KAUNAS UNIVERSITY OF TECHNOLOGY

ANDRIUS LAURAITIS

HYBRID CLASSIFICATION MODEL FOR NEURAL IMPAIRMENT DETECTION

Doctoral Dissertation
Natural Sciences, Informatics (N 009)

2020, Kaunas

This doctoral dissertation was prepared at Kaunas University of Technology, Faculty of Informatics, Department of Multimedia Engineering, during the period of 2015–2019.

Scientific Supervisor:

Prof. Dr. Rytis MASKELIŪNAS (Kaunas University of Technology, Technological Sciences, Informatics Engineering, T007 and Natural Sciences, Informatics, N 009).

Doctoral dissertation has been published in:
http://ktu.edu

Editor:

Brigita Brasienė (Publishing house "Technologija")

KAUNO TECHNOLOGIJOS UNIVERSITETAS

ANDRIUS LAURAITIS

NEUROLOGINIŲ SUTRIKIMŲ APTIKIMO HIBRIDINIS KLASIFIKAVIMO MODELIS

Daktaro disertacija
Gamtos mokslai, Informatika (N 009)

2020, Kaunas

Disertacija rengta 2015-2019 metais Kauno technologijos universiteto Informatikos fakultete Multimedijos inžinerijos katedroje.

Mokslinis vadovas:

Prof. Dr. Rytis MASKELIŪNAS (Kauno technologijos universitetas, technologijos mokslai, informatikos inžinerija, T 007 ir gamtos mokslai, informatika, N 009).

Interneto svetainės, kurioje skelbiama disertacija, adresas:
http://ktu.edu

Redagavo:

Brigita Brasienė (leidykla “Technologija”)

Table of Contents

List of figures

List of tables

Terms and abbreviations

$\wedge$ – logical 'and'
$\vee$ – logical 'or'
A – spiral contour with representation, $A: [0, 1] \rightarrow SF$
A (α(t)) – spiral point in t time moment
AD – Alzheimer disease
$AdjM_i$ – *i*-th transformed adjacency matrix of a predefined graph (e.g., patient formed)
ANN – artificial neural network
AUC – area under ROC
A_r – normalizing factor for the *r-th* Mel-filter
A_s – logarithmic frequency abscissa
A_{tdee} – age in years (TDEE task)
a – Archimedean spiral turn parameter
$a_{k,t}$ – power spectral density of the *k-th* harmonic
a_{maxx} – maximum finger acceleration in x direction
a_{maxy} – maximum finger acceleration in y direction
a_x – rate at which finger velocity changes in x direction
a_y – rate at which finger velocity changes in y direction
B – patient drawn (followed) contour with representation $B: [0, 1] \rightarrow SF$
B (β(t)) – patient drawn point in t time moment
B_e – value of established bill (*Q2* in similarities, calculation task)
BiLSTM – bidirectional LSTM
BMR – basal metabolic rate
BPNN – back-propagation neural network
b – Archimedean spiral distance between successive tuning control parameter
bin – a segment of the frequency axis that collects the amplitude, magnitude or energy from a small range of frequencies
b_1, b_2 – band edges, in bins, over which to calculate the spectral method (e.g., slope)
C – number of correctly drawn CDT components
Cal100 – input product calorie norm (for 100 grams)
CCI – correctly classified instances
CD – current day
CDT – clock drawing task
CEP – cepstrum pitch determination
CM – current month
CNSD – central nervous system disorders
COG – centre of gravity
CPU – central processing unit
CY – current year
C0_C – correctly classified instances, target class = 0
C0_I – incorrectly classified instances, target class = 0
C1_C – correctly classified instances, target class = 1
C1_I – incorrectly classified instances, target class = 1
D – day input (integer number)
DCN – denomination coins (*Q3* in similarities, calculation task)

DCT – discrete cosine transform
DEF – number of required operations defined (problem solving task)
DFT – discrete Fourier transform
D_i – *i-th* activity duration in minutes (TDEE task)
Diff – difference between 2 dates (in days)
DNN – deep neural network
D_x – finger position change count in x direction
D_y – finger position change count in y direction
d – *SF* distance function (defines a distance between each pair of elements of a set)
$delta_i$ – distance between *i*-th object centre coordinates and *i*-th screen touch
delcount – number of patient performed delete operations (problem solving task)
E(f) – amplitude spectrum signal (SRH method), computed for each Hanning-windowed frame, covering several cycles of the resulting residual signal
EC – error count (classifier evaluation)
Elem – number of correctly listed items (verbal fluency task)
err_binary – total number of binary connection violation for a particular node
err_contour – number of active zone violations in CDT task
err_exist – total number of errors when already existing connection is selected
err_logical – total number (counter) of wrongly connected nodes
err_opviolation – total number of operation violations (problem solving task)
err_outside – total number of clicks outside the node
err_toofew – total number of errors when too few nodes are selected when forming a connection
err_tlimit – total number of time limit exceed errors
err_toomany – total number of errors when too many nodes are selected when forming a connection
err_samenode – total number of errors when clicking on the same node
F – Frechet distance
FFBP – feed-forward backpropagation neural network
FFT – fast Fourier transform
FLDA – Fisher's linear discriminant analysis
FMTF – finger motion tracking features
FN – false negative
FP – false positive
FPR – false positive rate
FTT – finger tapping tasks
F1 – F-Measure
f – frequency (Hz)
f(i,j), g(i,j) – enumeration functions (functions that give the maximum similarity value for each node in the given node list)
f_k – frequency in Hz corresponding to bin *k*
φ_{ftmf} – average *atan2* angle (radians) (between positive x-axis of a plane and point given by coordinates (x, y))
$\varphi\omega(t)$ – designed Gabor wavelet with properties: $\omega = 2^{k/Q}$, Q is the number of wavelets per octave, $k \in Z$
G – *SAGE* questionnaire variant (group) in interval [1; 4]

GPU – graphics processing unit
$GTCC_t$ – Gammatone cepstral *t-th* coefficients
GT – Gammatone filter
g(*t*) – properties of Gammatone filter
graph_NM – similarity measure between 2 graphs (neighbour matching method) in an interval [0; 1]
$gtcc_t$ *[m]* – GT analysis for GTCC calculation in *m-th* signal sample at time *t*
H(s) – audio signal modelling (LHS method), $s = \log_2 f$
HD – Huntington disease
H_{tdee} – height in meters (TDEE task)
h_n – equal to 0.84^{n-1}, i.e., decreasing sequence, implying that higher harmonics contribute
ICI – incorrectly classified instances
Img1 – first image in picture naming task
Img2 – second image in picture naming task
id() – in-degree of the node
inf – infimum ($a \leq x,\ \forall\ x \in SF,\ SF \subseteq PMS$)
inscount – number of patient performed insert operations (problem solving task)
JA – *Jaro Winkler* algorithm for symbol-by-symbol string matching
K – number of peaks in audio signal
KP – Kappa coefficient
kNN – k-nearest neighbours classifier
L – edge parallelism in degrees (3-D construction task)
L_r – DFT lower index ($V_r[k]$ calculation)
LDA – linear discriminant analysis
LHS – log-harmonic summation
LMT – logistic model trees
LSTM – long short-term memory
LWL – locally weighted learning
M – month input (integer number)
M_n – specified money value (*Q2* in similarities, calculation task)
MAE – mean average error
MCC – Matthew's correlation coefficients
MCI – mild cognitive impairment
MET_i – item MET coefficient (Metabolic equivalent of task)
$MFCC_t$ – Mel-frequency cepstral *t-th* coefficient
MF_t *[r]* – Mel-frequency spectrum at analysis time *t* for $r = 1,2, \ldots R=22$
MLP – multi-layer perceptron
MSE – mean squared error
mp – Archimedean spiral point match percentage
msp – number of patient clicks in device screen
mj – number of matching symbols for *JA*
m1, *m2* – slopes of two analysed graph edges
mpatient – input product quantity (in grams)
mnorm – value of 100 (grams)
m_{tdee} – number of patient activities (inputted),
μ_f – mean frequency
μ_s – mean spectral value

μ_1 – spectral centroid
μ_2 – spectral spread
N – number of harmonics taken into account
$N_t(f)$ – power spectral density of the unwanted noise
NCF – normalized correlation function
NITS – neural impairment test suite
N_{vx} – magnitude of the rate at which pen tip changes its position in x direction
N_{vy} – magnitude of the rate at which pen tip changes its position in y direction
n – total number of visual objects (circles)
n_c – compression factor
nsp – number of Archimedean spiral points
n_{tdee} – number of patient input products
OER – open educational resources
od() – *out*-degree of the node
P – norm type (2 or 1 scalar)
PAL – physical activity level
PCA – principal component analysis
PD – Parkinson disease
PEF – pitch estimation filter
PMF – probability mass function
PMS, SF – metric space (e.g., 2-D)
PRC – precision-recall plot
PrC_0 – prediction confidence, target class 0
PrC_1 – prediction confidence, target class 1
PS – problem solving
p_{avg} – average finger screen pressure [0; 1]
p_i – i-th object centre coordinates *(x, y)*
p_d_gained – patient daily gained calories (from food)
p_d_burned – patient daily-burned calories (from activities)
p_d_balance – patient daily calorie balance (from activities)
p_e – hypothetical probability of chance agreement
p_n – logarithmic value of finger travelled path divided by φ_{ftmf}
p_0 – relative observed agreement among raters
psp – indicator (1 yes, 0 no) if screen touch point match a point from Archimedean spiral contour
Q1 – first question in similarities, calculation task
Q2 – second question in similarities, calculation task
Q3 – third question in similarities, calculation task
RAE – relative absolute error
RBF – radial basis function
RMSE – root mean squared error
RNN – recurrent neural networks
ROC – receiver operating characteristics, curve
RRSE – root relative squared error
r, θ – Archimedean spiral representation in polar coordinates
rt_i – i-th object touch reaction time (seconds)

rsp – Archimedean spiral point radius
S – how many coins are needed to collect (*Q3* in similarities, calculation task)
SAGE – self-administered cognitive testing
S_{avg} – average finger touch area ratio [0; 1]
SAGE_TOTAL – sum of SAGE task scores
SD_{vx} – standard Deviation of velocity in x direction
SD_{vy} – standard Deviation of velocity in y direction
SGD – stochastic gradient descent
$S_{in}(i,j)$ – in degree similarity of node *i* in one graph and *j* in the second graph
$S_{out}(i,j)$ – out degree similarity of node *i* in one graph and *j* in the second graph
SMO – sequential minimal optimization
SRH – summation of residual harmonics
STFT – short time Fourier transform
SVM – support vector machine
s_k – spectral value at bin *k*
st_i – *i*-th screen touch coordinates *(x ,y)*
s1 – predefined line input text (e.g., string for holding correct answer for task question)
s2 – line input text (provided by a patient)
T_C, T_E –threshold centroid, threshold energy (speech analysis)
TDEE – total daily energy expenditure
TDictionary – correct answer dictionary
TFC – total functional capacity
TN – true negative
TP – true positive
TPR – true positive rate
T1 – computerized task to solve sequential touch problem
T2 – computerized task to solve rainbow-colour touch problem
T3 – computerized task to solve multi-touch problem
T4 – computerized task to solve Archimedean spiral problem
T5 – computerized task to answer questions about general insights
T6 – computerized task to solve orientation (current date) problem
T7 – computerized task to solve picture-naming problem
T8 – computerized task to solve similarities, calculation problem
T9 – computerized task to 3-D construction problem
T10 – computerized task to solve CDT
T11 – computerized task to solve verbal fluency problem
T12 – computerized task to solve modified trials problem
T13 – problem solving computerized task
T14 – computerized task for voice recording
T15 – computerized task to solve TDEE problem
T0 – computerized task to solve memory problem
t – transpositions
t_i – *i*-th object touch time (seconds)
test_duraiton – total test execution time (seconds)
tsage1 – *SAGE* score for orientation (current date) task
tsage2 – *SAGE* score for picture naming task

tsage3 – *SAGE* score for similarities, calculation task (*Q1*)
tsage4 – *SAGE* score for similarities, calculation task (*Q2*)
tsage5 – *SAGE* score for similarities, calculation task (*Q3*)
tsage6 – *SAGE* score for memory task
tsage7 – *SAGE* score for the 3-D construction task
tsage8 – *SAGE* score for CDT task
tsage9 – *SAGE* score for the verbal fluency task
tsage10 – *SAGE* score for the modified trials task
tsage11 – *SAGE* score for problem solving task
UCI – University of California, Irvine
U_r – DFT upper index ($V_r[k]$ calculation)
$V_r[k]$ – triangular weighting function (*k*-th harmonic) for the *r-th* filter
v_{avg} – average speed of finger while it is moving on the surface (*s* – travelled path in time *t*)
v_x – finger velocity in x direction
v_y – finger velocity in y direction
v_{xmax} – maximum finger velocity in x direction
v_{ymax} – maximum finger velocity in y direction
W_s – spectral window function
W_{tdee} – weight in kilograms (TDEE task)
WST – wavelet scattering transform
x_{ij}^{k+1} – similarity of *i*-th node of one graph and *j*-th node of the second graph in (*k+1*) th iteration
Y – year input (integer number)
$Y_t(f)$ – audio signal modelling (PEF method)
less to the pitch than lower harmonics do noise
Z – set of integer numbers

1. INTRODUCTION

1.1. Relevance of the research

Computational intelligence provides many models with devoted applications for the classification of health disorders. Some solutions are developed for medical image processing, while others use data from clinical examinations to predict the state of health. With the development of technology, new forms of smart diagnostic systems such as for in-vitro medical testing [1] are introduced. Nowadays, mobile smart devices are widely used around the globe. The applications on smartphones make use of multimedia capabilities, which can provide an extensive information about various features of our bodies. In order to compose a smart system that can actively participate in medical examinations and provide clinical decision support, or simply support users in a daily routine, the classification models that are able to use data collected in the process of automated medical examinations are needed.

Mobile health applications are on the rise, with many people that are utilizing apps successfully in diverse ranges of medical practices. The apps range from medical information systems [2], physiological health care systems [3], health alert/identification systems and connection with healthcare providers [4] to calculating required doses of medicine, such as insulin [5], and often in combination with some e-health wearable device [6], sometimes with a very specific application in detecting premature ventricular contraction [7]. Diagnostic apps have a huge potential to provide access to diagnostic definitions and might appeal to both healthcare professionals as it can invite a patient for an early diagnosis and patients themselves by offering a diagnostic adjunct [8]. The use of such systems can improve the treatment of the Huntington Disease (HD), support medical doctors caregivers in remote monitoring of the HD patients [9].

The central nervous system motor disorders (CNSD) are mainly related to problems that people may have related to controlling their bodies and mind actions. The number of people suffering from pathological tremor has been increasing steadily with the ageing of the population. As for numbers, 4% of the elderly (>65 years) suffer from essential tremor (ET), whereas 1% of people that are aged more than 50 years suffer from the Parkinson's disease (PD) [10]. The Alzheimer's Disease (AD) is a neurodegenerative disease, which symptoms are memory loss and behaviour problems that gradually worsen in time to language difficulties, disorientation and behavioural problems with managing self-care, what finally leads to dementia. The Huntington Disease (HD) is mainly inherited from the family and reveals itself in the death of brain cells. In the early stage, people have subtle problems with mental abilities, lack of coordination or unsteady gait.

At a later stage, patients have uncoordinated moves that gradually worsen to serious difficulties in talk and dementia. Mild Cognitive Impairment (MCI) is a condition in which an individual's thinking ability shows some mild changes that can be easily noticed by the people who are close to the affected person. Some cases of MCI are actually considered an early stage of AD. It is, therefore, an interest of researchers to detect these diseases in the early stage before they progress further and make medical treatments ineffective.

1.2. Aim of the research

To develop a computerized model, which predicts and identifies neural impairments for patients at the early stage of central nervous system disorders (CNSD).

1.3. Tasks of the research

Five tasks were set in order to achieve this aim:

1. Analyse and compare the state of the art data mining and machine learning approaches that involve patients with central nervous system disorders (CNSD) in classification problematics studies;
2. Create a computerized extended version of data collection methodology via smart interface (mobile phone and tablet), adapted for the early stage patients with CNSD;
3. Design a system for feature extraction mechanism based on cognitive, speech, energy expenditure impairments occurring in CNSD patients and collect features to a dataset.
4. Develop binary supervised learning classification models based on proposed methodological tasks in order to evaluate sick vs. healthy condition of CNSD patients;
5. Fuse different types (functions, Bayesian, trees, rules, lazy) of single supervised learning classifiers (hybrid model) that improve sick vs. healthy classification accuracy results and decrease performance errors.

