TALLINN UNIVERSITY OF TECHNOLOGY
Faculty of Information Technology
Department of Software Science

Elisa Tamm 179465IAIB

ASSOCIATIONS BETWEEN FINE MOTOR TESTS AMONG PARKINSON'S DISEASE PATIENTS AND CONTROL GROUPS

Bachelor Thesis

Supervisor
Sven Nõmm
PhD
Co-supervisor
Aaro Toomela
PhD

Tallinn 2020

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond
Tarkvarateaduse instituut

Elisa Tamm 179465IAIB

PEENMOTOORSETE TESTIDE VAHELISTE SEOSTE LEIDMINE PARKINSONI TÕVEGA PATSIENTIDE JA KONTROLLGRUPPIDE SEAS

Bakalaureusetöö

Juhendaja
Sven Nõmm
PhD
Kaasjuhendaja
Aaro Toomela
PhD

Tallinn 2020

Author's declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Elisa Tamm

Date: 18.05.2020

Annotatsioon

Peenmotoorsed testid, mille hulgast võib leida Luria vahelduvate seeriate testi, on neuroloogiliste tõvede diagnoosimiseks kasutusel mitmeid kümnendeid. Traditsiooniline testide läbiviimine paberkandja ja pliiatsiga on tehnoloogia arengu tulemusel viiakse läbi palju uurimusi, et asendada need tahvelarvutite ja digitaalsete pliiatsitega. Selline arengusuund võimaldab arvutada lisaparameetreid, milleks antud töös on kinemaatilised ja surve parameetrid. Nende abil on võimalik andmete analüüs masinõppe mudelite alusel.

Sarnased uurimustööd on antud võimalusi kasutades viinud läbi mitmeid uuringuid, kuid ei ole fokusseeritud testide patarei koostamisele. Testide patarei uurimiseks võrreldi antud töös peenmotoorsete testide patareis olevaid seoseid nende parameetrite alusel. Lisaks treeniti lähtuvalt kogutud andmetest erinevaid masinõppe mudeleid, et teostada peenmotoorseid teste teinud inimese testitulemuste analüüs üksikisiku tasandil.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 41 leheküljel, 6 peatükki, 10 joonist ja 9 tabelit.

Abstract

Fine motor tests, such as Luria's alternating series tests, have been in use for the diagnostics of neurodegenerative disorders, like Parkinson's disease, for several decades. Traditionally conducted by pen and paper, the evolution of electronics has introduced the digitisation of these tests. Using tablet computer with digital pencil gives the opportunity to analyze additional kinematic and pressure parameters that enlarge the scale of application of machine learning models.

Even though researches have been conducted to apply these methods, no focus on the possible optimisation of the test battery has been studied. In this thesis, the relations between parameters describing the fine motor test battery is investigated along with the analysis of individual subject who has carried out fine motor test belonging to the test battery examined in the thesis.

The thesis is in English and contains 41 pages of text, 6 chapters, 10 figures and 9 tables.

List of abbreviations and terms

AST	Alternating Series Test
DT	Decision Tree
HC	Healthy control
JSON	JavaScript Object Notation
KNN	K-Nearest Neighbours
LR	Logistic Regression
MaxEnt	Maximum-entropy
PD	Parkinson's Disease
RF	Random Forest
SVM	Support Vector Machine

Table of Contents

List of Figures

List of Tables

1. Introduction

In the course of recent years, the advancements of data digitisation have led to multiple new possibilities in the assessment of cognitive and motor functions. Replacing the traditional setting of paper and pencil with touch screen tablet technology offers the opportunity to acquire additional precise information, allowing the computation of kinematic and pressure parameters.

One of the world's most spread neurodegenerative disorders Parkinson's disease (PD) affects fine motor skills, mainly initiation and execution of voluntary movements [1]. While applications containing digitised versions of tests used to diagnose PD based on these symptoms have been developed, the analysis, if conducted, has been done by specific test, alternating between researches [2, 3].

The aim of this thesis is to use attained parameters to train machine learning models and apply them in the analysis of each test on an individual result. For that purpose, an assisting application is developed, which provides extraction of kinematic and pressure parameters from raw data provided by files in JavaScript Object Notation (JSON) format and analyzes received data.

Such method relieves doctors and medical workers from having to subjectively assess the probability of PD and compare individual subject data to the metrics of both PD diagnosed patients and healthy controls (HC). To differentiate these two groups, it is relevant to draw boundaries between them.

Along with the problem of growth in parameters, the high number of conducted tests raises serious problems that when possible, could be reduced through optimisation. When subjects, who possibly suffer from PD have to endure filling out several tests, which in essence correlate with one another, that time and energy could be conserved, without compromising the integrity and accuracy of results.

Furthermore, the thesis aims to compare machine learning models trained by test types to the models trained by test battery. This type of assessment, to the knowledge of the author has not been previously conducted.

2. Background and problem statement

2.1 Parkinson's disease

Most commonly diagnosed among the age of 60 and above and independently among males, PD is one of the world's most spread neurodegenerative disorders, second only to Alzheimer's disease. The incidence of PD in average is around 0.1%. As it raises according to age, for the population over the age of 65, incidence reaches 1-2% [1].

As the causes of PD remain largely unknown, the diagnosis and detection relies solely on the occurring symptoms of a subject. PD most prominent symptoms that can be grouped by an acronym TRAP, include: Tremor at rest, Rigidity, Akinesia (or bradykinesia) i.e poverty of movement and Posture disturbance [4]. All of which have impacts on subject's fine motor skills.

Non-motor symptoms, such as emotional and behavioral, occur only in 40% of subjects and in the event of PD aggravation, one of five subjects will have diagnosis of dementia [1]. Without treatment, PD symptoms have mayor progressive deterioration, gait freezing and subjects experience frequent falls [1]. Thus the early diagnosis and treatment of PD is of utmost relevance.

Contrary to most of the neurodegenerative disorders, through right treatments, mitigation of symptoms is possible [5]. Several medications and therapies can improve subject's daily activities and in association subjects often reach near-normal life expectancy. In cases of early disease discovery, the efficiency of these remedies prove to be higher [1].

Although diagnosing PD in it's classical presentation is straightforward, separating it from other forms of parkinsonism proves to be difficult, as the early stages of the disease show symptoms that overlap with other syndromes. Assessment of patients is done by clinical criteria and without the existence of a definitive test. The clinical criteria is a combination of cardinal motor features used to determine PD. Possible diagnosis is defined by at least two of the four symptoms characteristic to PD and probable diagnosis requires at least three features. Throughout years, the only definite criterion standard has been pathological conformation on autopsy [4].

The absence of standardised and trustworthy test paired with the notion of medical workers subjective conclusions regarding the clinical criterias used to determine PD, aid to the objective of integrating statistics and machine learning for the purpose of supporting the diagnostics.

2.2 Luria's Alternating Series Test

Alexander Romanovich Luria, often recognised as the father of modern neuropsychological assessment, published a simple clinical battery, containing several techniques which investigated motor functions as well as hand movements. One of those tasks are Luria's Alternating Series Test (AST). It is a graphic task where subject is asked to draw a pattern consisting of alternating shapes, most commonly squares and triangles. It's universality and simplicity makes it a popular procedure among researchers and neurologists. Furthermore, as it engages cognitive processing, motor activity and inhibitory control, it may be used at various stages of diagnostics of many neurodegenerative diseases [6].

Alternating series test is used to assess the state of planning and execution of fine motor motions. Initially, three kinds of series were advised to detect instabilities in the comprehension, planning and implementation of tasks followed by the subjects. As completing an AST requires constant effort, it engages both physical side, needing the stable work of muscles and hand motions and mental side, involving thought processes, which for healthy individuals may seem regular as opposed to subjects with neurodegenerative disorders affecting also fine motor skills. Continuing the pattern of the aforementioned test targets the assessment of the state of planning function, tracing the pattern is used to assess the state of implementation function and ability to switch task can be assessed by asking the patient to copy a sequence of shapes [7].

