.814	0.841
std	0.059	0.069	0.123	0.075

Table 26: Binary classification results with resampling of training/ testing data with stratification.

It can be seen in Table 26 that all the performance metrics go up when resampling the training- and testing patients. This hints that the original testing data is not representative of the original training/ validation data.

6.1.2 Level 1 problem

In the same way, and for the same reasons as discussed earlier, the results for this model will be presented in three parts where it is evaluated on the original testing data, randomly sampled testing data, and stratified sampled testing data.

accuracy	precision	recall	f1_score
0.854	0.889	0.800	0.842

Table 27: Results on the original testing data.

	accuracy	precision	recall	f1_score
mean	0.897	0.921	0.827	0.867
std	0.045	0.065	0.103	0.070

Table 28: Results for level 1 binary predictor for a set of 100 training/testing splits selected at random.

	accuracy	precision	recall	f1_score
mean	0.894	0.913	0.828	0.864
std	0.044	0.070	0.104	0.070

Table 29: Results for level 1 binary predictor for a set of 100 training/testing splits selected with stratification.

Only a small improvement is observed. Although the truncated data is likely less noisy, there is also less data to make the inference on. It is therefore reasonable that the performance is not increasing much.

6.2 Regression problem: ϵ-SVR

6.2.1 Level 2 problem

ρ	ρ non zero	MSE	MSE nonzero	MAE	MAE nonzero	worst absolute error
0.80	0.66	0.20	0.35	0.32	0.51	1.18

Table 30: Results for proposal 1 on the testing data set. When looking at these results, one should consider Table 4, where the correlation between the two doctors' labels is presented. The correlation between the two doctors' labels is 0.686 for all samples and 0.605 those labeled as nonzero. These results are therefore in line with what could be expected of an assessment by a third doctor.

ρ	ρ non zero	MSE	MSE nonzero	MAE	MAE nonzero	worst absolute error
0.79	0.70	0.25	0.52	0.31	0.61	1.47

Table 31: Results for proposal 2 on the testing data set. Slight improvement in nonzero correlation, but worse performance on the other metrics than proposal 1.

	ρ	ρ nonzero	MSE	MSE monzero	MAE	MAE nonzero	worst
mean	0.840	0.615	0.204	0.375	0.318	0.510	1.415
std	0.060	0.152	0.065	0.131	0.058	0.098	0.305

Table 33: Results for proposal 1, resampled testing data.

	ρ	ρ nonzero	MSE	MSE monzero	MAE	MAE nonzero	worst
mean	0.844	0.669	0.218	0.43	0.301	0.555	1.448
std	0.068	0.15	0.056	0.088	0.053	0.065	0.221

Table 34: Results for proposal 2, resampled testing data.

It is also interesting to see the correlation of the model with both of the individual doctors' assessments.

ρ expert A	ρ expert B
0.57	0.87

Table 32: Correlation of model output with individual doctors' assessments, on the testing data set. Here, we see very strong correlation with one of the doctors' assessments, while the other is weaker. Still, both these numbers are in the same ballpark as the doctors' correlation with each other, as presented in Table 4.

Because the results here are also dependent on how the test set is chosen, the same method of resampling testing data was applied to this problem.

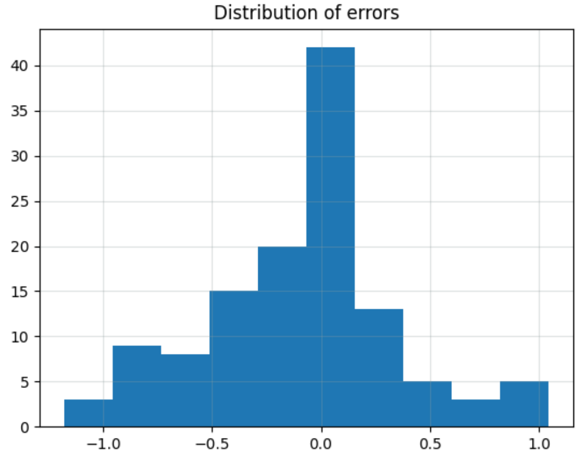

Figure 12: CDRS prediction error distribution for proposal 1 on the original testing data set.

6.2.2 Level 1 problem

Here, results for regression on data truncated to when the patients are sitting down, is presented. The same set of features, i.e., feature map 2 on θ and feature map 3 on linear acceleration are used, and the same model architecture is used as in the level 2 problem.

ρ	ρ nonzero	MSE	MSE monzero	MAE	MAE nonzero	worst
0.827	0.552	0.188	0.278	0.296	0.443	1.430

Table 35: Results for proposal 1 on original testing data. The results are similar to the level 2 predictor.