1.4. Scientific novelty

These novelty factors are listed:

1. Computerization of predefined Self-Administered Gerocognitive exam (SAGE) methodology for the detection of early signs in memory or thinking cognitive impairments to evaluate patient health state automatically with the proposed smart interface (mobile application);
2. Extension of SAGE methodology with extra tests for tremor, speech and energy expenditure impairments.
3. Innovative patient health state monitoring method for self-assessment and comparison with measurements of healthy subjects.
4. Implementation of hybrid classification model (all tasks are considered from the proposed extended methodology) with improved accuracy (compared to the standalone models) to predict possible deterioration of CNSD patient health status.

1.5. Research object

Data mining techniques for pattern recognition showing healthy vs. sick (person suffering from neurological disorders such as Huntington Disease) classification intuition and machine learning models for the prediction of future outcomes.

1.6. Practical significance

Developed mobile application "Neural Impairment Test Suite" (NITS) [18] for CNSD patient health state monitoring is published in Google Play free of charge. The app is disseminated through various media channels in Lithuania and worldwide [19, 20, 21], presented to the medical community. At the time of writing this dissertation, more than 500 users are currently using NITS app.

1.7. Dissertation statements

1. The proposed extended SAGE methodology enables innovative health state monitoring for people suffering from central nervous system disorders (e.g., Huntington, Parkinson Diseases, Mil-Cognitive Impairment, cerebral palsy), based on tremor, cognitive, speech and energy expenditure impairments.
2. The created NITS system can be adapted for the CNSD patients as an assistive device for self-assessment (e.g., at home or medical institution) and give feedback about the early diagnosis of disease status.
3. The implemented classifier fusion mechanism results in a more accurate and improved performance (less prone to errors) hybrid model, as compared to a single supervised learning classifiers, for solving binary (sick vs. healthy) classification problem.

1.8. Scientific approval

Articles indexed in the Web of Science and Scopus with Impact Factor:

1. Lauraitis, A., Maskeliūnas. R., Damaševičius, R., Polap, D., Wozniak, M. (2019). A smartphone application for automated decision support in cognitive task based evaluation of central nervous system motor disorders. *IEEE Journal of Biomedical and Health Informatics*. Vol. 23, issue 5. Electronic ISSN: 2168-2208. DOI 10.1109/JBHI.2019.2891729. Author contribution: 0.400.
2. Lauraitis, Andrius; Maskeliūnas, Rytis; Damaševičius, Robertas. ANN and fuzzy logic based model to evaluate Huntington disease symptoms // Journal of healthcare engineering. New York : Hindawi. ISSN 2040-2295. eISSN 2040-2309. 2018, vol. 2018, art. no. 4581272, p. 1-10. DOI: 10.1155/2018/4581272. [DOAJ; Scopus; MEDLINE; Science Citation Index Expanded (Web of Science). Author contribution: 0.334.
3. Ivanavičius, Arnas; Simonavičius, Henrikas; Gelšvartas, Julius; Lauraitis, Andrius; Maskeliūnas, Rytis; Cimmperman, Piotras; Serafinavičius, Paulius. Real-time CUDA-based stereo matching using Cyclops2 algorithm: research // EURASIP journal on image and video processing. Cham: Springer Open. ISSN 1687-5176. eISSN 1687-5281. 2018, vol. 2018, art. no. 12, p. 1-15. DOI: 10.1186/s13640-018-0253-2. [DOAJ; Scopus; Science Citation Index Expanded (Web of Science). Author contribution: 0.142.

National and international conference proceedings:

1. Lauraitis, Andrius; Maskeliūnas, Rytis. Investigation of predicting functional capacity level for Huntington disease patients // Information and software technologies: 23rd international conference, ICIST 2017, Druskininkai, Lithuania, October 12–14, 2017: proceedings / Robertas Damaševičius, Vilma Mikašytė (Eds.). Cham : Springer, 2017. ISBN 9783319676418. eISBN 9783319676425. p. 142-149. (Communications in computer and information science, ISSN 1865-0929, eISSN 1865-0937 ; vol. 756). DOI: 10.1007/978-3-319-67642-5_12. Author contribution: 0.500.
2. Gelšvartas, Julius; Simonavičius, Henrikas; Lauraitis, Andrius; Maskeliūnas, Rytis; Cimmperman, Piotras; Serafinavičius, Paulius. Realtime stereo matching using maximally stable extremal regions // Transylvanian review. Author contribution: 0.166.
3. Gelšvartas, Julius; Lauraitis, Andrius; Simutis, Rimvydas; Maskeliūnas, Rytis. Review of assistive technologies for disabled people // Biomedical engineering 2016 : proceedings of international conference / Kaunas University of Technology, Lithuanian Society for Biomedical Engineering. Kaunas : Kauno technologijos universitetas. ISSN 2029-3380. 2016, p. 142-146. Author contribution: 0.250.

4. Maskeliūnas, Rytis; Lauraitis, Andrius. Neural impairment test suite mobile app // Technorama 2019: from vision to innovation! : [2019 m. gegužės 9 d.]: innovation catalogue. [Kaunas]: KTU. 2019, project no. 49, p. 59. Author contribution: 0.500.

In 2018, the author of the dissertation participated in the summer school of Tallinn University and completed the course "EXPERIMENTAL INTERACTION DESIGN", 4 ECTS. The author won R1 price in Technorama 2019 conference.

1.9. Dissertation structure

The dissertation is outlined in 5 chapters. Chapter 2 provides an overview of central nervous system disorders symptoms, including those occurring in Huntington Disease as well as related work of state of the art data mining and machine learning. Chapter 3 covers the materials and methods used in the dissertation, i.e., the design of computerized extended SAGE (Self-administered cognitive testing) methodology, methods for feature extraction and analysis of mathematical models for investigated classifiers. Chapter 4 describes four conducted experiments to solve sick vs. healthy classification problem. Experiment one (E1) applies feature distribution approach for individual tasks. Experiment two (E2) uses integrated feature set for building single unified classifiers and implies a fusion mechanism for developing a hybrid model. Experiments three (E3) and experiment four (E4) deals with voice signals to track speech impairments. Chapter 5 contains discussion and conclusion. Chapter 6 provides a list of literature references; chapter 7 provides dissemination channels.

2. LITERATURE REVIEW

There have been conducted many researches in the computer science field that involve studies with central nervous system disorder (CNSD) patients. Mostly, such investigations include Huntington Disease (HD), Parkinson Disease (PD), Alzheimer Disease (AD), mild cognitive impairment (MCI) and dementia test subjects. These diseases cover a broad range of symptoms occurring in CNSD patients. In particular, tremor (involuntary movements, body balance disorders, muscle stagnancy etc.), cognitive (decision-making difficulties, behavioural distractions, problems to focus attention, memory loss, poor self-care etc.), speech (lack of pronounced words, the use of shorter phrases, pauses) and energy expenditure (weight loss, mainly due to the development of early negative energy balance) impairments.

Computer scientists aim at the development of diagnostic aided tools for neural impairment screening and models for decision support making [179], [18]. In many cases, the aim to be digitalised is based on a set of methodological tasks, approved and validated by medical staff from clinical institutions. Most commonly known methodologies are Montreal Cognitive Assessment (MoCA) [177], Mini-Mental State Examination (MME) [178], Self-Administered Gerocognitive Examination (SAGE) [78] and its electronic version (e-SAGE) [176]. MoCA was developed to target cognitive impairments, which are best adapted for a screening test in MCI and AD patients. MoCA covers short-term memory recall tasks, visuospatial abilities, phonemic fluency, repetition, orientation, trial making (executive), target detection, language assessment. MME test includes orientation in time and place, repeating the named prompts, spelling a given word backwards, speaking a phrase or drawing a shown figure. Similarly, SAGE methodology covers general insights task, orientation (current date), picture naming, similarities, basic calculations (mathematical division and subtraction), construction (3-D figure, analogue clock), language (verbal fluency), executive (modified trials, problem solving) and memory. All three methodologies have a scoring (task evaluation) system, i.e., if the total score of a patient is less than a predefined threshold value, an impaired neuropsychological health state is assessed. The decision support making in the presented context is the process of integrating classification models to the proposed screening tools (refer to 2.2.1 for details).

The literature analysis chapter is structured as follows. Firstly, an overview of Huntington Disease (HD) symptoms is provided. The reason for considering HD is that it covers all four types of targeted impairments. In addition, HD symptoms are the most versatile, as compared to the other neurodegenerative disorders. Moreover, the majority of CNSD test subjects that were involved in the work of this dissertation are HD patients. Next, the related work is outlined in five subsections, each of which coverers computerization aspects (mobile devices for data collection, methodological tasks and classification methods) that are proposed in this

dissertation. Finally, the summary of analysis is written to emphasise the drawbacks of existing solutions and give motivation for the proposed approach.

2.1. Overview of symptoms occurring in CNSD patients: A case with Huntington Disease

Huntington disease (HD) is a progressive genetic neurodegenerative disorder, causing involuntary movement and cognitive problems that significantly affect daily life of HD patients. HD and Parkinson (PD) are currently incurable; thus, most of the current research in this area focuses on identifying the deficits at the early stage of disease in order to benefit from future medical interventions that may help delaying the progress of the disease [11, 12]. HD prevalence map [13] worldwide is provided in Appendice A. Infrequent sickness rate (from 0.5 to 1 cases is Finland, Zimbabwe and very rare in China, Japan and Nigeria). There is no sufficient information on the prevalence of Huntington's disease in countries with white colour on the map. Statistically, the highest sickness rate of HD is in Venezuela (1 in 700), about 1 in 10,000 to 20,000 people in Europa, up to 30,000 in America. HD is 10–100 times more common in Europe than in East Asia. The indicator for such circumstances are chromosomes, which are divided into 3 A, B and C haplotype (specific gene groups that are inherited from one parent) groups. In Europe, the majority of HD patients have A (highest risk) groups, whereas in East Asia they have C (lowest risk).

According to the statistical data gathered from HD association in Lithuania and Vilnius University Hospital Santara Clinic in year 2015, there are 177 officially registered patients in Lithuania [14]. Huntington Disease is incurable; thus, the majority of treatment methods include giving drugs that suppress the symptoms of HD. Moreover, the alternative methods (kinesiotherapy, psychotherapy, rehabilitation, ergo therapy, logo therapy etc.) are available. The possible side effects of defined medications exist for HD patients [14, 17]. Table 2.1 provides an overview of HD characteristic symptoms.

Table 2.1. Summary of Huntington Disease symptoms

Stage	Impairment	Symptoms
Early	Tremors	Anxiety, fussiness, involuntary movements of face and limbs
Moderate		Chorea, dystonia (abnormal muscle tone)
Advanced		Rigidity, akinesia (immobility due to paralysis), Hypokinesia (muscle stiffness), Dysarthria (speech articulation disorder), Dysphagia (swallowing disorders)
Early, Moderate, Advanced	Cognitive, Behavioural (emotional)	Decision making difficulties, problems to focus attention, memory loss, poor self-care, depression, anxiety, sleep disorder, obsessive–compulsive symptoms, irritability, speech, aggression, irritability, suspicion, anger, elaborate

		and eccentric behaviour, inappropriate sexual behaviour, jealousy, apathy
	Speech, Energy, Expenditure	Lack of pronounced words, usage of shorter phrases, pauses, weight loss, negative energy balance

2.2. Related work

2.2.1. Overview of classification algorithms for decision support making to CNSD patients

The problematics of data prediction evolved with the rise of artificial intelligence (AI), machine learning (ML), data mining methods and algorithms. Artificial neural networks (ANN), such as Multi-layer perceptron (MLP), can be used for the classification of accelerometer based tremor signals, invoked by Parkinson patient's involuntary movements [22]. The prediction of Parkinson disease onset by adapting Radial Basis Function Neural Network (RBFNN) for tremor activity data, recorded via stimulation electrodes, using electromyography (EMG) signals, is described in [23]. Dynamic Neural Network (DNN) is used to detect time-varying occurrences of tremor and dyskinesia from the time series data that were acquired from EMG sensors and tri-axial accelerometers that were worn by Parkinson patients [24]. Another approach for designing a prediction model for Parkinson's disease uses a decision tree and Iterative Dichotomiser (ID3) methods to analyse data collected from HD symptoms such as trembling in legs, arms, hands, impaired speech articulation and production difficulties [25].

Hybrid models combine different AI and ML approaches for reproducing intelligent human reasoning process [26]. By using information fusion, hybrid models combine heterogeneous ML approaches and improve the quality of reasoning for complex regression and classification problems [27].

Neuro-fuzzy systems combine neural network and fuzzy logic paradigms to avoid the limitations of neural network explanations to reach decision and limitations of fuzzy logic to automatically acquire the rules that are used for making those decisions [28]. Fuzzy expert systems such as a neuro fuzzy system (ANFIS) can be applied in the assessment of Parkinson's disease with a non-invasive screening system for quantitative evaluation and analysis by using amplitude, frequency, spectral characteristics and trembling localization parameters of input data [29]. The hybrid model is adapted in designing a Decision Support System (DSS) for the intelligent identification of Alzheimer, where neuro-fuzzy system explores approximation techniques from the neural networks to find the parameter of a fuzzy system [30].

Hybrid systems are as well used as a classifier fusion strategy (Bayesian, SVM, k-nearest neighbours) in the prevalence of age-related diseases like Alzheimer's and dementia [31], diagnostics and measurement [32] with wavelet

scattering transform (WST) and norm entropy feature extraction methods. The DSS that uses MLP, RBFNN is applied for monitoring patients with neurological disorders [33]. The data is collected by using non-invasive smart devices (modified mouse and 3-axis accelerometer sensor). The integration of neuro-fuzzy networks and information fusion for multimodal human cognitive state recognition is described in [34]. The projection based learning for meta-cognitive radial basis function network (PBL-McRBFN) is applied to predict Parkinson's disease [35]. Other hybrid systems and applications include nonlinear adaptive system, which fuses brain and gait information algorithmically by using Multistate Markov Model [36].

Accurate Parkinson disease diagnosis model that is based on cluster analysis uses Random Tree, Classification and regression tree (C-RT), ID3, Binary Logistic Regression, k-NN, Partial Least Square Regression (PLS), Support Vector Machines (SVM) [37] and Fuzzy c-means clustering (FCM) [38]. Table 2.2 provides a summary of methods used by the other authors.

Table 2.2. Comparison of various Machine Learning (ML) methods, adapted for neurodegenerative disorders such as Huntington or Parkinson disease to solve prediction and classification problems

Work ref.	ML method	Learning approach	ML Problem	Size of the data set	Number of test subjects	Target group	Involve HD patents
[22]	ANN, MLP	Supervised	Classification	-	21	PD, healthy	No
[23]	RBFNN	Supervised	Regression	-	-	PD	No
[24]	DNN	Supervised	Classification	-	12	PD (8), healthy (4)	No
[25]	Decision tree, ID3	Supervised	Classification	195	31	PD (23), healthy (8)	No
[29]	Adaptive neuro fuzzy	Hybrid		100	-	PD	No
[30]	Neuro fuzzy system	Hybrid		-	-	AD	No
[31]	Fusion of classifiers (Bayesian, SVM, k-nearest neighbour)	Hybrid		640	AD (13), PD (15), HD (16), healthy (16)	AD, PD, HD	Yes
[32]	Neuro fuzzy system	Hybrid		-	-	Only survey was done	No

[33]	ANN + MLP, RBFNN)	Hybrid		-	-	PD	No
[34]	Neuro fuzzy system	Hybrid		-	-	-	No
[35]	PBL-McRBFN	Supervised	Classification	22283	72 (50 PD, 22 healthy)	PD, healthy	No
[36]	Multistate Markov Model	Hybrid		2500	72 (82 PD, 62 healthy)	PD, healthy	No
[37]	Random Tree, (C-*RT*), ID3, Binary Logistic Regression, k-NN, (PLS), (SVM)	Supervised	Classification	195	31 (23 PD, 8 healthy)	PD, healthy	No
[38]	FCM	Unsupervised	Clustering	195	-	PD	No

Additional work that involves the author of this dissertation includes the development of text input-based system for evaluating the condition of Huntington's patients [39] and the use of ANN for predicting the functional capacity of HD [40].

The problem of timely and accurate diagnosis of PD and HD has led to the proposal of many computational intelligence algorithms [41], [42], [43], [44]. A multi-purpose Medical Decision Support System (MDSS) can be implemented via mobile app model, allowing an efficient diagnosis, e.g., using Feed Forward Artificial Neural Networks [45], [46]. Alternatively, the augmented Long and Short-Term Memory (LSTM) framework [47], where the network architecture can naturally incorporate the inputs from embedding vectors of patient symptoms and provide an initial disease hypothesis that are provided by a prediction. Alternatively, the authors of [48] suggest using a Naïve Bayesian algorithm. Convolutional Neural Networks (CNN) have been successfully used to diagnose the Alzheimer's disease (AD) and Mild Cognitive Impairment (MCI) by using structural Magnetic Resonance Imaging (MRI) scans [49].