As Luria's AST has shown to be of valuable use, it has since then evolved by being digitised. The digitisation, as has been demonstrated in various researches, has allowed the distinction between groups of PD patients and healthy controls (HC) by using some of the kinematic and pressure parameters. These parameters are used to perform an analysis using segments of data, calculated features, their selection and classification algorithms [2].

2.3 Data acquisition

Dataset used within the scope of present thesis was acquired from using specially developed digital data recording application for touch screen tablet device and accompanying digital

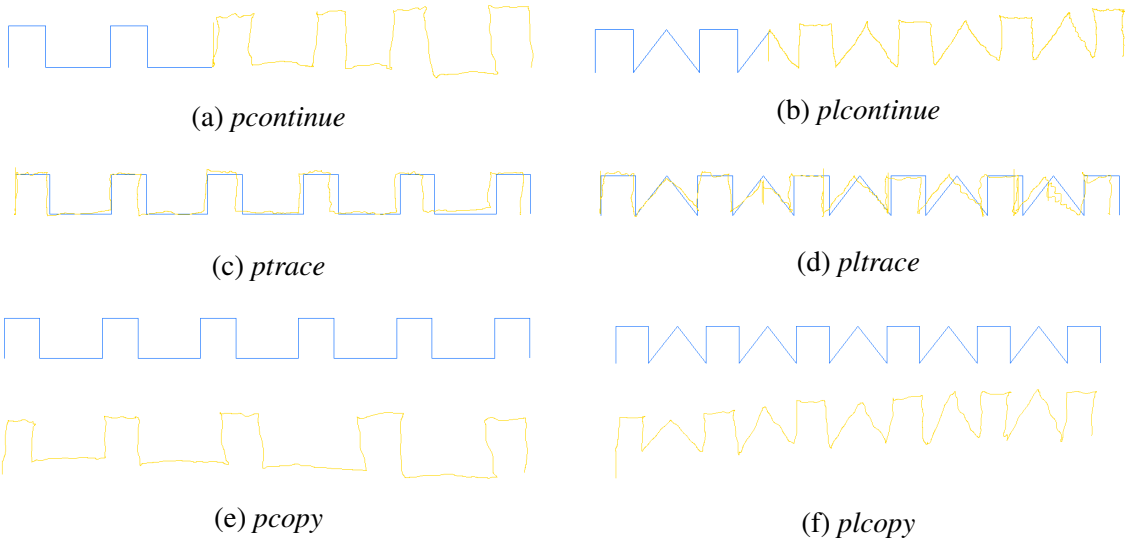
pencil. This application records data about attained parameters with the approximate frequence of 200 points per second [2].

The digital Luria's AST were carried out by providing the subject with touch screen tablet. Subject was seated to better support the hand that was used to grip the digital pencil used to draw on the tablet and to complete chosen tests. Depending on the type of shown test, subject was asked to either continue, trace or copy the pattern shown on screen. On completing first test from the battery, another chosen one was relieved automatically until the completion of all.

Parameters were saved by point and included x and y, indicating the horizontal and vertical axes coordinates where pen was in contact with tablet surface measured in pixels, t denoting timestamp in seconds since Mac OS X epoch i.e the amount of seconds since midnight, January 1, 2001, a and l i.e altitude and longitude, denoting tilt angles between pen and tablet surface and p, the amount of exerted pressure on the pen amidst drawing. Per test taken by a subject, a file containing mentioned data in the JSON format is exported [2]. Parameters used in computations relevant to thesis, four from the set of the parameters: x, y, t and p were accounted for.

All recorded datasets were divided into movements, marking the separation of continuous movement, that means in the time of subject conducting a test, the subject's hand and with that the digital pencil left the surface of the tablet. In the time of such event, no further information was recorded until the returning of pencil connection with the surface.

From the application's possible choices of test battery, the following, as variations of digital Luria's ASTs, were chosen for examination: plcontinue, pltrace, plcopy, pcontinue, ptrace, pcopy. On Figure 1, the line marking the alternating series which were shown on the tablet screen is of the color blue and the line marking the pattern which was drawn by the subject is of yellow color.



(a) *pcontinue* (b) *plcontinue*

(c) *ptrace* (d) *pltrace*

(e) *pcopy* (f) *plcopy*

Figure 1. *Example of digital Luria's alternating series tests in examined test battery*

After extracting the provided data by application developed for the purposes of this thesis, first basic parameters describing kinematics and geometry of the motions were computed. After that, derived data, which describes a set of values being mathematically derived i.e calculated from the set of initial parameters was also computed. In addition the set of integral mass parameters referred to as motion mass is computed. Dataset used in thesis includes in total 52 subjects, containing samples of subjects diagnosed with PD and HC individuals of approximately the same age (mean age of 65) and equal gender proportions.

2.4 Problem statement

The main goals targeted by the thesis are following:

1. Investigate the existence of linear relations between examined features collected in battery
2. Compare classificators trained within the scope of each individual test versus classificators trained in the scope of examined battery
3. Develop a program to choose a trained classification model and use it to analyze on the basis of an individual, including the segmentation between PD and HC groups

To reach these goals, the use of several statistical and algorithmic computations was needed. Inclusively, the calculation of correlations, feature selection and validation of trained classifiers.

3. Method

3.1 Scaling and Normalization

In the raw format, data may have several inconsistencies, missing values or errors. The application of preparation to the management of raw data and it's analysis has been used in such situations. Therefore, when wanting to use data in effective way and draw conclusions that correspond to requirements, a step, called data preparation phase is needed [8].

As the input data with which present thesis worked, was deemed, in examined scope, suitable in most parts, the other forms of data preparations, except for data scaling and normalization, were omitted.

As the name suggests, scaling and normalization refers to the scenario, where different features constitute different scales and may accordingly not be comparable to one another. For example, many machine learning classifiers calculate the distance between two points by the Euclidean distance and in the event of omitting preparations for data, will be directed by feature with broader range of values.

To diffuse this problem, an approach of data standardization was used, as it is a common requirement for machine learning estimators. At the time of the creation of current thesis, two most common ways were looked at: standard scaling and min-max scaler, resulting in the choice of min-max scaler. Although this approach is not effective in the event of extreme value outliers, caused by mistakes in data collection, it was noted that such mistakes had not occurred and as such would be in more understandable format, for the reason that min-max scaling will normalize the majority of values in the range [0,1]. In the other hand, standard scaling results in values that lie in the range [-3,3] [8].

Min-max scaling used to transform features to the default range [0,1], is given by the formula presented below [8]:

$$y_i^j = \frac{x_i^j - min_j}{max_j - min_j}. \tag{3.1}$$

where min_j and max_j represent the minimum and maximum values of attribute j.

3.2 Fisher's score

Fisher score is used to order features by the discriminating power and thus decide the best features for a given class through supervised feature selection. Feature selection is of relevance because of the curse of dimensionality, that occurs during classification when the input data is of high dimensions. The curse of dimensionality refers to the over-fitting of learning methods which leads to less interpretable results.

In this thesis, to overcome these problems, a generalized Fisher score is used. The cause being it's popularity and usage among similar research papers [2]. The main idea of Fisher score is to find a subset of the features represented by two classes and in doing so, separate each data point in such way that the distances between data points in the same class are as minuscule as possible and between different classes as large as possible [9].

The formula for calculating Fisher's score is the following [8]:

$$F = \frac{\sum_{j=1}^{k} P_j(\mu_j - \mu)^2}{\sum_{j}^{k} P_j \sigma_j^2}. \tag{3.2}$$

where k denotes the count of classes, P_j the amount of samples belonging to the class j, μ_j is the mean of of features across attribute j, μ denotes the mean of all classes and σ_j^2 is the variance of the class j.