ρ	ρ nonzero	MSE	MSE monzero	MAE	MAE nonzero	worst
0.784	0.623	0.227	0.448	0.290	0.515	1.463

Table 36: Results for proposal 2 on original testing data.

Because the features and model was optimised for the level 2 problem, it is highly likely that the level 1 model would perform better with a specialized set of features. There is room for a lot less noise in the signal if the person wearing the sensor is sitting down. This is also backed by the fact that the more complex model (proposal 1) is performing better. Now to checking the results on resampled testing data.

	ρ	ρ nonzero	MSE	MSE monzero	MAE	MAE nonzero	worst
mean	0.837	0.594	0.217	0.415	0.301	0.527	1.506
std	0.076	0.173	0.081	0.152	0.067	0.109	0.194

Table 37: Results for proposal 1 with resampled testing data.

	ρ	ρ nonzero	MSE	MSE monzero	MAE	MAE nonzero	worst
mean	0.810	0.609	0.243	0.526	0.313	0.602	1.720
std	0.081	0.163	0.055	0.126	0.046	0.086	0.270

Table 38: Results for proposal 2 with resampled testing data.

The level 2 predictor seems to perform better than the level 1 predictor in this case.

6.3 Comparison to previous work

Because the models made by Liljeroth were set up as classification models and not regression models, there is no obvious way of comparing the results. Further, because the resolution of the annotations is double when taking the average of two doctors' labels, the corresponding classification problem would have twice as many classes, and not be opt for comparison. To get around this, two models were trained, one for each of the doctors' labels and rounded the result to get a surrogate for a classification. Further, in [9], each sample was split up to a set of subsamples where each label had the "global" label. In order for the comparison to be opt, the confusion matrices presented below are normalized so that the sum of all elements is 100. For conciseness, only the best model is presented, which is proposal 2, level 2. The true label is corresponding to the *i*:th row and the prediction is corresponding to the j:th column.

61	7	1	0
5	7	2	0
1	4	7	1
0	0	3	0

Table 39: Level 2, proposal 2 for model trained on expert A's annotations. Accuracy was 76%.

56	2	1	0
8	4	5	0
2	3	12	1
0	0	3	3

Table 40: Level 2, proposal 2 for model trained on expert B's annotations. Accuracy was 75%.

0	23	1	0
0	24	11	2
0	4	14	5
0	0	9	6

Table 41: Confusion matrix presented in previous work [9]. Accuracy was 44%.

It seems like the models proposed in this thesis are competitive with previous models. It should however be stressed that the prior models might still be competitive, but as the code for training the same system was not available at the time of this report, the prior models could not be retrained with the new data.

7 Future work

There is a lot of potential for models to become better. This includes both collecting more data, but also improving the models. First, the data aspect of the problem can formulated as follows:

- Given the data available in this study, we have no way of knowing whether the samples are representative for a day in real life for the participants. Wearing the sensor throughout a day might not be a representative mix of activities as the case in this study. Ergo, we have no way of knowing if the models proposed in this thesis are as good as when applied to the real world. The solution is to collect more data over larger time spans where the participants are going about their daily businesses.
- A key part to continuing this work is to source more data, preferably with labels. To source more data, one might have an app that both collects data, and also prompts the person to do self assessments and perhaps a set of tasks. Although the self assessment is not always reliable, it would be a means of collecting a lot more data to validate the models.

Further, the following ways to make the models themselves better should be proposed:

- It is possible that more data is fit for use in the model. For example, age, sex or BMI. Because this data is available, there is no reason for it not to be tried.
- Make individual models for each patient.

Further, it should be stressed that the approach in this thesis is chosen with high regard to the small number of available samples. Self-engineered features is essentially a way of applying biases to the problem at hand. Without much data however, biases are needed to be able to get any predicting power. When getting more data, the strive should be to eliminate as much as possible of the biases, and let the data explain itself.

8 Conclusion

The aim of this thesis has been to show that it is possible to utilize mobile sensors to collect valuable feedback from people who suffers from PD, w.r.t. hyperkinesia. This has been investigated mainly on an entire data sequence, which contains a set of 4 different tasks as well as noise that would be present in the real world. When considering the results by measures of Pearson correlation and mean absolute error, it is similar to what could be expected of an assessment by a third doctor. The following can therefore be concluded:

- It is possible to collect valuable feedback with smartphones, and the models proposed in this thesis are hands-on examples of this.
- To gain confidence with the models and develop them further, more work is needed. It includes both collecting more data as well putting in more work on the models themselves to retrieve a pipeline that can be trusted.

9 Acknowledgements

There are a number of people I would like to thank.