The accelerometer measurements of muscle tremor yielded significant findings for HD patients and their genetic risk to the offspring. The irregularities in tremor patterns can intensify in time due to the onset of the disorder and, in addition, reflect the effects of drug therapy for those on medication [50]. The tremor symptoms of HD patients are very often confused with PD patients. The spectral analysis methods can be used for the classification of accelerometer-registered tremor signals that are invoked from Parkinson patient's involuntary movements [51]. It is possible to use readings from sensors to predict the PD onset by adapting the Artificial Neural Network (ANN) such as Radial Basis Function Neural Network (RBFNN) for analysing tremor activity from the data recorded via stimulation electrodes [52]. Dynamic Neural Network (DNN) solutions are reported to detect time-varying

occurrences of tremor and dyskinesia of time series data that is acquired from electromyography (EMG) sensors and tri-axial accelerometers hat were worn by the PD patients [53].

The approach of designing a prediction model for PD can be based on the evaluation of movements of lower and upper limbs (trembling in legs, arms, hands) [54], while additional data come with age, environmental factors, impaired speech articulation and production difficulties. In addition, the system was developed by using Arduino hardware to monitor tremors in patients with Parkinson's disease, which were further processed in Android device, analysing postural tremor and kinetic tremor of hands and resting tremor amplitude, with successful prediction results [55].

The investigations in the field of neurological disorders are carried out in using neural networks (NN) for detecting PD by visual attributes in handwriting [59], [60] as well as by upper-limbs movement activities [61], gait and tremor activity [62] [63] or voice data with Principal Component Analysis (PCA) method [64]. Support vector machine (SVM) is applied for MCI prediction in AD [65], NN for prognosis of MCI and dementia in elderly people [66]. As for classification of HD cases with acoustic and lexical features, Deep Neural Networks (DNN), and Long-Short-Term Memory Recurrent Neural Networks (LSTM-RNN) are used [67].

2.2.2. State of the art research findings to detect tremor (motor) impairments

The motor impairments for central nervous system disorder (CNSD) patients can occur in various body parts: hands, feet, jaws, tongue or even an entire body. This section focuses on research for hand tremor detection, based on finger tapping tasks (FTT) and Archimedean spiral contour following (SCF) tasks. According to the state of the art, the research findings, these two types of tasks are considered significant factors for the early detection of motor impairments.

FTT was evaluated in mild cognitive impairment (MCI), Alzheimer's disease and Parkinson's disease (PD) with a light beam two photo-diode sensitive device, i.e., capturing a raw number of taps and inter-tap interval (e.g., the time between consecutive taps) [180]. In Roalf, et al. work [181], autism spectrum disorder (ASD) patients are targeted, and FTT is quantified with a magnetic detection system (consists of oscillation and detection coils, a magnetic sensor and a standard personal computer), which measures temporal processing and finger motion dynamics (velocity, acceleration, distance between two fingers). Roalf, et al. [181] require a special environment and hardware, including a supervisor for the experiment execution.

The method described by the authors Rose, et al. in [182] measures FTT stimulus for PD patients with bass drum sound music box equipment by applying beat perception ability (beat alignment test, *BAT*) and Goldsmiths Musical Sophistication Index evaluation criteria (*Gold MSI*). FFT is applied for the assessment of movement slowness, i.e., bradykinesia (measured as the number of

taps over 15 and the time between taps dwell for each tap) in PD patients by using special gloves that are adapted for touchscreen laptop [183].

Researcher Binder in [184] uses a custom-made electronic motion capture smartphone app (combined with a laptop) to assess FFT based on predefined measures (taping as fast as possible, performing 30 seconds of alternating pro- and supination cycles as fast as possible while holding a phone in the testing hand) for Huntington Disease (HD) patients. Muhamed, Siti Anizah, et al. in [185] adapt Polhemus Patriot Electromagnetic (EM) tracking sensors (hardware) for FFT (tracking performance of ten finger taps as fast and as wide as possible for a number of cycles) in PD. Authors Milica, et al. in [186] place special miniature inertial sensors on fingertips and a force sensor for FFT task. Force sensor is adapted to measure the contact force between the fingers. Milica, et al. in work [186] require a specific environment which uses a wireless transmit and a remote computer through the interface unit (PD patients were involved). The authors Suzumura, Shota, et al. in [187] use a custom smart device (terminal) with magnetic sensors to evaluate FFT based on rhythmic tapping (left or right hand, both hands) and sounded rhythmic tapping. The research involves AD and MCI patients. Janzen, et al. in [188] apply FFT for auditory stimulus tracking in PD patients, i.e., using an armless chair and a recording device (contact plate and a metallic probe attached to the subject's index finger with medical tap) combined with special software for detecting hand synchrony as compared to a metronome.

Table 2.3 provides an overview of related work for tracking finger tapping (FFT) impairments to central nervous system disorders.

Table 2.3. Comparison of related work findings for FFT tasks

Work ref.	Target group	Use of special equipment (custom hardware or software)	Special environment for experiment	Use mobile technology
[180]	PD, MCI, AD	Yes (light beam diode)	No	No
[181]	ASD	Yes (magnetic detection system with a PC	Yes	No
[182]	PD	Yes (music box)	No	No
[183]	PD	Yes (special gloves for touchscreen laptop)	Yes	No
[184]	HD	No	No	Yes
[185]	PD	Yes (tracking sensors)	No	No
[186]	PD	Yes (inertial, force sensors)	Yes	No
[187]	MCI, AD	Yes (terminal with magnetic sensors)	No	Yes

[188]	PD	Yes (armless chair and metal contact plate)	Yes	No

SCF is a very widely used state of the art technique for detecting changes in handwriting movements of patients with central nervous system disorders (CNSD) [56], [81], [82], [83]. The changes in handwriting can be seen as an early marker for PD, but it often suffers from poor sensitivity and specificity due to the inter-subject variations [56]. The graphical tasks, such as shape tracing, writing and a Fitts' task, are valid to assess the upper limb function in PD patients by using movement time and the presence of tremor. The research by authors E. J. Smits, et al. in [57] showed significant correlation of movement time on the graphical tasks, while successful medication can decrease the movement time and tremor power, thus validating the use of these measures for HD and PD diagnostics. However, the features that were obtained from sketching the Archimedean guided spiral on the screen can be used for discriminating between the PD patients and healthy people [56], [58], as people with PD produce less unstable patterns.

Researchers Passos, Leandro A., et al. in [189], Pham, Hung N., et al. in [191], Gupta, Deepak, et al. in [192], Pereira, Clayton R., et al. in [193] and Sharma, Prerna, et al. in [195] adapt the already existing *HandPD* spiral database [200], which is composed of handwritten exams, i.e., spiral drawing images for PD identification. Poon, Christine, et al. in [190] target the impaired handwriting (dysgraphia) for PD patients by conducting spiral drawing tests with tablets. The authors Poon, Christine, et al. in [190] consider graphic (stroke length, height, width, duration) and kinematic (speed, velocity, acceleration, jerk, grip angle, in-air time, contact time, change in pressure) features of the evaluation. Additionally, the researchers Pereira, Clayton R., et al. in [193] consider images as well as signals from smart pens, which analyse activity of movement for quantifying the normal motor activity in a healthy individual, as well as the dysfunction of PD patients. The spiral trajectory drawing test was chosen for the differentiation between PD and essential tremor [194]. In authors Chen, Kai-Hsiang, et al. work [194], Wacom tablets (together with special computer software for data capture) were used for the proposed task (tracing a given spiral, following a guided point) and velocity parameters as evaluators. In authors Danna, Jérémy, et al. research [196], Archimedean spiral task for PD was executed by using an ink pen on a sheet of paper (A4 format), affixed to a graphic tablet. The instruction was to draw spiral according to the defined temporal (spontaneous vs. as quick as possible) spatial (small vs. big) and handedness (dominant vs. non-dominant hand) conditions.

The researchers Bernardo, Lucas S., et al. in [197] developed a special interface (adapted for desktop in MS Windows operating system) for PD patients to conduct spiral drawing task and evaluate it by using Euclidean distance, relative distance, circular distance, Manhattan distance, pixel similarity, design speed, design time features.

The authors Moetesum, Momina, et al. in [198] approach adapts Parkinson's Disease Handwriting Database' (*PaHaW*) which contains (*x, y*) coordinates of the pen trajectory (spiral contour) as well as the pen status (whether touching surface or not), which is afterwards transformed into an image. DelMastro, Heather M., et al. in [199] examine the characteristics for understanding the relationship between tremor and patients with multiple sclerosis (MP) based on the digital spiral drawing task. The measures include the following: Segment Rate (SEGRT), Standard Deviation (SD) of Radial Velocity (VSD-R), SD of Tangential Velocity (VSD-T), SD of Overall Velocity (VSD-O), Mean Drawing Velocity (MNV-O) and Mean Pen Pressure Acceleration (MNA-P). Data in [199] research was collected with a tool, approved by International Classification of Functioning, Disability and Health (ICF) model.

The author notes that the majority of reference literature resources in this chapter as well involve the usage of classification algorithms and their fusion combinations (in most cases for determining between sick and healthy control subjects). Refer to Table 2.4 for comparison of SCF task related works.

2.2.3. State of the art research findings to detect cognitive impairments

The findings of researchers Craufurd, David, et al. in [201] confirm that behavioural problems are common among the patients with Huntington Disease (HD). Mild cognitive impairment as a prodromal period is a valid concept in pre-HD, with nearly 40% of individuals, showing this level of impairment before diagnosis [202]. The most common symptoms were loss of energy and initiative, poor perseverance and quality of work, impaired judgment, poor self-care and emotional blunting. The affective symptoms such as depression, anxiety and irritability occurred in about half of the patients that were studied, contributing to the functional capacity [203]. The patients with Juvenile Huntington Disease can show disease symptoms through nonspecific features, mostly psychiatric and cognitive difficulties. This can lead to misdiagnosis or diagnosis delay, especially in cases without a familial history of HD [204].

Table 2.4. Comparison of related work findings for Archimedean Spiral tasks (- indicates that no information was provided)

Work ref.	Target group	Targeted impairment	Evaluation metrics	Use existing dataset	Use mobile technology	Classification accuracy % (method)
[56]	PD	impaired handwriting (dysgraphia)	pen tip pressure, finger direction changes (x, y), grip angle, logarithmic value of distance travelled divided by angle	No	No (C# for desktop)	83.2 (Naive Bayes)
[81]		bradykinesia, dyskinesia	drawing speed, skewness of speed, radial velocity, angular velocity		No (telemetry device)	84 (multilayer perceptron)
[82]		essential tremor	spiral smoothness, speed/direction, power spectrum density analysis, asymmetry in spiral		Yes (Android app)	90.2 (logistic regression)
[83]		tremor, rigidity, bradykinesia and postural instability	average radial error, standard deviation of the radial error, maximum radial error			-
[189]		dysgraphia	images (pixel matrixes)	Yes (HandPD)	No	96 (optimum-path forest)
[190]			graphic (stroke length, height, width, duration) and kinematic (speed, velocity, acceleration, jerk, grip angle, in-air time, contact time, change in pressure)		Partly (WACOM tablets and MATLAB software)	90 (logistic regression)
[191]			x, y, z position of the screen, pressure of digital pen, grip angle, system time sample	Yes (UCI repository)	Yes (mobile app)	99.6 (k-NN)
[192]		dysgraphia	images (pixel matrixes)	Yes (HandPD)	No	94 (tuned cuttlefish algorithm)

[193]				Yes (HandPD) + author contribution	Partly (smart pens)	95 (CNN)
[194]	PD, ET	tremor	velocity parameters	No		-
[195]	PD	symptoms at premature PD stage	images (pixel matrixes)	Yes (HandPD)	No (Python software)	94.83 (Modified Grey Wolf Optimization)
[196]		motor, dysgraphia	temporal, spatial, handedness	No	Partly (ink pen on a sheet of paper (A4 format), affixed to a graphic tablet)	-
[197]			feature vector extracted from images (Euclidean distance, relative distance, circular distance, Manhattan distance, pixel similarity, design speed)		No (interfaces for MS Windows)	100 (SVM)
[198]		impaired handwriting	images (pixel matrixes)	Yes (HandPD)	No	83 (SVM)
[199]	MP	tremor	segment rate (SEGRT) from images	No	Partly (digital tablet + MATLAB tool for processing)	overall accuracy of proposed model not provided

The psychotic symptoms (hallucinations and delusions) were rarely reported. The factor analysis distinguished three clusters of behavioural symptoms, which were interpreted respectively as reflecting Apathy, Depression and Irritability. The 'Apathy' factor was highly correlated with duration of illness, whereas no such relationship was observed for the `Depression' and `Irritability' factors. The usage of assistive technology for cognition (external devices aimed at supporting cognitive function) can ameliorate negative effects of cognitive impairment and improve Health-related Quality of Life; though, this is not yet proven in the case of HD patients [205]. Six factors to note were identified by researchers: (1) speed/inhibition, (2) verbal working memory, (3) motor planning/speed, (4) attention–information integration, (5) sensory–perceptual processing and (6) verbal learning/memory. The factor scores were sensitive to the worsening of cognitive functioning in pre HD, typically more than performances on individual tests comprising factors. Only the motor planning/speed and sensory–perceptual processing factors predicted time to diagnosis in the case of study [206], suggesting that motor planning/speed and sensory–perceptual processing are important markers of disease prognosis and showing the need of the combined approach of early diagnosis systems.

The authors Lunven, Marine, et al. in [207] introduce an investigation to explore the capacity of digitalised arithmetic task to highlight cognitive decline in HD patients within a year.

As for the state of the art research, the computerized systems were adapted for providing cognitive rehabilitation support in Parkinson Disease (PD) patients. In authors Alloni, Anna, et al. work [208], the ontology based system (supervised by a therapist) is proposed to conduct exercises (finding a category, exclude a word from the list, select necessary elements for executing an action etc.), leveraging on a stimuli set (words, sounds and images).

The cognitive deficits in PD are targeted by authors Flannery, Samuel L., et al. in [209] where data was collected via an external two-button box and pre-processed with special software. The computerized tasks include selecting two consonants displayed on left and right, measuring stimulus response to a letter in the centre of the screen (with a distractor), following the sequence of flashcards.

Personal computer-based cognitive training in PD is presented by authors Maggio, Maria G., et al. in [210], based on combination of cognitive training programs and screening tools which consist of a series of specific exercises aimed at improving cognitive domains (i.e., attention, memory, spatial cognition and verbal and non-verbal executive functions).

The researchers Wu, Jia-Yun, et al. in [211] propose a computerized cognitive assessment system (composed of testing the orientation, attention, calculation, naming, language, memory, executive function and visuospatial abilities), aimed for dementia patients in the mobile game form.

The monitoring of Dementia patients was investigated by authors Wiloth, Stefanie, et al. in [212] by using a comprehensive motor-cognitive intervention program and special device (hardware) Physiomat for training. This program includes tasks for balance control with cueing instructions, following cursor from the centre of the screen directly to the targets highlighted as a moving yellow ball as fast as possible.

The authors Betrouni, Nacim, et al. in [213] use electroencephalogram (EEG) data (taken from previous cross-sectional observational study) and spectral power analysis by using a fast Fourier transform with a 2-second duration (50% overlap) features to classify PD patients into five different groups, according to the severity of their cognitive impairments. Refer to Table 2.5 for comparison of related work that is analysed for cognitive impairments.

2.2.4. State of the art research findings to detect speech impairments

The methodology for speech analysis could be adapted from [92], i.e., using OpenSMILE features, Essentia descriptors, MPEG7 descriptors, KTU, jAudio, YAAFE, Tsanas features and Random Forest (RF) supervised learning algorithm (method) to detect PD and fuse information in the form of soft decisions, obtained by using various audio feature sets from separate modalities.

Cepstral Separation Difference (CSD) for the quantification of speech impairment in Parkinson's disease could be applied as applied in [93]. The feature extraction by using signal-to-noise ratio (SNR), harmonic-to-noise ratio (HNR), glottal to noise excitation (GNE), vocal fold excitation ratio (VFER) and empirical mode decomposition excitation ratio (EMD-ER) methods with Random (RF) Forests and support vector machines (SVM) for classification algorithms were used in [94].