As the count of classes examined consists of only two: PD and HC, the calculation can simplified to the following state:

$$F = \frac{P_{pd}(\mu_{pd} - \mu)^2 + P_{hc}(\mu_{hc} - \mu)^2}{P_{pd}\sigma_{pd}^2 + P_{hc}\sigma_{hc}^2}. \tag{3.3}$$

3.3 Pearson's correlation coefficient

To execute the measure of association between test battery examined in current thesis, an algorithm of statistical coefficient of correlation was exerted. Between a pair of items, a commonly used statistical measure of correlation is the Pearson coefficient. It can be used to summarize the strength of linear proportions between two data samples. The Pearson coefficient between two variables X and Y is defined by following equation [8]:

$$\rho = \frac{E[X \cdot Y] - E[X] \cdot E[Y]}{\sigma(X) \cdot \sigma(Y)}. \tag{3.4}$$

The coefficient of correlation is defined by range of [-1,1]. Positive correlation is indicated by the maximum value i.e 1 and the value of -1 indicates negative correlation. Values nearing 0 indicate weak or nonexistent correlation between variables [8].

To validate the existence or absence of correlations between examined battery of tests, Pearson coefficient of correlation was computed for Luria's ASTs. The following groups of datasets were compared: ptrace pltrace, pcontinue plcontinue, pcopy plcopy. Results were exported to Excel file and coefficients alongside p-values and graphics were recorded.

Highest recorded positive correlation value reached the amount of 0.43 and negative reached -0.15.

3.4 Classifier models

3.4.1 Decision Tree

Decision Trees (DTs) are a supervised learning method used for classification and regression purposes. DTs are considered to be one of the most efficient classification methods, exceeding expectations over numerous other types. Also, the tree is constantly natural and easily understood, as it forms a tree-like structure comprising of three basic segments: root node, few hidden nodes and terminal nodes, also known as leaves or child nodes [10].

The model created to predict the target value of a target variable is created by following simple guidelines, consisting of a set of if-then-else decision rules. With the raise of steepness, the decision rules get more complex and the model gets more fit [10].

DT used in thesis learned to partition data on the basis of the attribute value. The tree was partitioned in a recursive manner, called recursive partitioning, to handle each node containing best attribute and make that a decision node, which in turn breaks the dataset into smaller subsets that would be as accurate as possible within the data used for training.

3.4.2 Random Forest

Random Forest (RF) classifier or random decision forest, is comprised of DTs. RF creates decision trees on data samples that are selected at random, thereupon RF attains prediction from each tree and the best prediction is selected by getting the vote of each tree, where the class with most in favor will be chosen [10].

It is considered highly accurate and robust method, as it contains a number of DTs participating in the process. Furthermore, another advantage is that the problem of overfitting the dataset will render redundant, as taking the average of all predictions cancels out the biases [10].

3.4.3 K-Nearest Neighbours

K-Nearest Neighbours (KNN) algorithm is a supervised learning algorithm that assumes the proximity of similar data points, by calculating the distance between those points for each example in the data and classifying the sample to the class most frequently occurring amongst the considered amount of k nearest neighbours [11].

Despite that KNN rule for classification is considered as the most simple method among supervised classification approaches, KNN has proven to often outperform more sophisticated methods and has a sound theoretical basis in non-parametric density estimation [11].

3.4.4 Logistic Regression

Logistic Regression (LR) is a linear model for classification. It is also known as logit regression, maximum-entropy classification (MaxEnt) and log-linear classifier. It is often the first go-to method for binary i.e with two class values classification problems as it uses logistic function [12].

LR predicts the probabilities of the default class. In other words, comparing two arbitrary classes A and B, then the first class being A and the logistic regression model could be interpreted as the probability that an input X belongs to the default class A [12].

In that sense, LR could be seen as a model of probability, as opposed to model of classification. It must be noted, that the probability prediction will be transformed into binary values - either 0 or 1.

3.4.5 Support Vector Machine

The Support Vector Machine (SVM) algorithm is in essence an extension to KNN classification approach, also determining neighborhood of data points. However, SVM uses extra steps, by dividing learning data into partitions, using hyper planes i.e lines composed of more than three dimensions, which are created, using vectors of each predictor variable.

For that, hyper plane with biggest classification is used to conclude the nearest training data point of any class [13].

The generalization error of the SVM classifier is dependant on the size of this nearest data point, or functional margin, as it is mainly known [13].

3.5 Model features

From every stroke of the subject made on digital tablet, four parameters: x, y, p and t were used to calculate parameters referred to as derived data. This means features, which can be calculated from the set of initial parameters. It included parameters as difference between points, distance, velocity, acceleration, jerk, to name few, totaling around the size of 25 features. Features were selected based on similar researches conducted, studying fine motor tests being part of Luria's ASTs: [2, 14].

For the next step, aggregate data was also calculated. Additionally to derived features means, medians, maximum, minimum values, standard deviations and so on, a set of variables used to describe movement, known as motion mass parameters were computed. In total, 288 features per subject, test type pair were provided.

The first parameter in motion mass is the actual distance of the drawn trajectory of each interest point, summed. In short, it describes the amount of movements performed. Denoted by L_T and computed by following equation [14]:

$$L_T = \sum_{i=1}^{N} l_i. \tag{3.5}$$

where l_i is the distance between point i and i-1.

Total length calculations are used for several features. L_{Tx} denotes the length calculation for horizontal i.e x axis length. Similarly L_{Ty} is based on vertical i.e the length of y axis. Variable L_{Te} is based on the Euclidean distance ($\sqrt{{l_{x_i}}^2 + {l_{y_i}}^2}$) between data points i.e length.

The sum of absolute velocities computed at the ith observation point is referred to as velocity mass (3.6).

$$V_T = \sum_{i=1}^{N} |v_i|. \tag{3.6}$$

Where v_i is the velocity at the ith observation point.

Acceleration mass describes the smoothness of the motions in the course of taking the test. The lower values indicating smoother motions. To calculate Acceleration mass, the absolute values of all the acceleration values at each data point must be summed (3.7).

$$A_T = \sum_{i=1}^{N} |a_i|. \tag{3.7}$$

Where a_i is the acceleration calculated at the ith observation point.

Jerk mass is defined by equivalent equation (3.8).

$$J_T = \sum_{i=1}^{N} |j_i|. \tag{3.8}$$

Where j_i is the jerk i.e mathematical derivative of velocity at the ith observation point.

Pressure mass follows the changes in applied pressure to the tablet screen in total (3.9).

$$P_T = \sum_{i=1}^{N} |p_i|. \tag{3.9}$$

Where pi is the pressure p calculated at the point i of the interval T.

To conclude, the sum of angle directions between each group of two vectors, also referred to as angular mass, is computed (3.10).

$$D_T = \sum_{i=1}^{N} |d_i|. \tag{3.10}$$

Where d_i is the angle between the directional vector and x axis of the screen at the ith observation point.

Therefore, the parameters can be defined as a tuple of motion mass features:

$$M_T = \{L_T, V_T, A_T, J_T, P_T, D_T\}. \tag{3.11}$$

As like in other conducted researches [15], occasionally additional parameters are added to this tuple. The features explored in this thesis can be seen in Table 9. Additionally, the same features were divided by strokes count and therefore computed for a stroke as well.

3.6 Feature selection

To remove redundant parameters and choose those of real and higher value, and also to avoid the curse of dimensionality the feature selection technique, as was described in the Section 3.2, established feature selection of highest values. The five best features used in thesis to train classifiers can be seen in Table 1.

Table 1. *The best five features by Fisher score*

n	feature	Fisher score
1	angular mass	0.76
2	yaw angle mass	0.67
3	median yaw angle	0.62
4	angular acceleration mass	0.61
5	minimal angle to x axis	0.59

3.7 Classifier validation

When planning to use the parameters that the prediction classifier was trained with to test the predictions, mistakes would arise. A model would simply repeat every decision made and thus be really efficient, unless it encounters new, unseen data. In that case, the model could be rendered useless. This situation is often referred to as overfitting.