First, I would like to direct a thank you to Erik Liljeroth, who provided a lot of good background information and context, as well as sharing practical know-how. This help has been invaluable. Second, I would like to thank Dr. Pontus Giselsson, whom I have had lengthy discussions with about how to go about crafting features as well as choosing model architecture. Further, the team at the department of neurology at Lund University for their tedious work in collecting all the data that made any of this work possible.

Further, thank you Elsa Blomskog, Andreas Hansson and Theodor Bucht for proof reading and encouragement in the process of writing this report!

I would also like to direct a big thank you to Dr. Sotirios Grigoriou at the dept. of neurology here in Lund for his patience in explaining the medical concepts to me, as well as his cheerfulness and celebration of the part-results I presented to him along the way.

Last but not least, I would like to direct a wholeheartedly thanks to my supervisor, Prof. Andreas Jakobsson, who has been invaluable in many aspects of this work, but perhaps foremost in pushing me beyond the barriers of comfort that otherwise would have hampered me.

References

[1] Grigoriou, S. Personal Communication

[2] Hjärnfonden (2022), *Parkinsons sjukdom*. Available at: https://www.hjarnfonden.se/om-hjarnan/diagnoser/parkinsons-sjukdom/(Accessed: 21-11-2022)

[3] Nobel prize committee (2022), *The Nobel Prize in Physiology or Medicine*. Available at https://www.nobelprize.org/prizes/medicine/2000/summary/ (Accessed: 21-11-2022)

[4] Daneault, JF et al. (2021), *Accelerometer data collected with a minimum set of wearable sensors from subjects with Parkinson's disease*. Sci Data 8, 48. doi: https://doi.org/10.1038/s41597-021-00830-0

[5] Zhang, A et al. (2018), *Automated Tremor Detection in Parkinson's Disease Using Accelerometer Signals*, IEEE/ACM International Conference on Connected Health: Applications, Systems and Engineering Technologies (CHASE), 2018, pp. 13-14, doi: https://doi.org/10.1145/3278576.3278582

[6] V. Sharma et al. (2014), *SPARK: Personalized Parkinson Disease Interventions through Synergy between a Smartphone and a Smartwatch*. DUXU 2014. Lecture Notes in Computer Science, vol 8519. Springer, Cham. doi: https://doi.org/10.1007/978-3-319-07635-5_11

[7] Xiaochen Z. et al. (2017), *Continuous Monitoring of Essential Tremor Using a Portable System Based on Smartwatch*,, Frontiers in Neurology, col 8, doi: https://doi.org/10.3389/fneur.2017.00096

[8] Contreras, R. et al. (2016), *Tremors quantification in parkinson patients using smartwatches*, IEEE Ecuador Technical Chapters Meeting (ETCM), pp. 1-6, doi: https://doi.org/10.1109/ETCM.2016.7750866.

[9] Liljeroth, E (2020), *Feature Extraction and Classification of Medication-Induced Hyperkinesia During Treatment of Parkinson's Disease*, Lund University

[10] Schulz, MA. et al. (2020), *Different scaling of linear models and deep learning in UKBiobank brain images versus machine-learning datasets*. Nat Commun 11, 4238). doi: https://doi.org/10.1038/s41467-020-18037-z

[11] Fuchs, C. et al., (2021), *Tremor assessment using smartphone sensor data and fuzzy reasoning*, BMC Bioinformatics 2021, 22(Suppl 2):57, doi: https://doi.org/10.1186/s12859-021-03961-8

[12] Lever, J., Krzywinski, M. and Altman, N. (2016) *Points of Significance: Model selection and overfitting*, Nature Methods, 13(9), 703+, doi: https://doi.org/10.1038/nmeth.3968

[13] Pedregosa, F. et al. (2011), *Scikit-learn: Machine learning in Python*. the J. of M.L. research, 12, 2825-2830.

[14] LeCun, Y. et al. (1998), *Gradient-based learning applied to document recognition*, Proceedings of the IEEE, 86, 2278–2324. doi: `https://doi.org/10.1109/5.726791`

[15] Apple, inc (2022), *Core Motion*, `https://developer.apple.com/documentation/coremotion`, accessed 2022-12-11

[16] Digital Neurosurgeon (2012), *Tremor 12*, `https://apps.apple.com/us/app/tremor12/id1005509327`, accessed 2022-12-11

[17] Nitesh V. et al. (2022), SMOTE: synthetic minority over-sampling technique.. J. Artif. Int. Res. 16, 1, 321–357. doi: `https://doi.org/10.48550/arXiv.1106.1813`

[18] Cristianini, N., Schölkopf, B. (2002). *Support Vector Machines and Kernel Methods*, AI Magazine Volume 23 Number 3, doi: `https://doi.org/10.1609/aimag.v23i3.1655`