The alternative approaches could be tested as introduced by authors in An G., et al. in [95], i.e., Syllable-level Features, Low-Level Descriptor (LLD) Features, Formant Features, Phonotactic Features with SVM classifier and features extracted by using principal Component Analysis (PCA), and [96]: Linear Discriminant Analysis (LDA), SVM, Adaptive Boosting (AdaBoost), K-Nearest Neighbour (KNN) and Adaptive Resonance Theory-Kohonen Neural Network (ART-KNN) classifiers.

Furthermore, the extensive state of the art studies is being conducted in detection of speech impairments in central nervous system disorder patients (CNSD). The researchers Gillivan-Murphy, Patricia, et al. in [214] research use voice recordings (collected in sound-treated laboratory with ambient noise levels measured at 50 dB sound pressure level by using a AKG-C420 head mounted microphone) to detect speech tremors in Parkinson Disease (PD). In [214], the acoustic analysis was performed with a special Voice and Tremor Protocol (VTP), i.e., adapts rate, periodicity, magnitude of frequency and amplitude of voice signal features.

Table 2.5. Comparison of related work findings for cognitive impairment tasks (- indicates that no information was provided)

Work ref.	Target group	Methodology	Dataset	Evaluation metrics	Type of system	Supervision of experiment	Statistical analysis (methods)
[207]	HD	-	34 HD and 23 controls (author contribution)	Response time (RT) and accuracy	Unknown (psychometry device)	Not mentioned	regression analysis $p< 0.05$ (linear mixed models)
[208]	PD	MME, MoCA	31 PD, creation of 2 databases (author contribution)	Find a category, exclude a word from list, select necessary elements for executing an action	Ontology (PC)	Yes	-
[209]		MoCA + additional (SAGE)	37 PD, 47 controls (author contribution)	Measure stimulus response with a letter in the centre of the screen (with a distractor), following a sequence of flashcard	Two-button box (hardware), Matlab software	Not mentioned	regression analysis $p< 0.05$
[210]		MoCA + additional (SAGE)	1 PD (author contribution)	Exercises for testing patient attention, memory, spatial cognition and verbal and non-verbal executive functions	PC (special software)	Yes	-

[211]	Dementia	MME, MoCA + additional (SAGE)	5 dementia and 5 control subjects (author contribution)	Attention, language, memory, visuospatial abilities, executive function and orientation	PC (Windows) and a mobile app (Android), gamification	Not mentioned	-
[212]		-	56 dementia, 43 control subjects (author contribution)	Temporal (duration: total test time in seconds) and spatial (accuracy: sway path in digits/ms	Comprehensive motor-cognitive intervention program, combined special device (hardware), Physiomat for training	Yes	regression analysis $p< 0.01$
[213]	PD	-	118 PD patients (no author contribution)	Electroencephalogram (EEG), spectral power analysis	Waveguard EEG recorder + special software for signal processing	Yes (neuropsychological assessment procedure)	88% classification accuracy (k-NN)

The authors Gaballah Amr, et al. in [215] investigate subjective and objective assessment of PD speech quality. The analysed features derived from the speech recordings (collected with seven different amplification devices) based on temporal, spectral, and/or cepstral parametrization (mel-frequency cepstral coefficients (MFCC) [112], discrete cosine transform (DCT) [247], gammatone frequency cepstral coefficients (GTCC) [112], [113], [114], speech-to-reverberation masking ration (SRMR) [248], modulation area (ModA) [249], Low Complexity Quality Assessment (LCQA)) [250]. Support vector regression, Gaussian process regression and deep learning methods were used for the correlation analysis (0.85 achieved accuracy).

The identification of distinctive acoustic and spectral features in Parkinson's disease is analysed in [216] with data recording at 44.1 kHz, 16bits per sample by using the same microphone. MFCC, linear prediction coefficients (LPC) [251], discrete wavelet transform (DWT) [251], Gaussian mixture model (GMM) [252], time domain entropy (ET) [253], spectral entropy (ES) [254] features were used for the evaluation (77.2% accuracy was achieved by using SVM algorithm with linear kernel).

The authors Wu, Kebin, et al. in [217] target learning acoustic features (MFCC, spherical K-means, pooling method) to detect PD. All data was collected in a soundproof room and then resampled at a sampling rate of 16 KHz. Random Forest (RF) and SVM methods were used for the evaluation of detection accuracy (best achieved result is 96.37% with RF classifier).

Perez, Matthew, et al. [218] study involves Huntington Disease (HD) patients for solving binary classification problem (differentiate between healthy controls and HD) based on acoustic and lexical features (MFCC, GMM, pause, speech rate, goodness of Pronunciation (GoP) [259]). This research uses data from the already carried out study at the University of Michigan. The results were evaluated with k-Nearest Neighbours (k-NN) and Long-Short-Term Memory Recurrent Neural Networks (LSTM-RNN) algorithms (0.87 best achieved correlation).

The researchers Sakar, C. Okan, et al. in [219] provide a comparative analysis of speech signal processing algorithms for PD classification. The approach described in [219] uses recurrence period density entropy (RPDE), detrended fluctuation analysis (DFA), pitch period entropy (PPE), MFCCs, wavelet transform (WT) methods for feature extraction [255], [256], [257], [258]. The used data was collected from the Department of Neurology in Cerrahpasa Faculty of Medicine, Istanbul University. The results were validated with a set of supervised learning classifiers (SVMs with linear and RBF kernels, Multilayer Perceptron, Naive Bayes, Logistic Regression, Random Forest and k-NN algorithms, ensemble approach) (0.86 best achieved correlation).

The classification of PD severity is introduced by Oung, Qi Wei, et al. in [220]. The data for speech signals were recorded by using a headset (Sennheiser DW Pro2) positioned in 5 cm distance from the subject's mouth. The researchers for

feature extraction in speech signals adapt these methods: wavelet energy (WE), Shannon wavelet entropy (ShWE), Renyi wavelet entropy (ReWe), Tsallis wavelet entropy (TsWe), permutation entropy (Pe) and fuzzy entropy (Fe). The classifiers used in the research were K-nearest neighbour (KNN), probabilistic neural network (PNN) and extreme learning machine (ELM), (best achieved results were 91.11%).

The authors Ali, Humair, et al. in [221] use Parkinson speech-based data set to investigate the classification of early diagnosis of PD. The data was adapted from the UCI machine learning repository, and 15 acoustic features were considered in research: jitter, number of pulses, number of periods, mean period, standard deviation of period, number of voice breaks, degree of voice breaks, mean pitch, standard deviation, minimum pitch, autocorrelation, noise-to-harmonic ratio and harmonic-to-noise ratio. Four classifiers were examined: Bayes Net, Random Forests, Decision Stump and SVM (95.6% best accuracy results).

The fusion of wavelet packet transform (WPT) and MFCC methods were applied for the diagnosis of PD based on the recorded speech signal by using Hidden Markov Models (HMM) and SVM classifiers [222]. The achieved accuracy results are not provided by the authors.

The authors Burk, Brittany R., et al. in [223] analysed acoustic recordings (special software and hardware were used for the data collection) from PD patients based on the cepstral peak prominence (CPP) and aerodynamic measures of transglottal airflow (TAF) features. The goal of [223] study was to distinguish between speakers with no tremor and tremor (correlation analysis 0.96).

Jeancolas, Laetitia, et al. in [224] target vocal impairment detection for early prediction in PD. The researchers applied Mel-Frequency Cepstral Coefficients (MFCC) and Gaussian Mixture Model (GMM) methods for feature extraction. The data (96 kHz audio samples) was collected with a professional head mounted omnidirectional condenser microphone that was placed by approximately 10 cm from the mouth of PD patient. The classification results were validated with bootstrap aggregation approach from log-likelihood on each frame (83% best achieved accuracy), refer to Table 2.6 for comparison of related works for speech impairments.

2.2.5. State of the art research findings to detect energy expenditure impairments

Energy Expenditure relates to the weight loss in Huntington disease (HD) and is directly linked to CAG repeat length and is likely to result from a hypermetabolic state [97]. HD patients often experience weight loss (including loss of muscle bulk), mainly due to the development of early negative energy balance, which in turn may cause weight loss with loss of muscle bulk in particular [98]. It may reflect a catabolic state secondary to hypothalamic pathology.

Table 2.6. Comparison of related work findings for speech impairment tasks

Work ref.	Target group	Dataset	Evaluation metrics (features)	Hardware	Software	Audio samples	Experiment supervision	Statistical analysis (methods)
[92]	PD	64 PD subjects, 34 healthy controls (author contr.)	Global, long-term or high-level descriptors, local, short-term or low-level descriptors (LLDs)	2 devices: acoustic cardioid and smartphone	Praat	.wav (16 bits, 44.1 kHz)	No	equal error rate (EER) = 19.27 (fusion of classifiers)
[93]	PD	60 PD and 20 controls (acquired from the previous study)	Cepstral Separation Difference (CSD), harmonic ratio, harmonic-to-noise ratio and Mel-frequency cepstral coefficients	Custom device for at-home testing purposes	Analysis of variance (ANOVA)	16 bits, 48 kHz	No	correlation analysis (0.78)
[94]	PD	14 PD subjects (author contr.)	Dysphonia (signal-to-noise ratio (SNR), harmonic-to-noise ratio (HNR), detrended fluctuation analysis (DFA) etc.), wavelets (*Fotime* series), MFCC	Head-mounted microphone	Audacity	16 bits, 44 kHz	Yes	90% accuracy (SVM)
[95]	PD	acquired from Interspeech Corpus	Syllable-level Features, Low-Level Descriptors (LLD), Formant Features, i-Vectors	Not provided	ALIZE	Not provided	No	regression (SMO), Spearman correlation coefficient = 0.0156

[96]	PD	23 PD and 8 controls (acquired from UCI repository)	Pitch period entropy (PPE), harmonic-to-noise ratio (HNR), detrended fluctuation analysis (DFA), jitter, average vocal fundamental frequency etc.	Not provided	Not provided	Not provided	No	79.17% accuracy (SVM)
[214]	PD	30 PD and 28 controls (author contr.)	Acoustic analysis was performed with a special Voice and Tremor Protocol (VTP), i.e., adapts rate, periodicity, magnitude of frequency and amplitude of voice signal features	AKG-C420 head-mounted microphone	SPSS	50 kHz (Kay data file)	No	regression analysis ($p < 0.05$) on acoustic features
[215]	PD	10 PD and 10 healthy controls (author contr.)	MFCC, DCT, GTCC, speech-to-reverberation masking ration (SRMR), modulation area (ModA), Low Complexity Quality Assessment (LCQA)	7 different amplification devices	Malcolm Slaney's auditory toolbox	16 bits, 16 kHz	Yes	correlation analysis (0.85)

[216]	PD	12 PD and 12 healthy controls (author contr.)	MFCC, linear prediction coefficients (LPC), discrete wavelet transform (DWT), Gaussian mixture model (GMM), time domain entropy (ET), spectral entropy (ES)	Microphone	Not provided	16 bits, 44 kHz	Yes	77.2% accuracy (SVM with linear kernel)
[217]	PD	27 PD and 446 healthy controls (author contr.)	MFCC, spherical K-means, pooling method	Soundproof room (equipment details not provided)	Matlab (Voice Analysis Toolbox)	16 kHz	Yes	96.37% (Random Forest) classifier
[218]	HD	31 HD and 31 healthy controls (acquired from the previous study)	Acoustic and lexical features, MFCC, GMM, pause, speech rate, goodness of pronunciation (GoP)	Not provided	Computerized Language Analysis (CLAN)	Not provided	Yes	correlation analysis (0.87)
[219]	PD	188 PD and 64 healthy controls (author contr.)	Recurrence period density entropy (RPDE), detrended fluctuation analysis (DFA), pitch period entropy (PPE), MFCCs, wavelet transform (WT)	Microphone	Praat	44.1 kHz	No	correlation analysis (0.86)

[220]	PD	65 PD (author contr.)	Wavelet energy (WE), Shannon wavelet entropy (ShWE), entropy (Pe) and fuzzy entropy (Fe)	Headset (Sennheiser DW Pro2)	Matlab, Simulink	16 bits, 44 kHz	Yes	91.11% (extreme learning machine)
[221]	PD	20 PD and 20 healthy controls (acquired from UCI repository)	Period, number of voice breaks, degree of voice breaks, mean pitch, standard deviation, minimum pitch	Not provided	Audacity, Praat and Weka	Not provided	No	97.6% (Random Forest)
[222]	PD	author contribution	Wavelet packet transform (WPT) and MFCC	Not provided	Not provided	Not provided	Not provided	not provided
[223]	PD	34 PD and 11 healthy controls (author contr.)	Cepstral peak prominence (CPP) and aerodynamic measures	Head-mounted condenser microphone	SPSS, analysis of Dysphonia in Speech and Voice (ADSV)	144 kHz	Yes	correlation analysis (0.96)
[224]	PD	75 PD and 54 healthy controls (author contr.)	MFCC and Gaussian Mixture Model (GMM)	Professional head mounted microphone	Praat	24 bits, 96 kHz	Yes	83%, bootstrap aggregation

According to the research [99], the energy expenditure was 11% higher in the HD subjects than in the control subjects. Similarly, in the research by authors Richard E. Pratley, et al. [100], 24 hour energy expenditure was 14% higher in HD patients than controls in absolute terms and after the adjustment to age, sex, fat mass and fat-free mass; however, the total free-living energy expenditure was smaller, because patients with HD appeared to engage in less voluntary physical activity. This difference was due to a higher waking metabolic rate [101], which was related to a significantly greater displacement of the centre of mass by HD subjects than control subjects. Weight loss was observed in the HD group even in presymptomatic carriers, although their caloric intake was higher than that of controls [102].

There are found only a few state of the art research resources in computer science field for tracking energy expenditure impairments for central nervous system disorder (CNSD) patients. The solution by researchers Cersosimo, Maria G., et al. in [225] propose a multivariable logistic regression analysis to determine the association between weight loss and PD motor manifestations by measuring differences of their body weight over the time.

Verbeken, Sandra, et al. in [226] introduced computerized tests, which display pictures of food items to display healthy and unhealthy food items in order to suggest a proper calorie balance for a daily food ration.

The authors Singh, Balbir, and Hissam Tawfik in [227] provide a machine learning (support vector machines, artificial neural networks) approach to predict weight gain risks in young adults (as well applicable for CNSD patients). In particular, body mass index (BMI) values are measured from Millennium Cohort Study (MCS) study data.

The author notes that energy expenditure impairments can as well be tracked with the use of mobile applications for calculating daily gained and burned calories (referenced in the next section). Refer to Table 2.7 for comparison of related works for energy expenditure impairments.

2.2.6. Mobile applications to detect early symptoms in CNSD patients

The mobile applications have already been developed for monitoring and assessing the symptoms of HD and PD; however, they usually lack scientific background in their methods and the assessment of user's health state [68].

ParkNosis app uses a simple touch screen and motion tests on Android smartphone for objectively measuring PD symptoms [69]. However, it was tested only on healthy subjects rather than HD patients. A smartphone application quantitatively measures hand dexterity and provides the assessment of overall motor function [70].

Table 2.7. Comparison of related work findings for energy expenditure tasks

Work ref.	Target group	Dataset	System type	Features	Experiment supervision	Statistics (method)
[225]	PD	144 PD and 120 controls (author contribution)	SPSS software for statistical analysis (PC application)	Body Weight (BW)	Yes	regression analysis ($p < 0.001$)
[227]	Youngsters with obesity (rehabilitation)	46 youngsters (author contribution)	ANOVA software for statistical analysis (PC application)	Body mass index (BMI), executive functioning	Yes	regression analysis ($p < 0.05$)
[231]	Students-volunteers	Empty (prototype)	Mobile application (Android OS)	Nutrition: diet plan with calendar, training plan with calendar	No	Not investigated
[232]	People, willing to improve eating habits	Not provided		BMI, BMR, TDEE		
[233]	Diabetic patients	Images (ETHZ Food-101 dataset)		Image pixels (matrix)		70% (CNN)

The majority of apps that were supported by clinical trials used a combination of smartphone and wearable sensors and included the assessment of subject's gait or posture abnormalities [71]. In addition, mobile app *Tremor12* (*iPhone*) that samples acceleration, rotation, rotation speed and gravity for essential tremor in PD is referenced in [228]. Pan, Di, et al. in [229] propose a mobile app for the Android platform to collect PD-related motion data by using the smartphone 3D accelerometer (data is used for the PD symptom severity estimation). Three tests are supported: hand tremor, walking and turning (with legs). SVM classifier was used for the system evaluation (82% accuracy). Kuosmanen, Elina, et al. in [230] presents a mobile application (both on Android and on *iOS*) named *STOP* for screening PD symptoms and medication intakes. The monitoring process involves a ball game that quantifies patient's motor symptoms and a medication journal where PD patient log their medication intake time.