To avoid the case of overfitting, the method of splitting the data to classes as test set and train set is used. Test set, as name suggests, is used to test the model and train set in the phase of training the model. As the data sets used in thesis of both PD and HC were rather small in size, k-fold cross-validation, that allows the use of test data for both phases: training and testing, was used [16]. The value of k in thesis was chosen to be 5 in relation

to the size of data samples.

K-fold cross-validation is a procedure, in which the training set is split into number of smaller sets, each containing k folds. For each of the small set, the model is trained using $k-1$ of the folds and then the trained model will be validated with the remaining part of the data that is different for every set. The resulting accuracy will be conducted as the average of accuracy of each iteration [16].

Every trained model, within both method scopes, was subjected to the computation of model accuracy, precision, recall and f1 score, which accordingly mean the fraction of correct predictions in the total number of predictions, the proportion of positive predictions being correct, the proportion of actual positive values being correctly identified and function of precision and recall.

For better understanding of these computations, the following formulas will better accentuate described measures.

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions}. \tag{3.12}$$

Even though outside the machine learning community accuracy is one of the most sought after measure, as the equation suggests, accuracy computation does not delve on the misclassification of false negative and positive values. On contrary, there are other measures which do that:

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} = \frac{True\ Positive}{Total\ Predicted\ Positive}. \tag{3.13}$$

Precision is a good way of measure in the cases when False Positives are detrimental.

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} = \frac{True\ Positive}{Total\ Actual\ Positive}. \tag{3.14}$$

Recall has advantages in the cases where False Negative has severe consequences. For instance, sick patient detection.

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}. \tag{3.15}$$

F1 score is often used as a balance between Precision and Recall and when the class distribution is uneven.

3.8 Classifier training

Due to the small sample of data, the deep learning models were substituted with a classical collection of models, including DT, RF, KNN, LR and SVM, that were trained according to parameters chosen by feature selection, each one of them with a varying number of features.

In current thesis the training of two types of classifiers was made to explore one of the stated problems. The classifiers were trained with data on the basis of every individual test type and as well as on the basis of the entire examined test battery. It was done so to compare the accuracy of predictions in one method versus the other and explore something, that to the knowledge of the author has not yet been done.

First, the dataset was scaled as discussed in Section 3.1. Two arrays were given as input: an array of size [n_samples, n_features] containing training samples obtained as discussed previously and an array of size [n_samples], holding class labels for the training samples. In current thesis class labels were conducted as follows: 0 denoting HC and 1 denoting PD.

The data classes were therefore fitted to the five brought out classifiers, which would therefore predict the class of a given input accordingly. The number of features selected to train the classifiers with highest scores, was found through the computation of all the measures discussed in Section 3.7. As the classification problem has sensitivity to false negative predictions, models were firstly ordered by recall that is in detail discussed in Section 3.7.

Followingly, the evaluation of classification results by test type are presented. The missing values are due to the unreliable results provided being omitted.

Table 2. *Classifier Recall for* ***plcontinue***

Classifier	n of parameters	Recall
DT	-	-
RF	3	0.970
KNN	2	0.802
LR	5	0.682
SVM	5	0.773

Table 3. *Classifier Recall for* ***pcontinue***

Classifier	n of parameters	Recall
DT	-	-
RF	3	0.971
KNN	5	0.727
LR	5	0.592
SVM	5	0.774

Table 4. *Classifier Recall for test* ***plcopy***

Classifier	n of parameters	Recall
DT	2	0.954
RF	2	0.921
KNN	5	0.670
LR	5	0.524
SVM	4	0.670

Table 5. *Classifier Recall for* ***pcopy***

Classifier	n of parameters	Recall
DT	-	-
RF	2	0.983
KNN	4	0.664
LR	4	0.571
SVM	3	0.636

Table 6. *Classifier Recall for test* ***pltrace***

Classifier	n of parameters	Recall
DT	-	-
RF	2	0.970
KNN	3	0.731
LR	4	0.731
SVM	2	0.747

Table 7. *Classifier Recall for* ***ptrace***

Classifier	n of parameters	Recall
DT	2	0.950
RF	2	0.950
KNN	2	0.703
LR	4	0.701
SVM	5	0.712

Evaluations to the extent of data from the battery examined is shown in Table 8.

Table 8. *Classifier Recall for* ***test battery***

Classifier	n of parameters	Recall
DT	2	0.950
RF	2	0.950
KNN	2	0.703
LR	4	0.701
SVM	5	0.712

As it seems, collating the Recall of individual tests to test battery, the values do not differentiate extensively. Additionally, in some cases the values of test battery classifier exceed the ones of an individual test.

3.9 Developed application

As part of the thesis, implementation of developed software was required to use the trained classification models for the prediction of HC or PD and to explore the visual side of those two classes, by demonstrating line charts, two dimensional and three dimensional graphs,

depending on the number of the trained model parameters. It was done with the purpose in mind that present analysis could be extended in the future.

In this section the functionalities analysis, used technologies, tools and user interface is discussed.

3.9.1 Functionalities

It should be noted, that as a point of start, the only certain knowledge was that as an input, the program requires files in JSON format.Reason being, the files that contain subject results are saved in such format.

As such, the application had to:

- Load data from a JSON file
- Allow the selection of trained machine learning models according to test type
- Predict the classification of subject (PD or HC)
- Visualize the position of subject data point in comparison to other PD and HC cases

3.9.2 Implementation

Firstly, the creation of a database was deemed necessary as the trained models and data used to train said models had to be stored. Initially the structure of database was different, but during the course of development it had to be modified several times to suit the additional requirements decided on during the discussions with the supervisor.

For the database implementation, MySQL database was chosen. The reasons being previous experience through the usage in university courses, quick and easy setup and the capability to use phpMyAdmin software to more easily handle the administration of the database.

Current state of database structure:

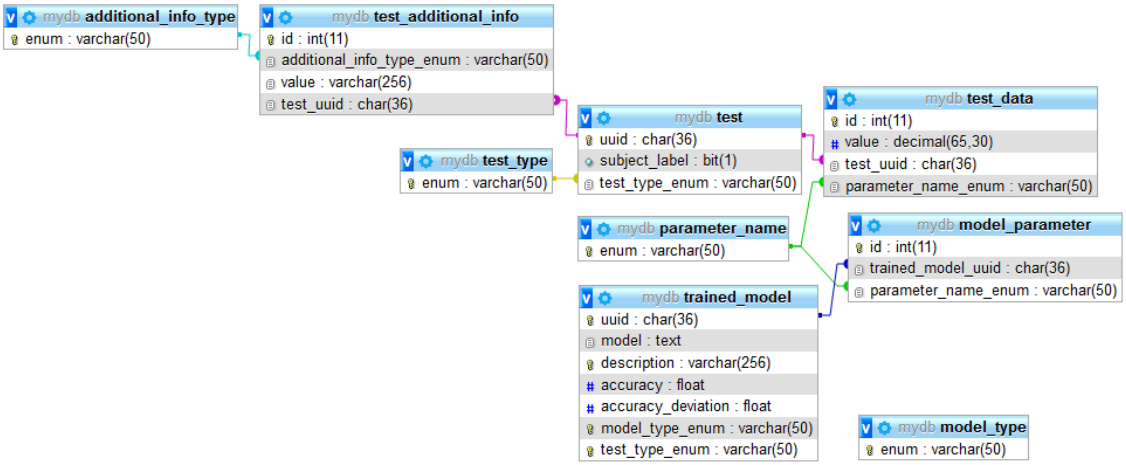


Figure 2. *Structure of created database*

Two directions of application development were considered for the realisation of set functionality requirements. At first, a desktop application seemed sufficient to provide all the needed functionality, but on the other hand, the advantages of web application and it's scalability were the key values toward choosing that type of architecture.