The computerized approach of energy expenditure impairments is presented in [231], where mobile application *CoviHealth* (Android OS) is proposed for controlling promoted healthy dietary habits and physical activity that is based on anthropometric parameters and gamification. Furthermore, the authors Tirasirichai, Benjarat, et al. in [232] introduce calorie balancing Android mobile application *Bloom Balance* for people who are willing to improve their health by controlling eating habits and doing more exercises (adaptable for CNSD patients). The app provides users with health profiles (*BMI* (body mass index), *BMR* (basal metabolic rate) and *TDEE* (total daily energy expenditure)) to suggest calorie balance plan (gained and burned measures). The research by authors Merchant, Kaiz, and Yash Pande in [233] describes an Android application (integrated with convolutional neural network (CNN) for image recognition that achieved 70% accuracy) to predict nutritional estimates to patients with eating disorders and recommending alternative food recipes. State of the art research confirms a more systematic approach to analyse mobile apps for tracking impairments of CNSD patients, i.e., by conducting summarised reviews of already existing solutions. Cohen, Shani, et al. in [234] provide an overview of mobile apps that are used for monitoring remote digital trials of Parkinson disease (PD) and Huntington disease (HD).

According to the authors of the research [234], such trials are related to quantifying symptoms of PD [235], as well as for home-based monitoring in PD [236], [237], [238], [239], [240] and [241]. Fewer smartphone based mobile solutions (as compared to PD) exist for HD, as it is naturally considered to be more challenging to recruit. In this context, the related studies are primarily focused on the feasibility or HD symptom quantification over several days [242], [243], [244].

In 2018, researchers Linares-Del Rey, et al. [245] investigated the mobile applications related to PD. The searches concluded with a total of 103 apps that were found (49 are available only for Apple devices, 39 for Android devices and 6 for Windows Phone devices, 8 are available for both Apple and Android devices, and one is available for all 3 operating systems). Seventy-four apps are free of charge,

and 29 are paid; 45 target patients with PD, 34 target healthcare professionals, and 24 target both groups. Please refer to table two in [245] for the completed list of reviewed apps, including prices. A recent study in 2019 identified 92 (Android and *iOS*) mobile applications for providing support to PD, based on symptom assessment (tremors, speech) and usability (visibility of system status, match between system and the real world, user control etc.) metrics [246]. Please refer to Table E.1 in Appendice E for comparison of the analysed mobile apps.

2.3. Review summary

Approximately 142 related work literature references were analysed in this section of the dissertation. The majority of these resources introduce the findings of investigations that are no older than two years. Such factors indicate high popularity, perspectives and potential of the proposed topic in this dissertation, i.e., neural impairment detection for patients, suffering from neurodegenerative disorders (Huntington Disease (HD), Parkinson Disease (PD), Alzheimer Disease (AD), Mild Cognitive Impairment (MC), dementia etc.). However, the existing solutions have limitations, which provide motivation to create a proposed method and system, based on these five conclusive points:

1. Analysis of tremor (motor) impairment related work considers finger tapping tasks (FTT) and spiral contour following (SCF) tasks. The FTT analysis study showed that none of state of the art findings were designed for HD patients. Moreover, the majority of FFT works require custom equipment (hardware, software) and special environment for conducting experiments (all of them require supervision). The minority of FFT solutions adapt mobile technology. In addition, SCF works mostly track handwriting impairments in PD. Half of the SCF solutions use an already existing datasets (e.g., *HandPD* for spiral images) in their research. The minority of SCF investigations use mobile Android apps for feature extraction, digital tablets or pens for data collection that are instead adapted. Statistically, the proposed models are evaluated with classification accuracy. Similarly, as in FFT, SCF related studies do not involve HD patients.
2. Analysis of cognitive impairment related work revealed that the majority of digital solutions are based on Mini-Mental State Examination (MMSE) or Montreal Cognitive Assessment (MoCA) methodologies. The investigations focus on the detection of memory loss (mainly for PD and dementia patients, only 1 research involved HD), based on attention, language, memory, visuospatial abilities, executive function and orientation tasks. The majority of computerized solutions adapt custom hardware and software (or combinations of them) in personal computer (PC) for the assessment of patient health state level. Most of these related works require supervision of the test execution. Statistically, the proposed related work models for cognitive impairment detection were evaluated by using regression analysis (best results were of *p*-value (tests the null hypothesis that the coefficient is

equal to zero) < 0.001) and classification methods (88% achieved with k-NN algorithm).

3. Speech impairments are very intensively analysed by the other computer scientists. The majority of related works involve PD patients. However, very different approaches are adapted for the collection of voice recordings, i.e., most solutions require custom hardware (special microphones, amplification devices or headsets) and audio signal processing software (*Matlab*, *Praat*, *Audacity*, *SPSS*) for test supervision. In addition, the evaluation metrics (features) that are used for detecting speech impairments cover a wide range of choices, i.e., from acoustic features (jitter, number of pulses, voice breaks etc.), Mel-Frequency Cepstral Coefficients (MFCC), Gaussian Mixture Model (GMM), spectrum (kurtosis, spread, entropy etc.), wavelet transforms (WT) to strategies for combining these features. Statistically, the proposed related work models for speech impairment detection were evaluated by using regression analysis (Spearman correlation coefficient = 0.0156) and classification methods (97.6% achieved with the Random Forest algorithm).
4. Analysis of energy expenditure impairment related work showed that very few computerized solutions (1 was found for PD patients) adapted for central nervous system disorders exist. None of the findings involve HD patients. The majority of existing approaches detect weight loss with body weight (BW) and body mass index (BMI) metrics. Half of works use Android mobile applications (other half PC based statistical analysis software (*ANOVA*, *SPSS*)) to evaluate nutrition and daily calorie balances. Moreover, the decision support making in the system, based on energy expenditure impairments, is rarely analysed.
5. Analysis of mobile applications, adapted for central nervous system disorder patients (CNSD), showed that the existing solutions have limitations: either applies different health state evaluation methodology or restricts the number of features extracted for the single test mode. In addition, none of the analysed PD detection mobile apps (92 are referenced from a survey conducted in 2019, Android, iOS or cross-platform) are designed for HD symptom tracking.

To sum up, the author of this dissertation presents a new approach for automated health state evaluation for CNSD patients by adapting mobile devices that do not require any extra sensors for data collection and a feature combination to track all four impairments (tremor (motor), cognitive, speech and energy expenditure). Each of these impairments has been proven (separately) to be significant indicators of early detection of neurodegenerative disorders; however, a combination of all of them has not been investigated. *SAGE* methodology (compared to *MoCA*) is adapted, as it was more recently designed and has an electronic model (e-*SAGE*), thus leaving space for researchers to improve (e.g., extend methodology) medically validated tool.

3. MATERIALS AND METHODS

For materials and methods, a dataset collection tool is proposed, based on self-administered cognitive testing methodology (SAGE), which is used to identify signs of mild cognitive impairment (MCI) and early dementia in Huntington disease (HD), Parkinson (PD), Alzheimer disease (AD) patients. SAGE [78], [176] is applied in practice (mostly in the USA) by medical practitioners-neurologists, i.e., submitting questionnaires to patients in paper form and self-assessing their condition manually. Please refer to Appendice F for a detailed description of the SAGE screening instrument.

In this chapter, the author presents an innovative approach of computerizing extended version of SAGE methodology by considering related research findings about tremor (motor), speech and energy expenditure impairments that are occurring to HD, PD and AD patients. The basic functionality, as defined in SAGE questionnaires, is included in the proposed system, i.e., all 12 tasks with the integration of 4 different group variants. All these indicators lead to improved patient health state monitoring by using automatic manner, i.e., without direct supervision of a doctor.

In addition, four groups of feature extraction methods are considered in this dissertation: tremor (motor), cognitive, speech, and energy expenditure. These groups of methods correspond to types of neural impairments that may occur to patients with central nervous system disorders (CNSD). Moreover, the targeted feature extraction methods have been shown to be significant factors for the identification of neurodegenerative diseases (e.g., HD, PD, AD), as referenced in the related work research findings section of this dissertation. Tremor (motor) and cognitive group include these methods: Euclidean Distance (measures the distance of two given points), Frechet distance for curve comparison, Jaro algorithm to compare string input symbol by symbol, a Neighbour matching algorithm for graph similarity measure, parallel line angle calculation of 3-D figure edges, finger motion tracking methods (e.g. velocity, acceleration). The analysis of speech impairments includes these methods: pitch, Mel-frequency cepstral coefficients (MFCC), gammatone cepstral coefficients (GTCC), Wavelet Scattering Transform (WST), spectral methods in frequency domain (slope, skewness, spread, flux, centroid, rolloff, decrease, flatness, kurtosis, entropy). The energy expenditure group consist of methods for the calculation of daily gained and burned calories by using Basal Metabolic Rate (BMR) metric and Mifflin St. Jeor equation [103].

The chapter is structured as follows: a brief overview of proposed health state evaluation system is provided in section 3.1. Next, a detailed mechanics of implemented methods for feature extraction are defined. Such mechanics include an illustration of computerized algorithms for each task (16 in total, 10 from SAGE methodology and 6 additional) as UML workflow diagram and formalization of

mathematical apparatus for the adapted methods. Finally, a summary of used materials and methods in this dissertation is described in section 3.3.

3.1. Proposed health state evaluation system for neural impairment tracking

The proposed health state evaluation system (Neural Impairment Test Suite, NITS) [18] is easily adaptable to various neurodegenerative diseases (HD, PD, AD) or dementia conditions (MCI). NITS is a mobile application, which serves as a smart interface-screening tool for patients with central nervous system disorders (CNSD). NITS system consist of six components. Four components are designed for evaluating targeted groups (tremor, cognitive, speech and energy expenditure) of impairments. The remaining two components are created for CNSD patients health state visual monitoring and modifying application options (e.g., changing username, language or sound). NITS is developed for Android OS with core supported software development kit (SDK), additionally including third party libraries and custom algorithms for the required functionality. The mobile app supports different screen sizes, i.e., is adaptable for mobile phones or tablets. A complete list of implemented tasks for the assessment of the patient health state is shown in Table 3.1. T5, T6, T7, T8, T9, T10, T11, T12, T13, T0 (referenced by a number of SAGE questionnaire) tasks are transferred from SAGE methodology, whereas T1, T2, T3, T4, T14, T15 tasks are extra (SAGE extension).

Table 3.1. List of all available tasks in proposed mobile app

No. (alias)	SAGE question no.	Task name	Impairment group
T1	Extension	Sequential Touch	Tremor, Cognitive
T2	Extension	Rainbow Colour Touch	Tremor, Cognitive
T3	Extension	Multi-Touch	Tremor, Cognitive
T4	Extension	Archimedean Spiral	Tremor, Cognitive
T5	Non-scored	Insights	Cognitive
T6	1	Orientation (current date)	Cognitive
T7	2	Picture Naming	Cognitive
T8	3, 4, 5	Similarities, Calculation	Cognitive
T9	7	Construction (3-D figure)	Cognitive, Tremor
T10	8	Construction (clock)	Cognitive, Tremor
T11	9	Verbal Fluency	Cognitive
T12	10	Executive: Modified Trials	Cognitive, Tremor
T13	11	Executive: Problem Solving	Cognitive, Tremor
T14	Extension	Voice Recorder	Speech

T15	Extension	Total Daily Energy Expenditure (TDEE)	Energy Expenditure
T0	6, 12	Memory	Cognitive

Two modes (training and testing) are available in NITS. In training mode, a patient is instructed to try out different tasks separately from the navigation drawer menu as an exercise. In testing mode, all tasks are integrated together by giving a single task only once and providing next task randomly with a shuffle algorithm. Such approach ensures that a patient does not memorize all the questions when repeated test attempts are taken. When the actual test finishes, the results may be presented to the patient (if asked). In addition, all extracted features are stored in mobile device storage, relational MySQL backend database and researcher's Google Drive account automatically. In case when no internet connection exists during the examination, but synchronization with Google Drive is turned on, the collected data is backed up and send immediately when internet becomes available. See Appendice B. for NITS mobile app illustrations and details. Refer to Appendice C for complete feature list with alias and names. Figure 3.1 illustrates the concept of NITS health state evaluation system.

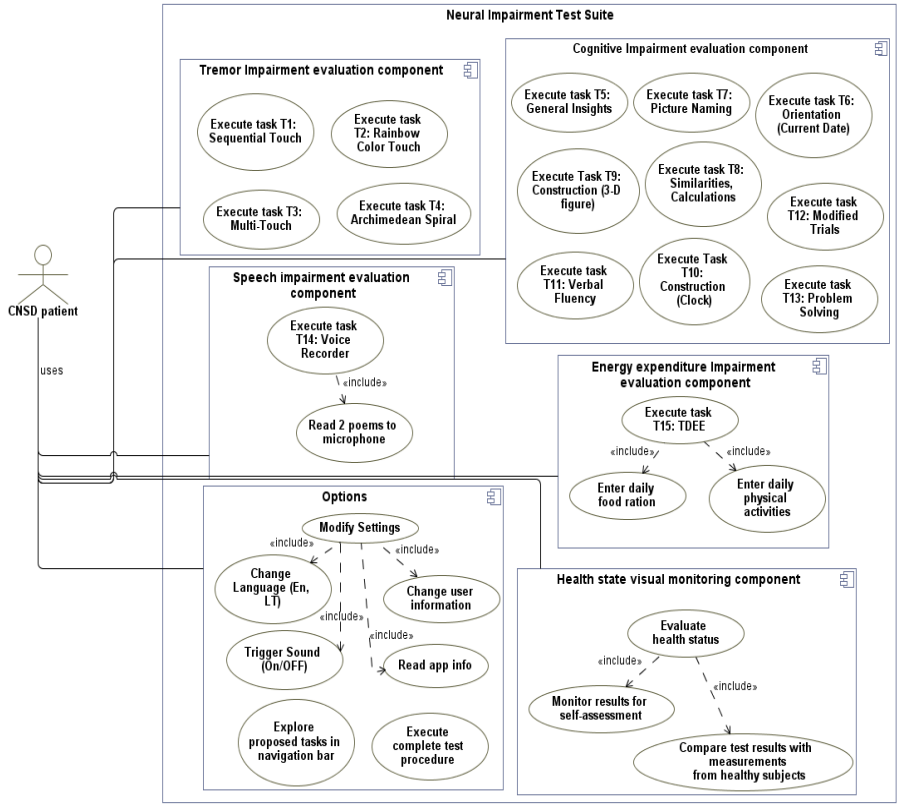


Figure 3.1 Schema of proposed health state evaluation system (NITS mobile app)

3.2. Algorithmic implementation of feature extraction for the proposed tasks

3.2.1. T1, T2, T3: Finger tapping tasks (FTT)

Three tasks are defined: T1 (Sequential Touch), T2 (Rainbow Colour Touch) and T3 (Multi-Touch). In these FTT tasks, a patient is instructed to touch circular shape objects by using his single finger (T1, T2) and multiple fingers (T3). The goal is to touch displayed objects as close to the centre and as quickly as possible. The assumption is that patients with neurological disorders will do the tasks slower and less precise than others. The bigger are the distances (delta) between the touched points, the stronger is hand tremor indicator, whereas higher response time indicates body stagnancy, while lower reaction time and smaller distances between original positions and touches show that the user is healthier.

Working principle of *T1*: circular shape objects (2, 3 and 5 circles at a time) are randomly generated of one particular colour on the mobile device screen. Each circle is located in different position of the screen, thus no possible collisions (overlapping) between two particular circles are possible. An active circle that needs to be touched is marked by a black contour to differ from the other objects.

Working principle of T2: circular shape objects of 7 different colours (taken from a spectrum of the rainbow) are randomly generated on the mobile device screen. No overlapping of objects is possible. In addition, a text label is shown on the screen to indicate which coloured circle needs to be touched. The screen is redrawn 5 times, i.e., test subject needs to touch the object for five times. Each time the coloured circle that needs to be touched is randomly selected and displayed in the text label. The main difference of T2, compared to T1, is that T2 attempts to provoke HD patient more by providing a greater level of uncertainty, which object needs to be touched.

Working principle of *T3*: in this test (tablet is preferred), circular shape objects are randomly generated by using three modes: 2, 3 and 6. No overlapping of objects is possible. In mode of 2 circles, the test subject needs to touch both of the circles by using multi-touch: it can be achieved with 2 fingers of 1 hand or 2 and using 1 finger. Afterwards, the fingers are released. The objects can be touched in any sequence that the test subject considers. In mode of 3 circles, 3 fingers need to be used and released afterwards. When 6 circles are displayed, 3 of which are painted in one red colour, another 3 in green, both hands need to be used: 3 fingers of the left hand and 3 fingers of the right. Each hand should touch only one particular colour. In 2 and 3 mode, the mobile device could be held in one hand; however, when 6 circles are displayed, the device should be put on a surface.

The author notes that a collision of circular shape object generation is possible when the total number of objects does not fit on the mobile device screen. In such cases, an error message occurs, indicating that the device screen is too small, and instructions are given to decrease the size of the circle radius.

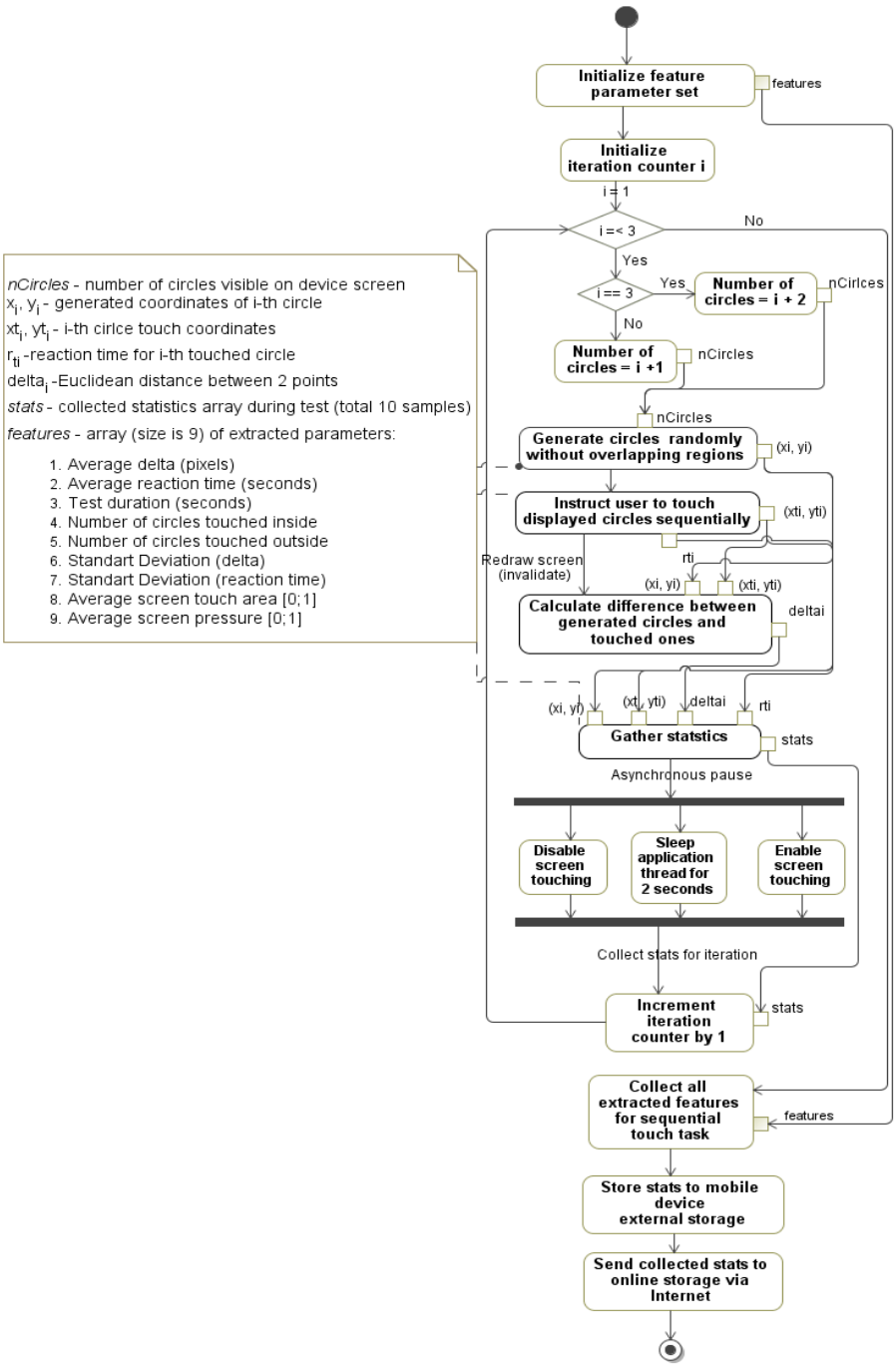


Figure 3.2. Algorithmic implementation of task T1: Sequential Touch

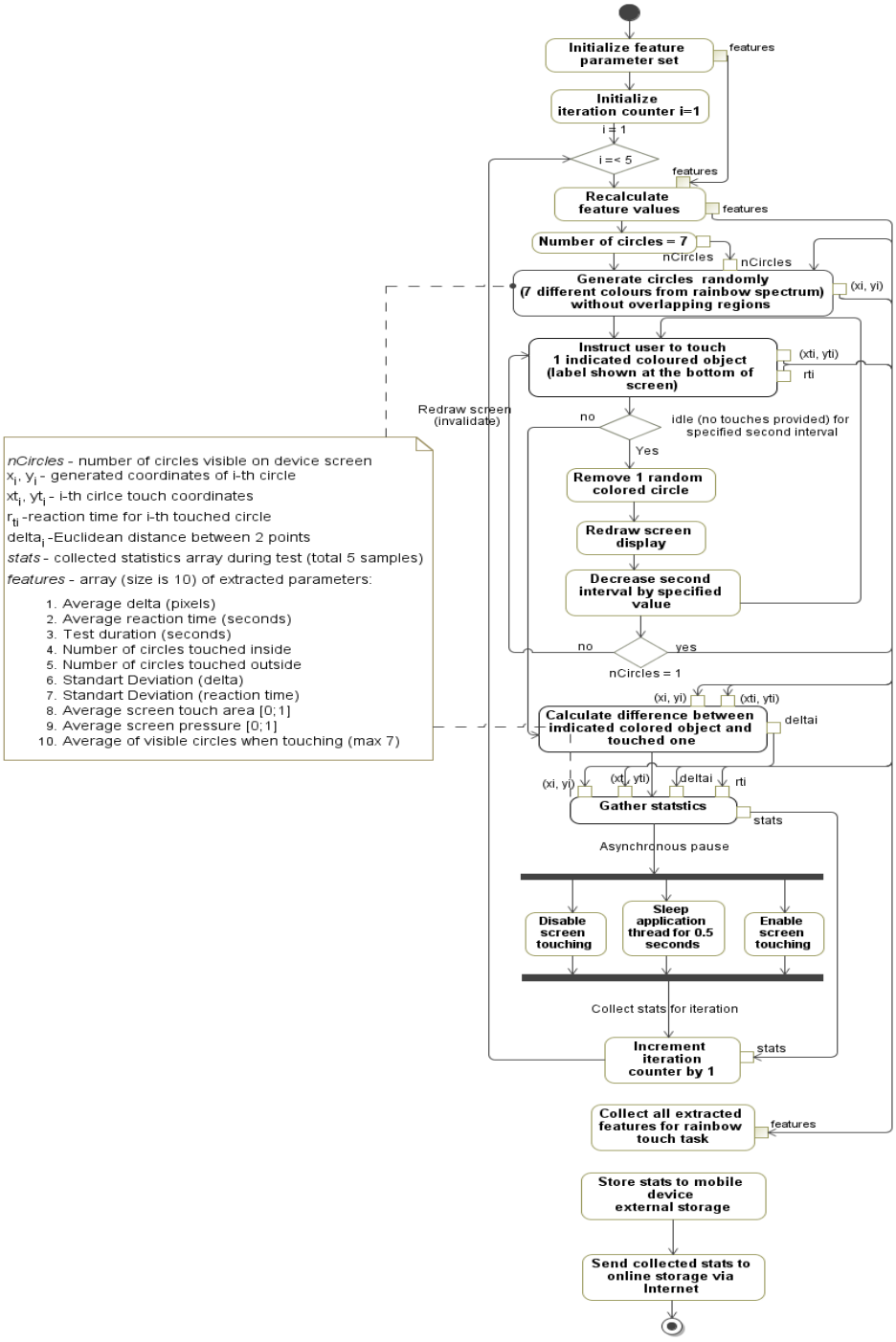


Figure 3.3. Algorithmic implementation of task T2: Rainbow Colour Touch

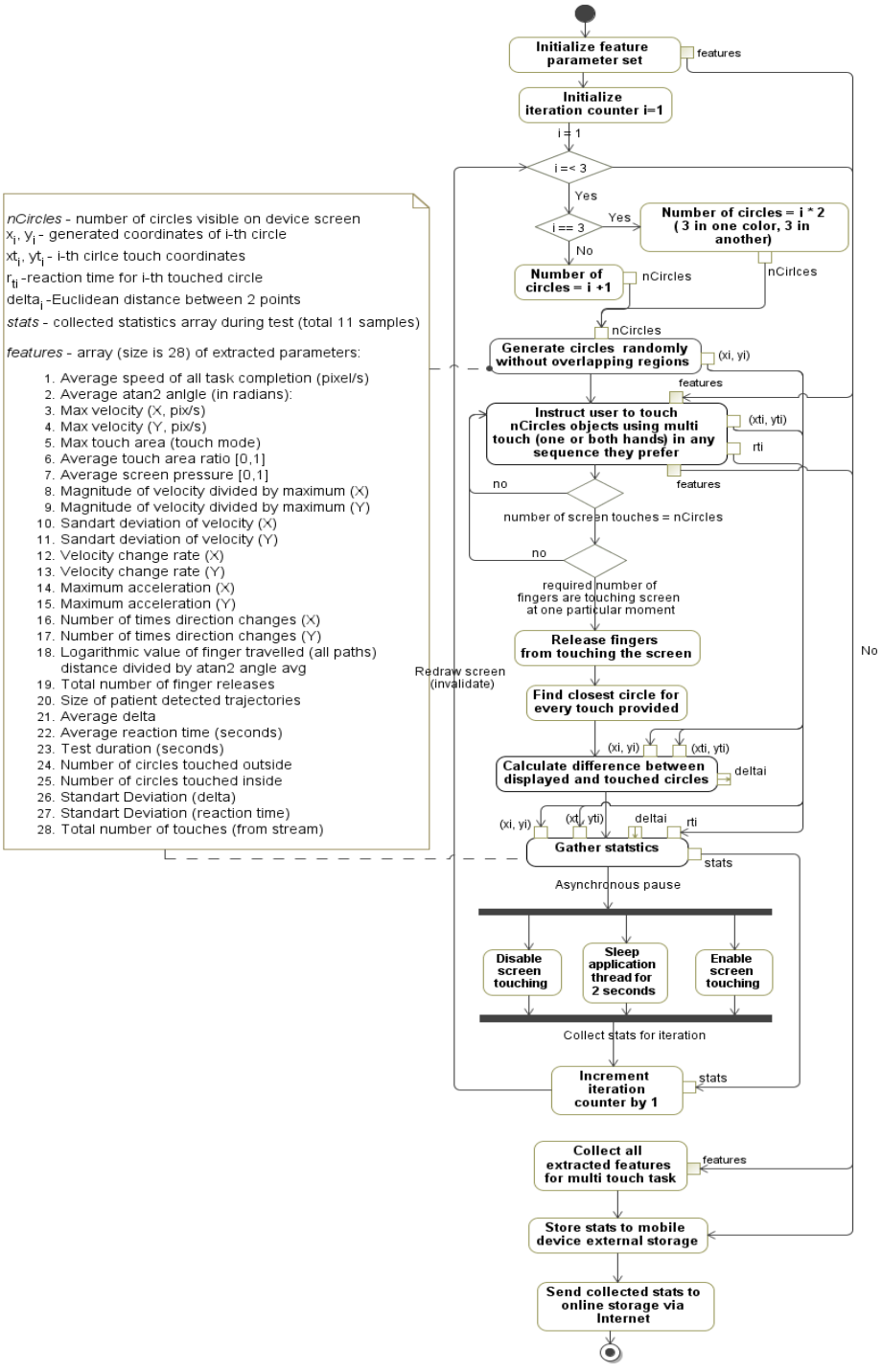


Figure 3.4. Algorithmic implementation of task T3: Multi-Touch

Methods for T1, T2 and T3 feature extraction:

$$delta\_avg = \sum_{i=1}^{n} delta_i \Big/ n, \quad delta_i = |p_i - st_i| \qquad \begin{matrix} n = 10 \text{ (T1)} \\ n = 5 \text{ (T2)} \\ n = 11 \text{ (T3)} \end{matrix} \qquad (1)$$

$$reaction\_time\_avg = \sum_{i=1}^{n} rt_i \Big/ n \qquad \begin{matrix} n = 10 \text{ (T1)} \\ n = 5 \text{ (T2)} \\ n = 11 \text{ (T3)} \end{matrix} \qquad (2)$$

$$test\_duration = \sum_{i=1}^{n} t_i \qquad (3)$$

n – total number of visual objects (circles), p_i – i-th object centre coordinates(x, y),
t_i – i-th object touch time (seconds), st_i – i-th screen touch coordinates,
rt_i – i-th object touch reaction time (seconds),
$test\_duration$ – total test execution time (seconds).

The same type of features (overall 9) are extracted to all three tests, only the number of total visual objects to be touched changes. The features: *t1_delta_avg*, *t1_reaction_time_avg*, *t1_test_duration*, *t2_delta_avg*, *t2_reaction_time_avg*, *t2_test_duration*, *t3_delta_avg*, *t3_reaction_time_avg*, *t3_test_duration*, are named respectively. Figure 3.2 illustrates the algorithmic implementation of T1, Figure 3.3 of T2 and Figure 3.4 of T3. For features element (vector) mentioned in Figure 3.2, Figure 3.3 and Figure 3.4 refer to 3.2.15 section of this work, i.e., a calculation of standard deviation (SD) for $delta_i$ and rt_i, finger motion tracking mechanisms. A circle is touched inside if $delta_i < radius$ and outside otherwise. The average screen touch area and pressure values in the range [0; 1] are accessed by using the built-in Android SDK functionality.

3.2.2. T4: Archimedean Spiral

Mathematically, Archimedean spiral is expressed in polar coordinates (r, θ) as indicated in formula (4) (changing the parameter a will turn the spiral, while b controls the distance between successive turning):

$$r = a + b\theta \qquad (4)$$

The working principle of T4: 2 different modes are supported. In the first mode, Archimedean spiral is shown to the user in the device screen clockwise. The patient is instructed to follow spiral contour with finger. When the finger is released and idle time of 3 seconds is reached or finger is released 3 times, the patient is redirected to second (drawing) mode. In the second mode, the patient is given 10 seconds to look at the Archimedean spiral drawn counter clockwise. After that, the mobile device screen is redrawn, i.e., left empty that the patient could try to replicate spiral contour with finger. T4 is finished when in the second mode, 3 second idle

time with one finger release is reached or finger is released 3 times. Figure 3.5 illustrates the working principle of T4 as UML activity diagram.

As for methods used for feature extraction, the author uses Frechet distance algorithm [85] to evaluate curve comparison between predefined spiral contour and patient's followed (drawn) contour as indicated in the formula (5). An analogy of the Frechet algorithm could be made with a man traversing a finite curved path while walking his dog on a leash, with the dog traversing a separate one. The algorithm corresponds to choosing the walk along the given paths where the maximum leash length is minimized. Ideally, Frechet distance outputs 0 for perfect match.

$$F(A,B) = \inf_{\alpha,\beta} \max_{t\in[0,1]} \left\{ d\left(A(\alpha(t)) \middle| B(\beta(t))\right) \right\} \tag{5}$$

$SF -$ metric space, $inf -$ infimum ($a \leq x,\ \forall\, x \in SF,\ \ SF \subseteq PMS$),
$A -$ spiral contour with representation A: $[0,\ 1] \rightarrow SF$,
$B -$ patient$'$ s drawn (followed) contour with representation B: $[0,\ 1] \rightarrow SF$,
$d - SF$ distance function (defines a distance between each pair of elements of a set),
$A(\alpha(t))$ –Spiral point in *t* time moment, $B(\beta(t)) -$ patient's drawn point in *t* time moment.

The proposed method considers the followed (drawn) spiral point match percentage (*mp*) by determining whether a patient's touched point is in radius (Euclidean distance between two points are calculated) of the closest found point from the predefined uniquely matched point set. Formula (6):

$$mp = \left(\sum_{i=1}^{msp} psp_i \right) \Big/ nsp \cdot 100$$

$$psp = \begin{cases} 1, & \sqrt{|x_{min} - x_{st}|^2 + |y_{min} - y_{st}|^2} \leq rsp \\ 0, & otherwise \end{cases} \tag{6}$$

$nsp -$ number of spiral points, msp –number of patient clicks in device screen,
$x_{st}, y_{st} -$ screen touch coordinates,
$x_{min}, y_{min} -$ closest spiral point to x_{st}, y_{st}, $rsp -$ spiral point radius.