As the author decided to develop a web application, several frameworks were considered. The most prominent one being Django. While the author had not used Django in any previous projects, the notions of other developers and enthusiasts were acquainted along with the documentation. Taking into account that learning the Django framework is considered tedious for beginners and contains a lot of overhead, the search for similar, yet lighter and more flexible framework was done [17, 18].

The resulting framework to meet such requirements is known as Flask. By design, Flask includes more flexibility as compared to Django and is meant to be extended. It is a great tool to learn web development best practices and fundamentals. The rise of microservices also accounts in favor of Flask framework. As stated in Flask documentation, it is "a microframework for Python based on Werkzeug, Jinja 2 and good intentions." [19].

Werkzeug is a Web Server Gateway Interface library. Jinja 2 is a template engine for Python, inspired by Django, but extended [19].

Rendered pages of the application, named templates, were written in Hypertext Markup Language. Charts of the application were done using Highcharts library, as a common and popular tool for creating interactable JavaScript charts for a web page [20]. As Flask does not support forms by default, the Flask-WTF library with WTForms was used.

3.9.3 User Interface

The main page of the developed application can be seen on Figure 3. In that view, user can easily and quickly select a file from the device and in the event of clicking on the primarised button "Load & Analyze" will be redirected to the second view of the application. The second view is presented on Figure 4 and contains a drop-down list of the selection of trained models, ordered by accuracy and including assigned descriptions about the classifier.

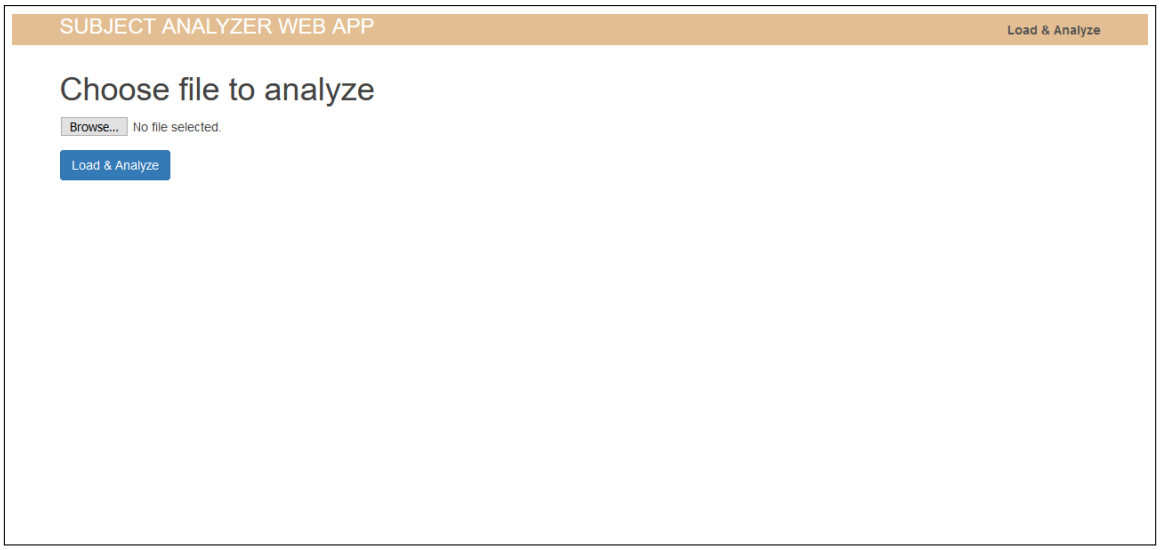


Figure 3. *Main page of developed application*

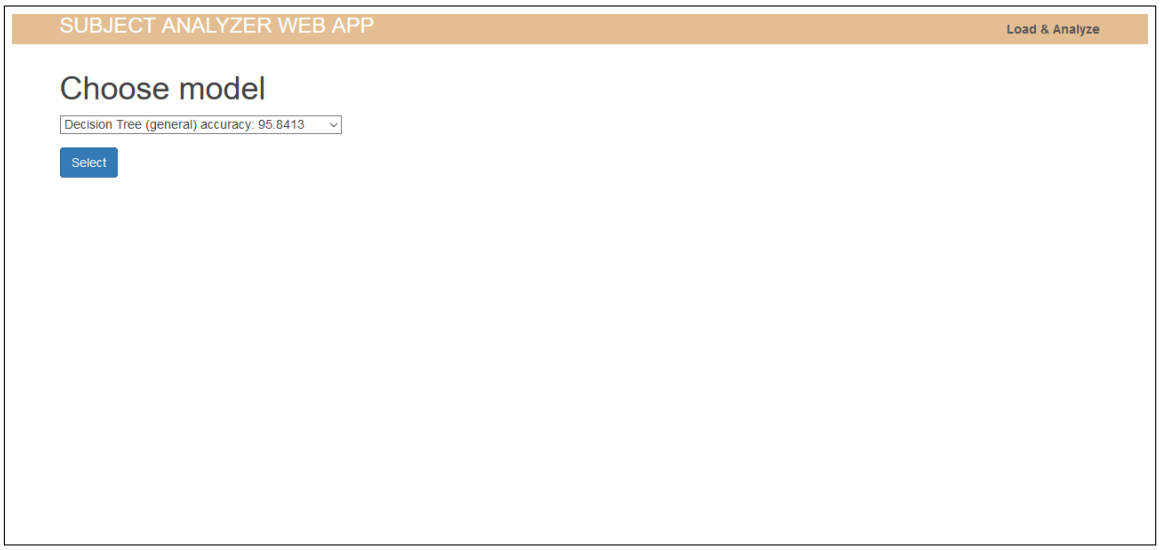


Figure 4. *Web page of developed application containing trained model selection*

After the selection of classifier model, a prediction about the data provided in the input file will be shown. In addition, a one, two or three dimensional graph will be shown, according to the amount of parameters with which the model was trained. An example of one dimensional graphs are presented in Figure 5.