T4 extracted features (both modes, with alias): *match_percentage1*, *fdistance1*, *match_percentage2*, *fdistance2*. Moreover, the set of finger motion tracking features.

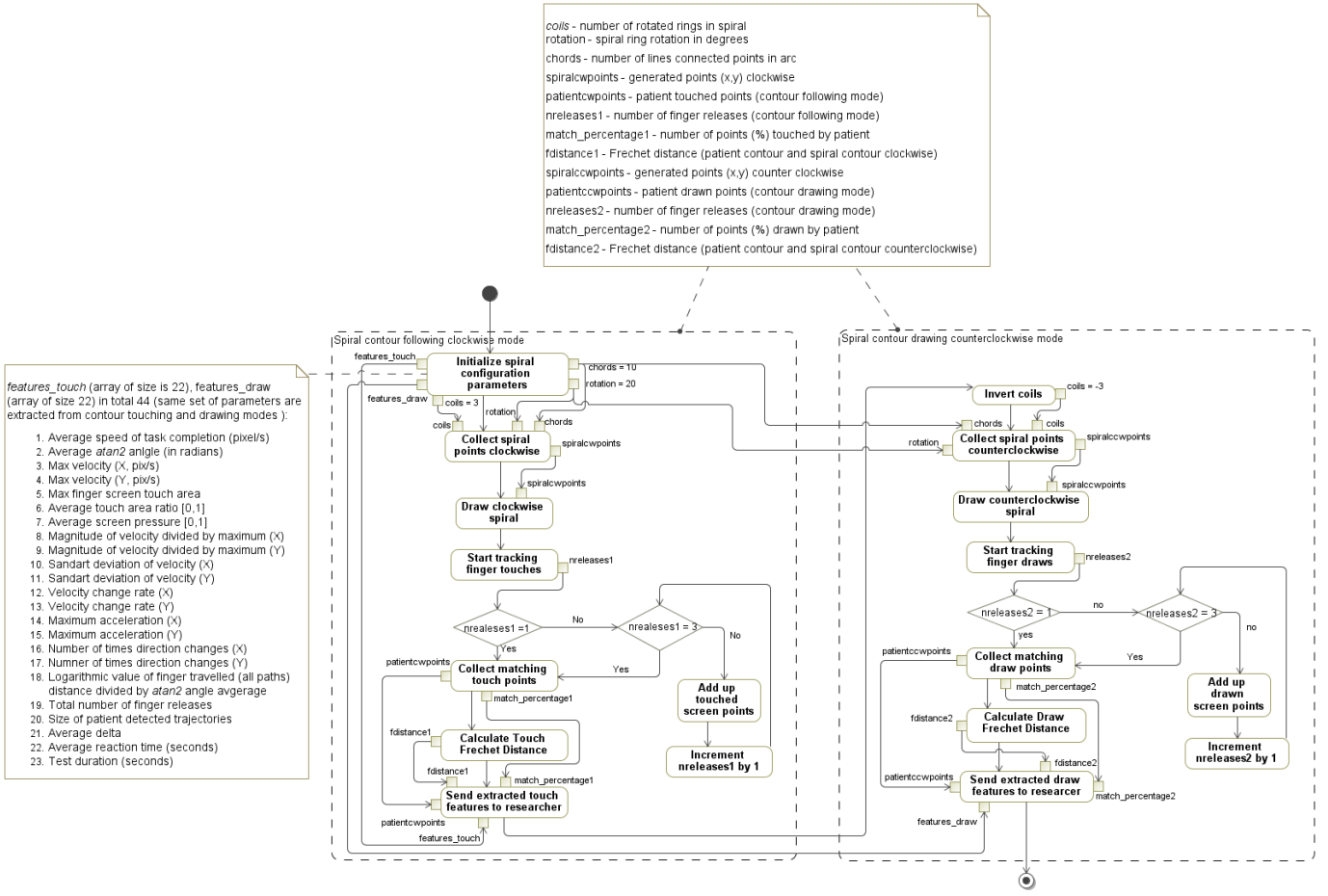


Figure 3.5. Algorithmic implementation of task T4: Archimedean Spiral

3.2.3. T5: General Insights

This is the first task from the adapted SAGE methodology. It is a non-scored item, i.e., basic questionnaire (9 questions in total) asking a patient about demographics, family history, problems of memory and thinking, motor symptoms, stroke symptoms, depression, personality changes, functional abilities.

As for computerization, the approach results in providing single question randomly (one at a time) and providing navigation (back and next buttons) between questions. In order to start, choose a question: a single option selection (Radio Group) is given and the text input (Edit Text) where needed. The test duration is included (feature *t5_duration_s*). The principle is illustrated in Figure 3.6 as UML.

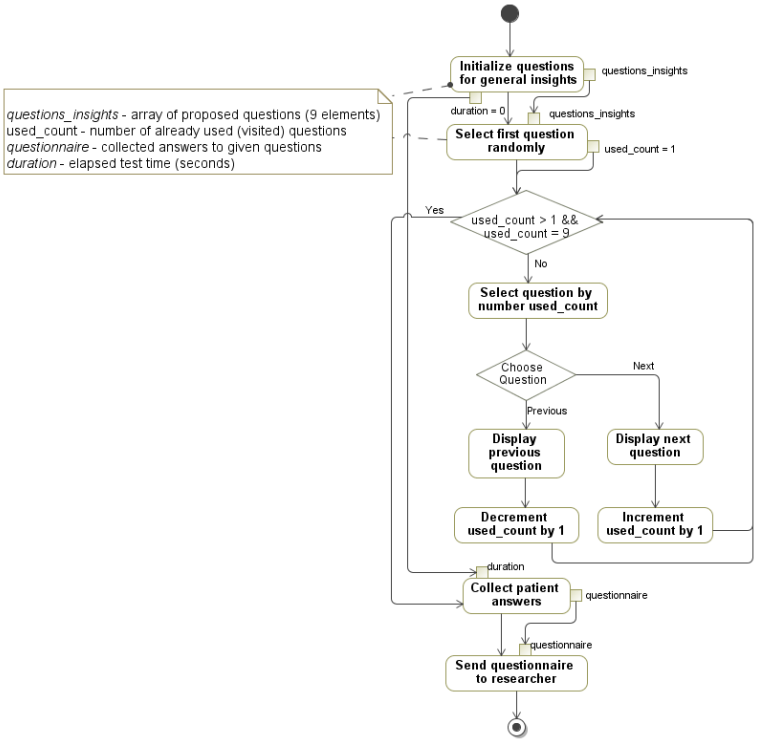


Figure 3.6. Algorithmic implementation of task T5: General Insights

3.2.4. T6: Orientation (current date)

This is the first scored item from SAGE. In this task, the patient is asked to enter the current date from his head (without cheating): year (Y), month (M) and day (D). The working principle is illustrated in Figure 3.7: a patient enters input data into three text fields. If data is valid, points are assigned for Y, M and D separately in order to accumulate SAGE score for T6.

T6 scoring instructions are adapted from SAGE and presented in (7) as automatic feature extraction mechanism (test duration is included):

$$tsage1 = Y + M + D \quad (7)$$

$$Y = \begin{cases} 1, & CY = Y \\ 0, & otherwise \end{cases}, M = \begin{cases} 1, & CM = M \\ 0, & otherwise \end{cases}, D = \begin{cases} 2, & CD = D \\ 1, & -3 \leq Diff \leq 3 \\ 0, & otherwise \end{cases}$$

Y – year input, *M* – month input, *D* – day input, *CY*, *CM*, *CD* – current year, month, day, *Diff* – difference between 2 dates (in days).

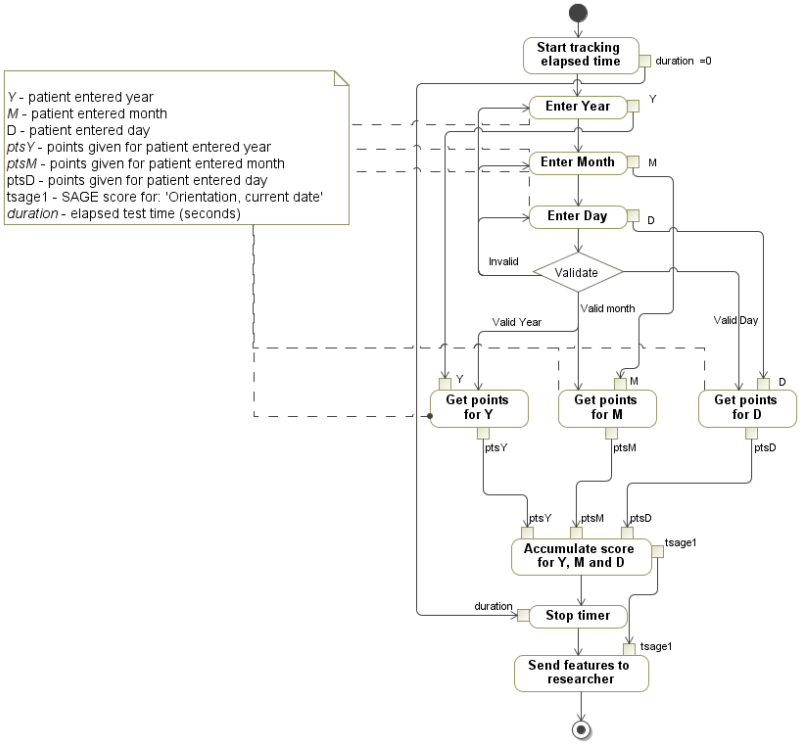


Figure 3.7. Algorithmic implementation of task T6: Orientation (current date)

3.2.5. T7: Picture Naming

The second scored item of SAGE methodology. In this task, a patient is instructed to name two pictures for visual inspection.

All pictures were collected from four SAGE forms (8 in total) to use the same images as proposed by methodology creators. Moreover, the procedure was improved by adding enhanced full colour *.jpg* images and extending image set with two extra pictures for a bigger surprise factor. Once the image set is associated with randomly selected SAGE group G [1; 4], a single picture is randomly (first or second) selected for the patient to name. Next, the patient is provided with an input field for entering text that is associated with the shown image. The navigation is available as well, i.e., a patient can go to the previous or next picture in case when an already given answer needs to be corrected.

The computerization approach is provided in Figure 3.8 as UML workflow:

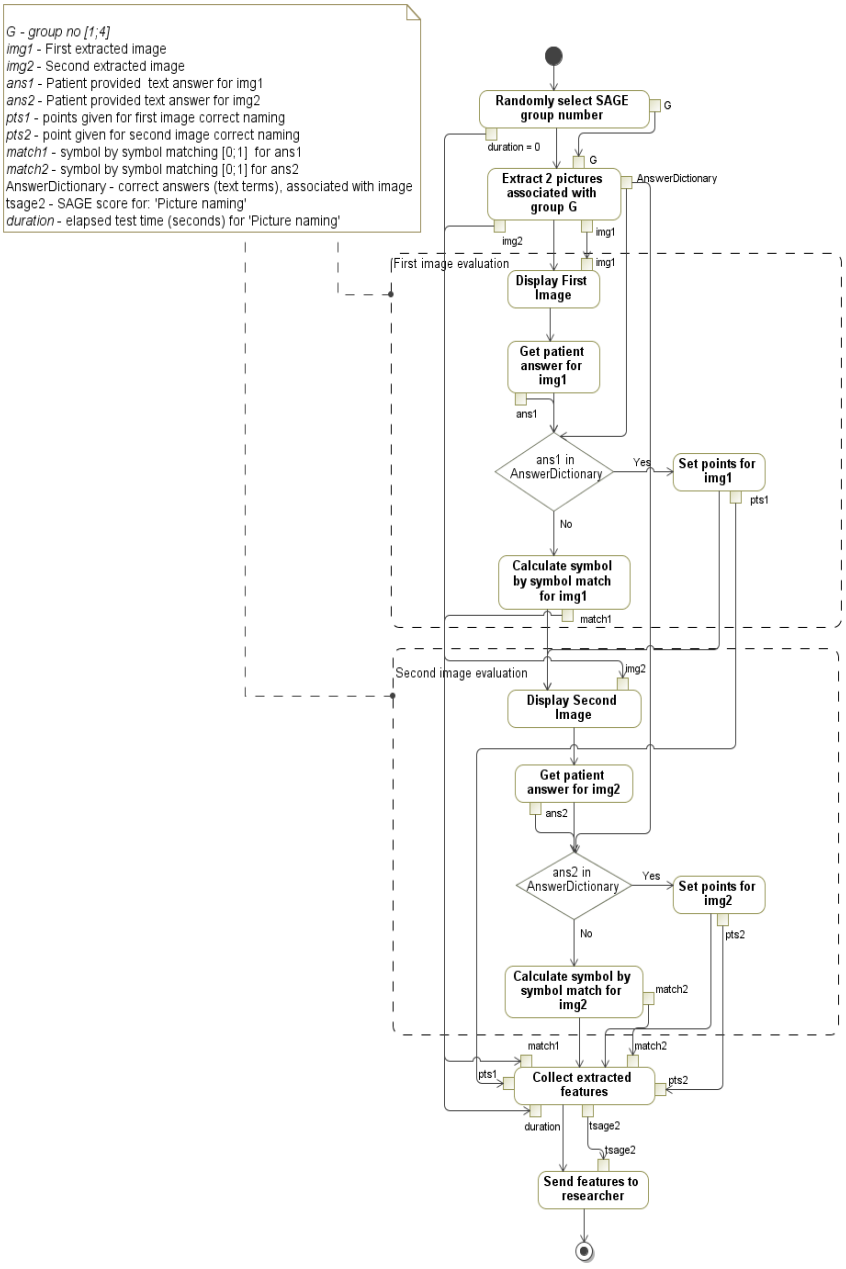


Figure 3.8. Algorithmic implementation of task T7: Picture Naming

As for patient's answers evaluation of T7 task, two strategies are defined, because SAGE states are not for mind spelling: 1) exact string matching, 2) imprecise string matching. For the first strategy, the author defines correct answer

dictionaries for every single picture and compares with the patient's given answer (ignore case string matching). The patient is evaluated by using formula (8), according to SAGE scoring instructions:

$$tsage2 = \begin{cases} 2,\ Img1\ \in TDictionary\ \wedge Img2\ \in TDictionary \\ 1,\ Img1\ \in TDictionary\ \vee Img2 \in TDictionary \\ 0,\ otherwise \end{cases} \quad (8)$$

Img1 – first image, *Img2* – second image, *TDictionary* – correct answer dictionary (for image),
$\wedge$ - logical 'and', $\vee$- logical 'or'.

In case of a situation when a patient actually knows the picture's name but makes a spelling error (e.g., tremor occurs etc.), the second strategy is adapted. For the imprecise string matching, the author applies Jaro-Winkler (JA) algorithm [85] as defined in formulas (9), (10). The comparison between patient's entered text and closest found match from *TDictionary* is executed. Jaro-Winkler method outputs 1 for the perfect string match, using symbol by symbol comparison technique. The proposed algorithm is useful when the SAGE score for T7 is 0, but a patient still has made reasonable efforts in picture naming (later on, some extra classifiers can be applied based on the observations of JA values).

$$JA = \begin{cases} 0, & mj = 0 \\ \frac{1}{3}\left(\frac{mj}{|s_1|} + \frac{mj}{|s_2|} + \frac{mj - t}{mj} \right), & otherwise \end{cases} \quad (9)$$

$$\left\lfloor \frac{\max(|s_1|, |s_2|)}{2} \right\rfloor - 1 \quad (10)$$

mj – number of matching symbols by equation (10), $|s_i|$ – length of s_i line,
t – transpositions: number of matching symbols in different order / 2.

T7 extracted features (with alias): *match1* (corresponds to JA for the first image), *match2* (corresponds to JA for the second image), *tsage2*. Moreover, an average of JA matches is calculated as a feature: *t7_ jaro_closest_match_avg*, *t7_duration.*

3.2.6. T8: Similarities, Calculation

T8 includes three questions (3-rd, 4-th and 5-th) of SAGE methodology to one task in the proposed mobile application.

In the first question (Q1), a patient is given a text query, which requires finding similarity between two listed items (e.g., a watch and a ruler or a corkscrew and a hammer). Two forms for answering are considered: 1) abstract, 2) concrete. The maximum score for Q1 is given when a patient answers in an abstract manner, i.e., finds a connecting above level category (in computer science terminology 'superclass') to which both items can be classified. In Q2 and Q3, patient's mathematical knowledge is tested.