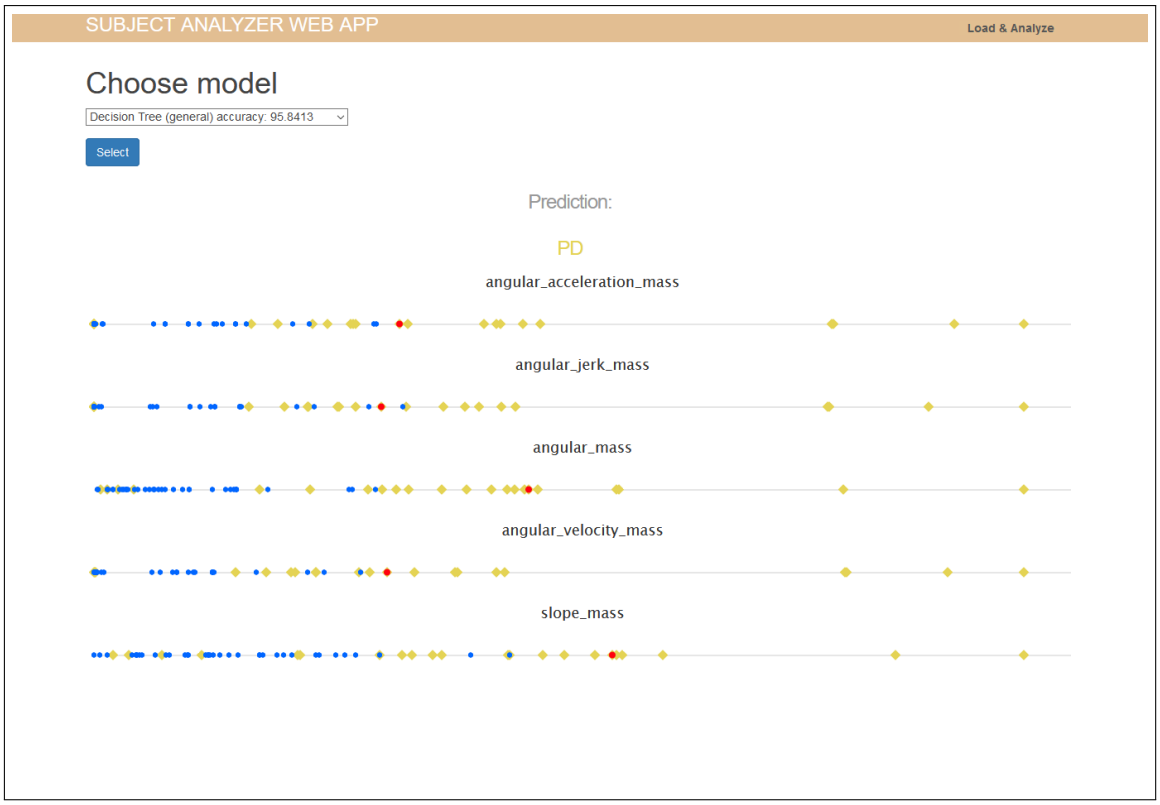


Figure 5. *Data visualization with one dimensional graph*

To better demonstrate the classification results made by trained machine learning models, data point of subject along with the scatter plot with training parameters and drawn decision boundary was presented.

To understand the meaning of decision boundary, one must first familiarize with decision regions. Decision region is used to name an area of classified data that corresponds to one specific output class, in current thesis either PD or HC. Within the dataset, there can be multiple decision regions. Decision boundary is the crossing point of two different decision region [21].

Decision boundary implementation was done similarly to research [22]. Firstly, a step was chosen and for the entire plane of data points, an abstract grid of lines after each step was placed along horizontal and vertical direction. From every appearing cell, the trained model was used to predict the class value inside. Then, the search of border points was conducted. For that, a three by three matrix was placed along both directions of the grid, recursively scanning through the predicted values and when the matrix contained at least one element from other class, it was marked as a border point. Decision boundary was the unification of all border points.

Examples of two and three dimensional graphics are presented in Figure 6 and Figure 7

accordingly.

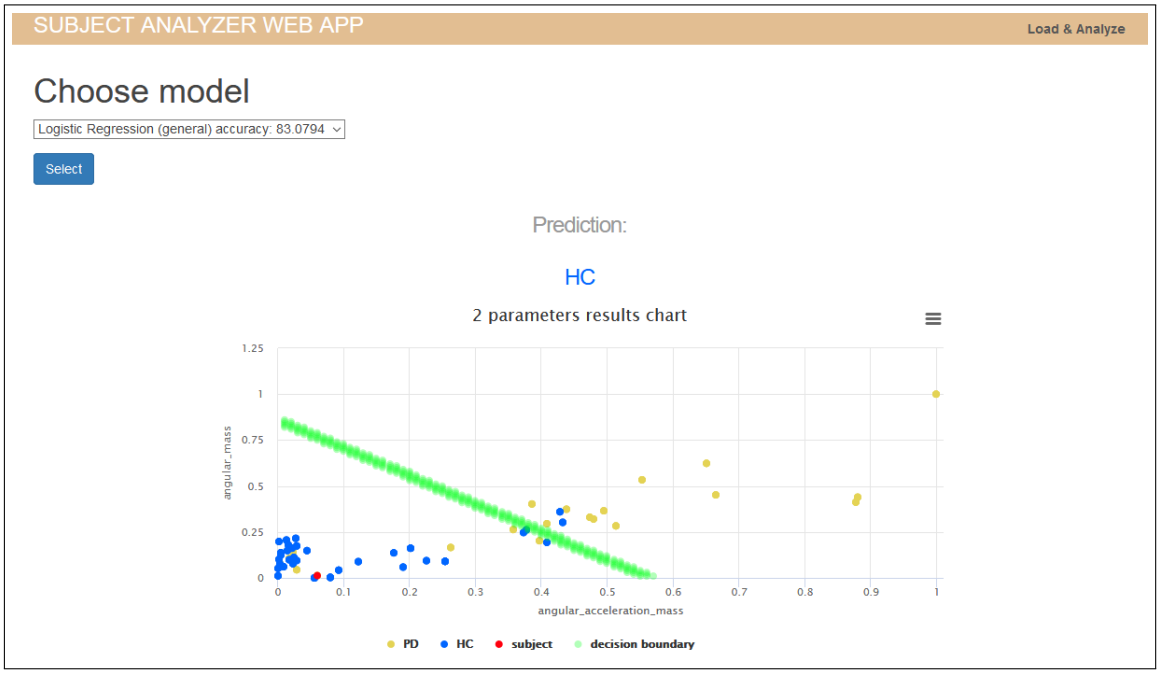


Figure 6. *Data visualization with two dimensional graph*

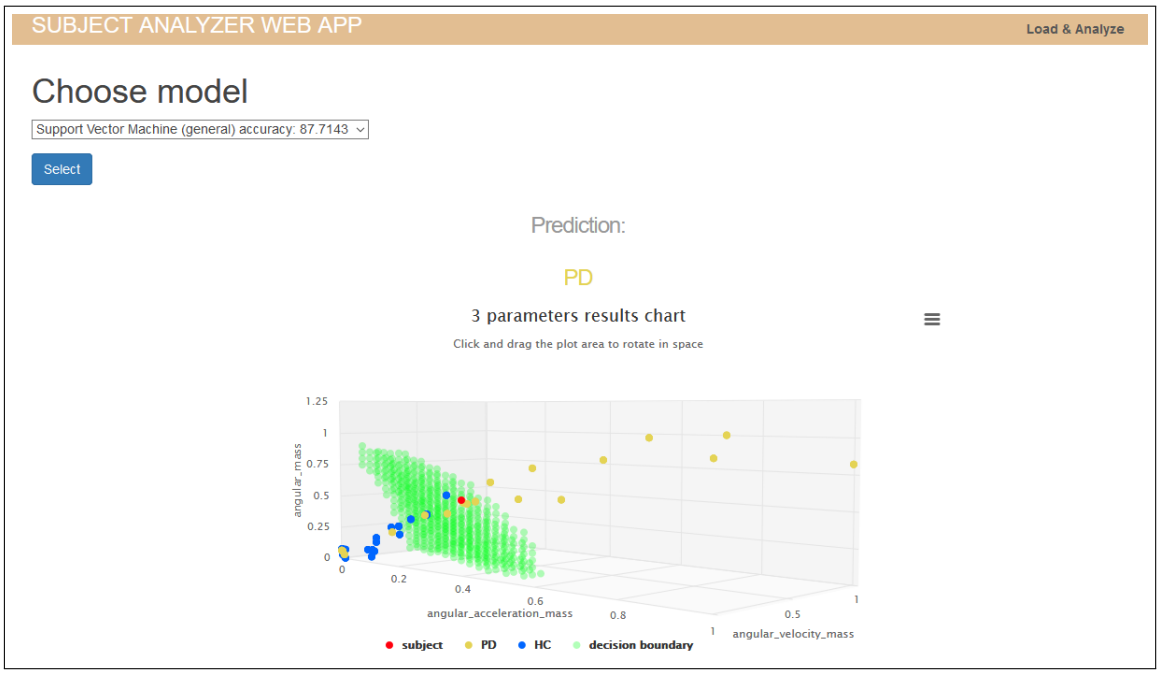


Figure 7. *Data visualization with three dimensional graph*

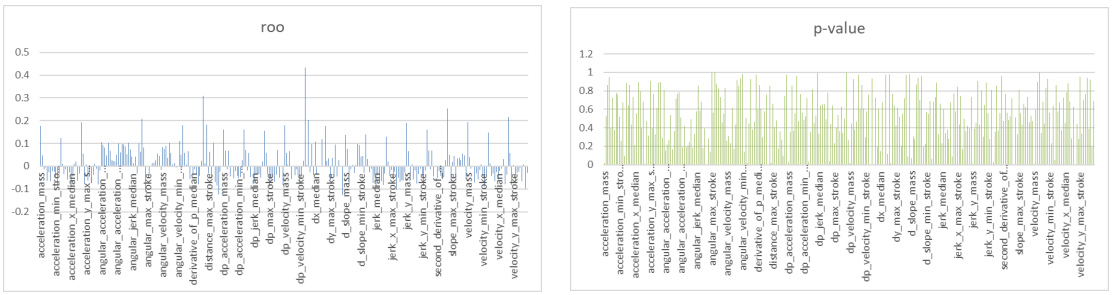
As it can be seen on Figure 7 the decision boundary is not in one linear format. Thus, the boundary points are scattered and warp into various shapes. During the development of the application, the author tried to modify the implementation of decision boundary visualisation through using polygon plot provided by Highcharts [20], but was unable to do so, for the reason that all of the boundary points were not always in the same groups and the connection of polygon series resulted in visual anomalies.

Additional drawback was that the step in the creation of the decision boundary for the three dimensional graph had to be rather big. When small step for boundary points creation was used, the resulting huge amount of points made the graph unresponsive.

4. Results

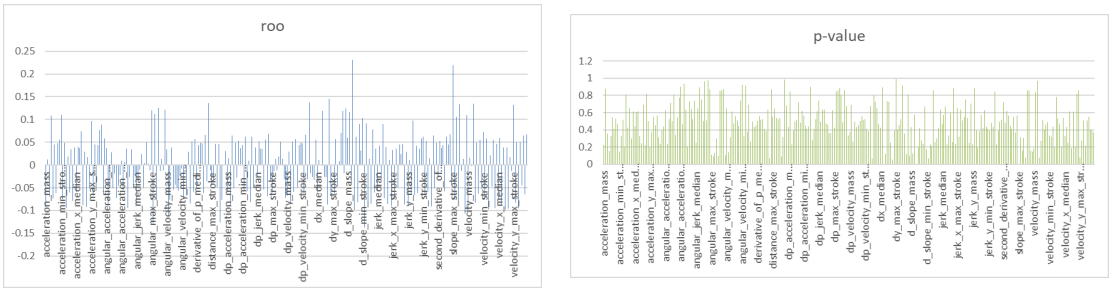
In this study Fisher's score, from Section 3.2 is used to ensure the best feature selection of training parameters for machine learning models. The selected features and their Fisher's scores are presented in Table 1.

In current thesis, the relation between test groups in the examined battery was also explored. For that, Pearson's correlation coefficient, discussed in Section 3.3 was used. As it was there stated, the maximum positive and negative correlation coefficients were rather low and did not have any significant effect. Obtained results are further presented in the Figures 8 - 10, confirming these claims. Therefore no vital linear correlations between examined battery of tests were detected.



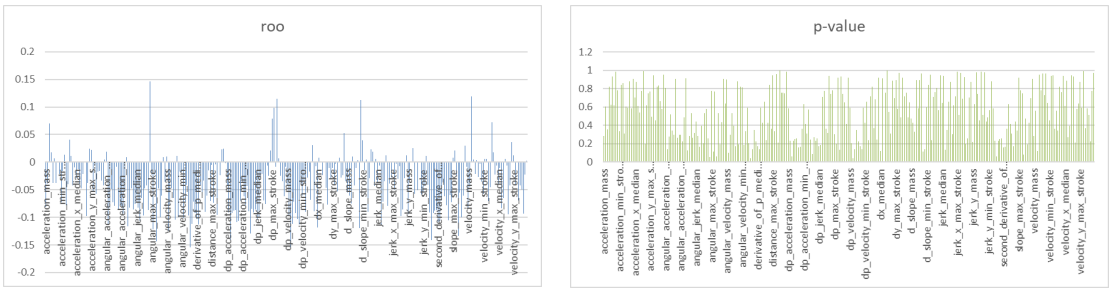
(a) *Pearson's correlation coefficients* (b) *P-values*

Figure 8. *pcontinue and plcontinue*



(a) *Pearson's correlation coefficients* (b) *P-values*

Figure 9. *ptrace and pltrace*



(a) *Pearson's correlation coefficients* (b) *P-values*

Figure 10. *pcopy and plcopy*

Finally, an application was developed that allows the analyzation of subject's fine motor test data and according to trained machine learning models, predict the class of said data. Two segmentation classes were explored: PD and HC. A visual demonstration consisting of data points used in model training, subject data and decision boundary was presented for better analysis of the prediction results. For that purpose one, two and three dimensional graphs were implemented.

5. Discussion

The Pearson's correlation coefficient has provided various interesting results. However, the analyzation could be extended to examine the relations between permutations of the test battery and in the feature include other fine motor tests that were not included in current thesis.

Although in the comparison of machine learning models trained by each individual test data and by test battery, the analysis parameters showed, in some cases, the superiority of model trained on test battery, the analysis was concluded with small sample size. In further developments, results with bigger sample sizes would be recommendable.

Finally, the realisation of decision boundaries could be modified to suit bigger sample sizes and to visually appear more distinguishable.

6. Conclusions

The first goal of the thesis was to investigate the relations between the tests in examined battery. The purpose being the optimisation of battery by omitting the correlating tests that overlap with one another. By the methods used on the provided data, the thesis found no linear correlations and therefore the optimisation of the battery including the tests explored, could not be done.

In the case of sample range within the size boundary included in this thesis, classificators trained in the scope of examined battery proved to even exceed some of the classifiers trained in the scope of each individual test. The results could not be directly applied to larger test samples. For that, further research must be carried out.

For the purpose of this research, an application was developed. The application allows the analization of performed fine motor test data and the visualization of machine learning model predictions along with the parameters of PD diagnosed patients and HCs. In future developments, the application could be used with other types of machine learning algorithms in addition to the ones already used.

Bibliography

[1] P.Johns. *Clinical Neuroscience*. 2014.

[2] Sven Nõmm, Konstantin Bardõš, Aaro Toomela, Kadri Medijainen, and Pille Taba. "Detailed analysis of the Luria's alternating series tests for Parkinson's disease diagnostics". In: *17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. 2018.

[3] Thomas-Bairam Toodo. "Assessment of parameters from the handwritten sentence test used to diagnose Parkinsons disease". MA thesis. Tallinn University of Technology, 2018.

[4] J. Jankovic. "Parkinson's disease: clinical features and diagnosis". In: *Journal of Neurology,Neurosurgery & Psychiatry* 79.4 (2008), pp. 368–376.

[5] Michael T. Hayes. "Parkinson's Disease and Parkinsonism". In: *The American Journal of Medicine* 132.7 (2019), pp. 802–807.

[6] P. Stępień, J. Kawa, D. Wieczorek, M. Dąbrowska, J. Sławek, and E. J. Sitek. "Computer Aided Feature Extraction in the Paper Version of Luria's Alternating Series Test in Progressive Supranuclear Palsy". In: *Advances in Intelligent Systems and Computing*. 2018.

[7] C. M. Kipps and J. R. Hodges. "Cognitive Assessment for Clinicians". In: *Journal of Neurology,Neurosurgery & Psychiatry* 76 (2005).

[8] Charu C. Aggarwal. *Data Mining The Textbook*. 2015.

[9] Quanquan Gu, Zhenhui Li, and Jiawei Han. "Generalized Fisher score for feature selection". In: *UAI'11: Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*. 2011.

[10] Xanthoula Eirini Pantazi, Dimitrios Moshou, and Dionysis Bochtis. *Intelligent Data Mining and Fusion Systems in Agriculture*. 2020.

[11] Timothy M.D. Ebbels. *The Handbook of Metabonomics and Metabolomics*. 2007.

[12] V. N. Gudivada. "Handbook of Statistics". In: Elsevier, 2016. Chap. Chapter 5 - Cognitive Analytics: Going Beyond Big Data Analytics and Machine Learning, pp. 169–205.

[13] Priyanka A.Abhang, Bharti W.Gawali, and Suresh C.Mehrotra. “Introduction to EEG- and Speech-Based Emotion Recognition”. In: Academic Press, 2016. Chap. Chapter 3 - Technical Aspects of Brain Rhythms and Speech Parameters, pp. 51–79.

[14] Sven Nomm, Aaro Toomela, Julia Kozhenkina, and Toomas Toomsoo. “Quantitative analysis in the digital Luria’s alternating series tests”. In: *14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. 2016.

[15] Sven Nõmm, Konstantin Bardõš, Ilja Mašarov, Julia Kozhenkina, Aaro Toomela, and Toomas Toomsoo. “Recognition and Analysis of the Contours Drawn during the Poppelreuter’s Test”. In: *15th IEEE International Conference on Machine Learning and Applications (ICMLA)*. 2016.

[16] Sanjay Yadav and Sanyam Shukla. “Analysis of k-Fold Cross-Validation over Hold-Out Validation on Colossal Datasets for Quality Classification”. In: *2016 IEEE 6th International Conference on Advanced Computing (IACC)*. 2016.

[17] Michael Herman. *Django vs. Flask in 2019: Which Framework to Choose*. [Accessed: 2020-03-03]. URL: https://testdriven.io/blog/django-vs-flask/.

[18] *Python Developers Survey 2018 Results*. [Accessed: 2020-03-03]. URL: https://www.jetbrains.com/research/python-developers-survey-2018/.

[19] *Flask documentation*. [Accessed: 2020-03-04]. URL: https://flask.palletsprojects.com/en/1.1.x/.

[20] *Highcharts*. [Accessed: 2020-03-08]. URL: https://www.highcharts.com/.

[21] D. R. Baughman and Y. A. Liu. “Neural Networks in Bioprocessing and Chemical Engineering”. In: Academic Press, 1995. Chap. 3 - Classification: Fault Diagnosis and Feature Categorization, pp. 110–171.

[22] Oriol Pujol and David Masip. “Geometry-Based Ensembles: Toward a Structural Characterization of the Classification Boundary”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2009.

Appendix 1 - Computed features

Table 9. *Explored features*

Feature	**Description**
L_T	Total length of the drawn line
L_{Tx}	Total length of the drawn line for x axis
L_{Ty}	Total length of the drawn line for y axis
V_T	Velocity mass
V_{Tav}	Average velocity
V_{Tmed}	Median velocity
V_{Tmax}	Maximal velocity
V_{Tmin}	Minimal velocity
V_{Tstd}	Standard deviation of velocity
V_{Tx}	Horizontal velocity mass
V_{Tav_x}	Average horizontal velocity
V_{Tmed_x}	Median horizontal velocity
V_{Tmax_x}	Maximal horizontal velocity
V_{Tmin_x}	Minimal horizontal velocity
V_{Tstd_x}	Standard deviation of horizontal velocity
V_{Ty}	Vertical velocity mass
V_{Tav_y}	Average vertical velocity
V_{Tmed_y}	Median vertical velocity
V_{Tmax_y}	Maximal vertical velocity
V_{Tmin_y}	Minimal vertical velocity
V_{Tstd_y}	Standard deviation of vertical velocity
A_T	Acceleration mass
A_{Tav}	Average acceleration
A_{Tmed}	Median acceleration
A_{Tmax}	Maximal acceleration
A_{Tmin}	Minimal acceleration
A_{Tstd}	Standard deviation of acceleration
A_{Tx}	Horizontal acceleration mass
A_{Tav_x}	Average horizontal acceleration

A_{Tmed_x}	Median horizontal acceleration
A_{Tmax_x}	Maximal horizontal acceleration
A_{Tmin_x}	Minimal horizontal acceleration
A_{Tstd_x}	Standard deviation of horizontal acceleration
A_{Ty}	Vertical acceleration mass
A_{Tav_y}	Average vertical acceleration
A_{Tmed_y}	Median vertical acceleration
A_{Tmax_y}	Maximal vertical acceleration
A_{Tmin_y}	Minimal vertical acceleration
A_{Tstd_y}	Standard deviation of vertical acceleration
J_T	Jerk mass
J_{Tav}	Average jerk
J_{Tmed}	Median jerk
J_{Tmax}	Maximal jerk
J_{Tmin}	Minimal jerk
J_{Tstd}	Standard deviation of jerk
J_{Tx}	Horizontal jerk mass
J_{Tav_x}	Average horizontal jerk
J_{Tmed_x}	Median horizontal jerk
J_{Tmax_x}	Maximal horizontal jerk
J_{Tmin_x}	Minimal horizontal jerk
J_{Tstd_x}	Standard deviation of horizontal jerk
J_{Ty}	Vertical jerk mass
J_{Tav_y}	Average vertical jerk
J_{Tmed_y}	Median vertical jerk
J_{Tmax_y}	Maximal vertical jerk
J_{Tmin_y}	Minimal vertical jerk
J_{Tstd_y}	Standard deviation of vertical jerk
M_T	Slopes mass
M_{Tav}	Average slopes
M_{Tmed}	Median slopes
M_{Tmax}	Maximal slopes
M_{Tmin}	Minimal slopes
M_{Tstd}	Standard deviation of slopes
θ_T	Angular mass
ψ_{Tav}	Average yaw angle
ψ_{Tmed}	Median yaw angle

ψ_{Tmax}	Maximal yaw angle
ψ_{Tmin}	Minimal yaw angle
ψ_{Tstd}	Standard deviation of yaw angle
ω_T	Angular velocity mass
ω_{Tav}	Average angular velocity
ω_{Tmed}	Median angular velocity
ω_{Tmax}	Maximal angular velocity
ω_{Tmin}	Minimal angular velocity
ω_{Tstd}	Standard deviation of angular velocity
α_T	Angular acceleration mass
α_{Tav}	Average angular acceleration
α_{Tmed}	Median angular acceleration
α_{Tmax}	Maximal angular acceleration
α_{Tmin}	Minimal angular acceleration
α_{Tstd}	Standard deviation of angular acceleration
ζ_T	Angular jerk mass
ζ_{Tav}	Average angular jerk
ζ_{Tmed}	Median angular jerk
ζ_{Tmax}	Maximal angular jerk
ζ_{Tmin}	Minimal angular jerk
ζ_{Tstd}	Standard deviation of angular jerk
P_T	Pressure mass
t	duration
ΔP_T	Change of pressure mass
ΔP_{Tav}	Average pressure change
ΔP_{Tmed}	Median pressure change
ΔP_{Tmax}	Maximal pressure change
ΔP_{Tmin}	Minimal pressure change
ΔP_{Tstd}	Standard deviation of pressure change
$\Delta P v_T$	Pressure change velocity mass
$\Delta P v_{Tav}$	Average pressure change velocity
$\Delta P v_{Tmed}$	Median pressure change velocity
$\Delta P v_{Tmax}$	Maximal pressure change velocity
$\Delta P v_{Tmin}$	Minimal pressure change velocity
$\Delta P v_{Tstd}$	Standard deviation of pressure change velocity
$\Delta P a_T$	Pressure change acceleration mass
$\Delta P a_{Tav}$	Average pressure change acceleration

ΔPa_{Tmed}	Median pressure change acceleration
ΔPa_{Tmax}	Maximal pressure change acceleration
ΔPa_{Tmin}	Minimal pressure change acceleration
ΔPa_{Tstd}	Standard deviation of pressure change acceleration
ΔPa_{T}	Pressure change jerk mass
ΔPa_{Tav}	Average pressure change jerk
ΔPa_{Tmed}	Median pressure change jerk
ΔPa_{Tmax}	Maximal pressure change jerk
ΔPa_{Tmin}	Minimal pressure change jerk
ΔPa_{Tstd}	Standard deviation of pressure change jerk