Q2 requires performing a mathematical subtraction operation of two floating numbers. Q2 question text context is about going to the grocery store and buying items for a specified money value M_n to calculate the change received from the established bill B_e ($B_e > Mm$ chosen randomly in interval [0;100]).

Q3 requires performing a mathematical division operation of two floating numbers. Q3 context is about having a particular sum of money S and denomination coins DCN (e.g., dime, nickel, quarter etc.) to calculate how many coins will be needed to collect S. For Q2 and Q3, a patient cannot use a calculator (only his head or a sheet of paper). The computerization approach is presented in Figure 3.9 as UML workflow.

As for methods used for the patient's evaluation of T8, in Q1 (tsage3), the author applies exact and imprecise string matching (including Jaro-Winkler method) as described in T7: Picture Naming. The difference is that predefined dictionaries with correct answers for a single term have associated point values (2 for abstract answer, 1 for concrete). For Q2 evaluation: 1 point is given (tsage4) when patient answerQ2 = $B_e - M_n$. For Q3 evaluation: 1 point is given (tsage5) when patient answerQ3 = S / DCN. All three extracted features are accumulated into one as defined in formula (11):

$$tsage3\_4\_5 = \begin{cases} 4, & Q1 \wedge Q2 \wedge Q3 \\ 3, & Q1 \wedge (Q2 \vee Q3) \\ 2, & Q1 \vee (\overline{Q2} \wedge \overline{Q3}) \\ 1, & (Q2 \vee Q3) \\ 0, & otherwise \end{cases} \quad (11)$$

$\wedge$ - logical 'and', $\vee$- logical 'or', *Q1* – first question (abstraction 2 points), *Q2* – second question (mathematical subtraction 1 point), *Q3* – third question (mathematical division 1 point).

T8 extracted features (with alias): *t8_ jaro_closest_match* (for Q1), *tsage3_4_5*, *t8_duration*.

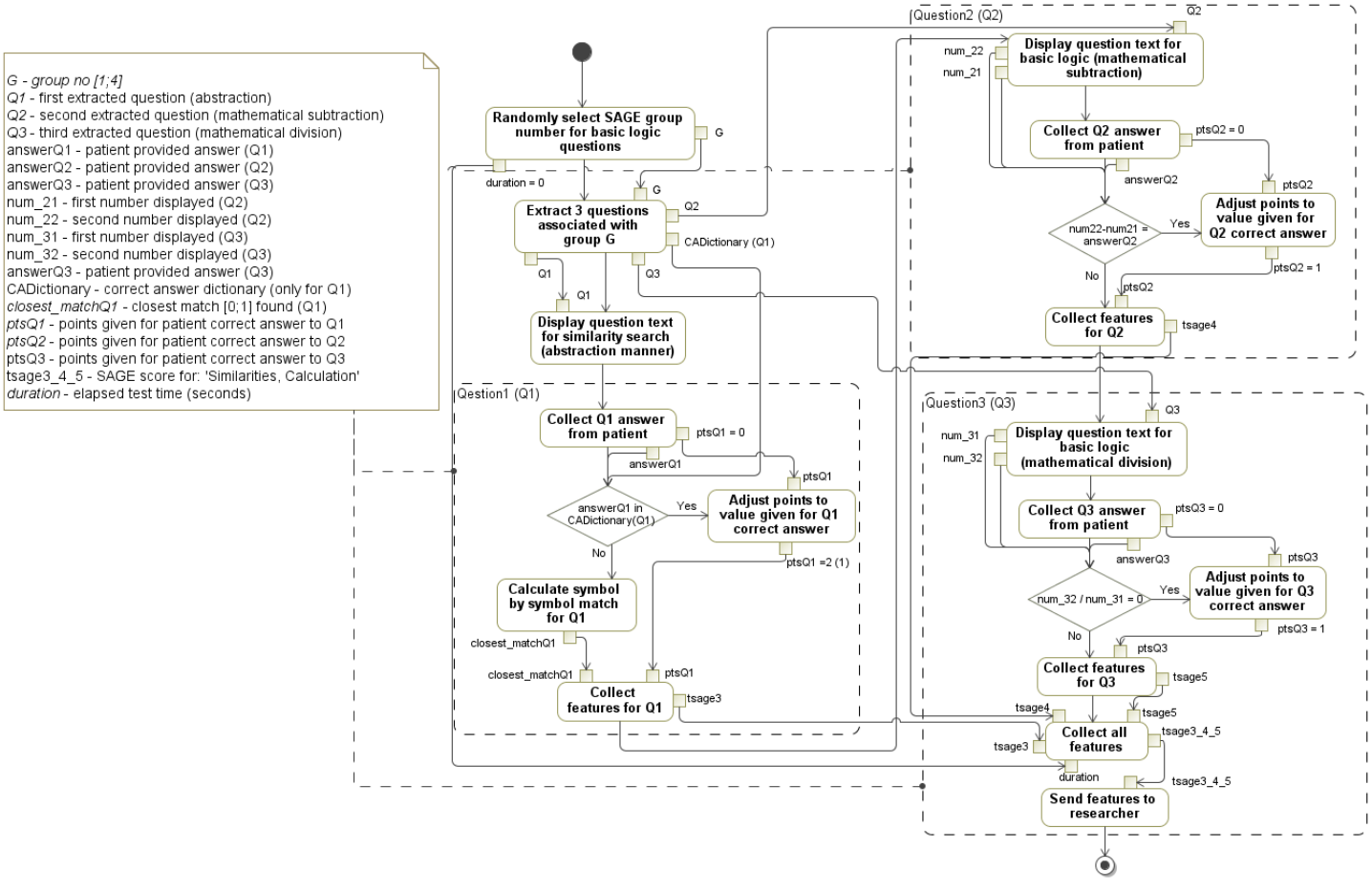


Figure 3.9. Algorithmic implementation of task T8: Similarities, Calculation

3.2.7. T9: Construction (3-D figure)

T9 covers 7-th question of SAGE methodology. T9 requires a patient to reconstruct a given 3-D figure (cube, ribbed rectangle) (Figure 3.10).

As defined in SAGE, there can be 4 different 3-D figures: cube, rectangle and variations (e.g., cube missing a surface). Thus, first, SAGE group G [1; 4] is randomly selected. Two modes are distinguished: preparation and actual task execution. In preparation mode, the proposed 3-D figure is drawn in the centre of the screen with instructions at the bottom for the patient to remember connections. In the background, the displayed figure is treated as 8-node bidirectional graph (connections → edges). A node is characterized by screen point coordinates (x, y) and edges to the other nodes (referred by node number [1; 8]). In addition, such graph is transformed into adjacency matrix *AdjM1*, where element is equal to 1 if an edge between two nodes exist and 0 otherwise.

After reading the instructions, the patient clicks on the mobile device screen and is redirected to task execution mode. There, eight graph nodes are displayed, but no edges are visible, because the task for a patient is to form proper connections. An edge is formed by using free drawing manner (adapted from SAGE), i.e., a patient must draw a path with his finger on screen between two nodes. The author assumes that constructing the 3-D figure in such way can trigger many errors (especially for patients with hand tremor symptoms); thus, these error statistical metrics are collected:

1. *err_logical* – total number (counter) of wrongly connected nodes;
2. *err_tlimit* – total number of time limit exceed errors (configuration setup is 3 seconds);
3. *err_toofew* – total number of errors when too few nodes are selected when forming a connection;
4. *err_toomany* – total number of errors when too many nodes are selected when forming a connection;
5. *err_exist* – total number of errors when an already existing connection is selected.

If any error occurs, a patient is instructed with a message (text label) at the bottom of the screen to try again. An edge (successful connection) is created when only two nodes are touched in-between time intervals (< 3 seconds) of finger releases. When the task is finished (special button is clicked in the action bar), a second graph is created with another adjacency matrix *AdjM2* (the one that patient forms).

For SAGE score assignment, neighbour-matching algorithm is adapted (returns one for perfect match, i.e., patient constructed 3-D figure ideally) for a graph similarity measure (*graph_NM*) [86] as defined in (12), (13), (14), (15), (16), (17) and (18) formulas:

$$x_{ij}^{k+1} \leftarrow \frac{s_{in}^{k+1}(i,j) + s_{out}^{k+1}(i,j)}{2} \tag{12}$$

$$s_{in}^{k+1}(i,j) \leftarrow \frac{1}{m_{in}} \sum_{l=1}^{n_{in}} x_{f_{ij}^{in}(l) g_{ij}^{in}(l)}^{k} \tag{13}$$

$$s_{out}^{k+1}(i,j) \leftarrow \frac{1}{m_{out}} \sum_{l=1}^{n_{out}} x_{f_{ij}^{out}(l) g_{ij}^{out}(l)}^{k} \tag{14}$$

$$m_{in} = \max(id(i), id(j)) \tag{15}$$

$$n_{in} = \min(id(i), id(j)) \tag{16}$$

$$m_{out} = \max(od(i), od(j)) \tag{17}$$

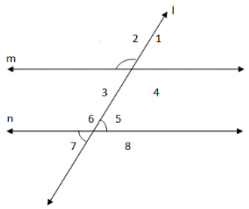
$$n_{out} = \min(od(i), od(j)) \tag{18}$$

$S_{in}(i,j)$ – the in degree similarity of node *i* in A graph and *j* in B graph, $S_{out}(i,j)$– the out degree similarity of node *i* in A and *j* in B, *id()* – in-degree of the node, *od()* – *out*-degree of the node, *f(i,j) and g(i,j)* – enumeration functions (functions that give the maximum similarity value for each node in the given node list).

Another aspect of evaluating patients in T9 task is the calculation of parallel line angles [87] between particular edge pairs in constructed 3-D figure. This is important, because the path (trajectory) drawing is performed. The parallel line angle calculation is done by taking arctangent of two slopes (*m1*, *m2*) of the analysed line pairs (19) and adding the extra traversal line (l) (20) to setup 8 angles. According to SAGE methodology, the tolerance of parallel lines is 10˚. The author adapted extra tolerance match percentage factor (configuration setup is 50%) for the estimation of how many parallel line pairs (9 total) were connected with angle < 10˚.

$$angle = tan^{-1}\left(\pm \frac{m_1 - m_2}{1 + m_1 m_2}\right) \tag{19}$$

$$m \,||\, n,\ l\ transversal$$
$$\angle 3 \cong \angle 5,\ \angle 4 \cong \angle 6,$$
$$\angle 2 \cong \angle 8,\ \angle 1 \cong \angle 7 \tag{20}$$



angle - between two edges (degrees), *m1*, *m2* – slopes of 2 analysed graph edges (e.g., *m* and *n*).

Finally, SAGE score (tsage7) is assigned by using formula (21):

$$tsage7 = \begin{cases} 2, & \text{proper } 3D \wedge L \leq 10^{\circ} \\ 1, & 3D \wedge L > 10^{\circ} \\ 0, & \text{otherwise} \end{cases} \tag{21}$$

∧ - logical 'and', 3D – proper figure constructed (true if g*raph_NM* = 1), L – edge parallelism in degrees.

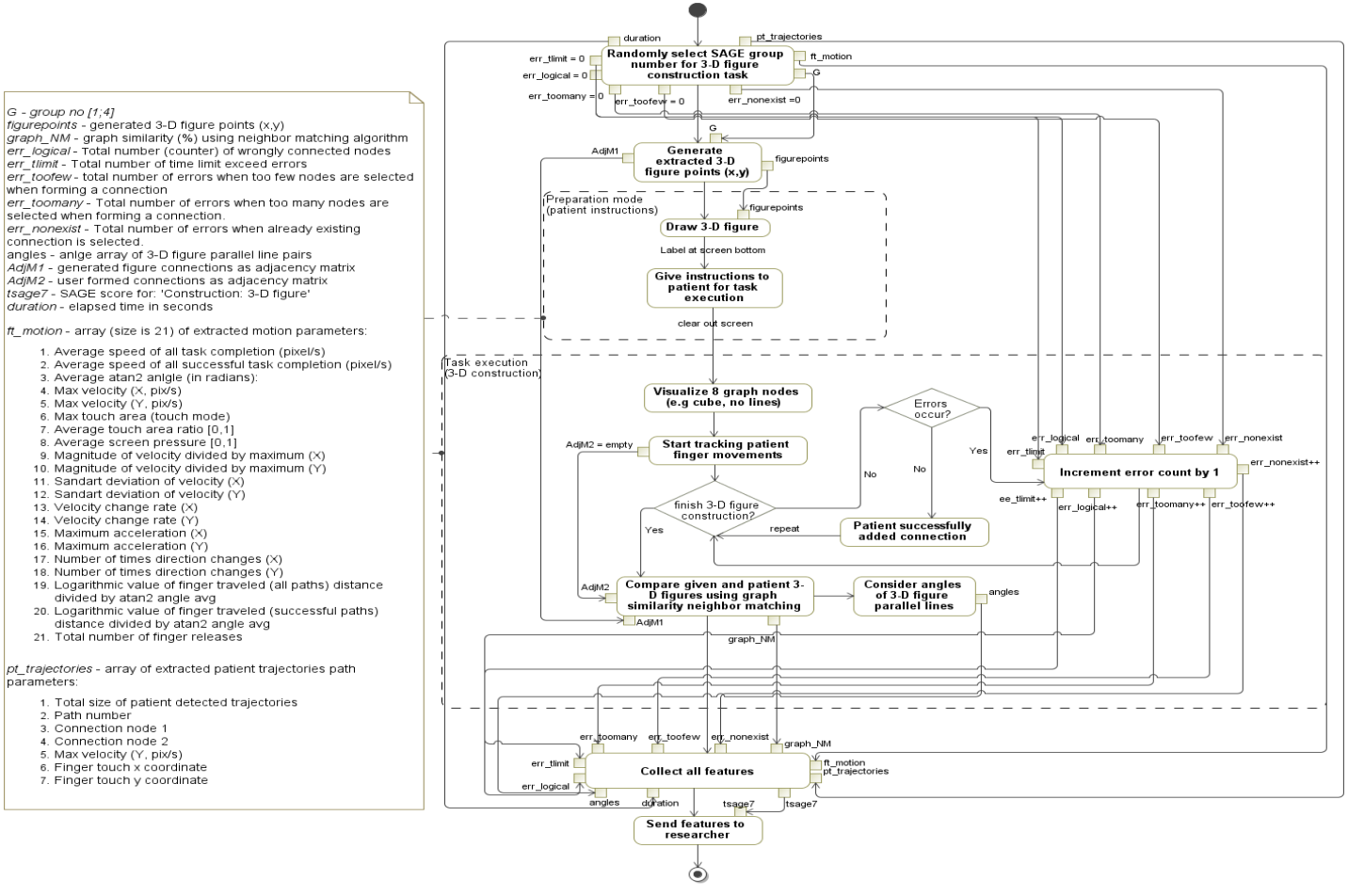


Figure 3.10. Algorithmic implementation of task T9: Construction (3-D figure)

3.2.8. T10: Construction (Clock)

T10 covers the 8-th question in SAGE survey. In T10, a patient is instructed to clock drawing test (CDT) by using his finger or a smart pen (if preferred). Figure 3.11 shows the computerization of CDT task in the proposed mobile app. Two modes are supported, i.e., preparation and CDT task execution. Firstly, in preparation mode, an example analogue clock, showing defined hours (H, random value in [1; 12]) and minutes (M, random value in [0; 60], 5 minute step), is displayed. The draw clock components are clock face and numbers, all 12 numbers, hand positions and hand labelling (S – hours, L - minutes). A patient is given instructions at the bottom of the screen (text label) to remember clock. When the mobile device screen is clicked once, a patient is redirected to CDT execution mode. There, an active zone (contour) is provided for drawing a clock. If a patient violates active contour, an error is triggered (*err_contour*). Similarly, if finger long press is triggered outside the active zone, the clock needs to be redrawn (*err_redraw*) and the screen cleared out. The task is finished (redirected for supervision) if a special indicator button in action bar is clicked. At that moment, the patient's evaluation of CDT task is semi-automatic, i.e., a supervisor is required to be near the patient and assess the drawn clock by using (22) formula:

$$tsage8 = \begin{cases} 2\,,\ C = 4 \\ 1\,,\ C = 3 \\ 0\,,\ C < 3 \end{cases} \tag{22}$$

C – number of correctly drawn components (clock face and numbers, all 12 numbers in correct order, hand positions, hand labelling).

T10 extracted features (with alias): *t10_err_redraw*, *t10_err_contour*, *tsage8*, *t10_duration*, finger motion tracking features.

3.2.9. T11: Verbal Fluency

T11 is the computerization of 9-th question of SAGE survey. In T11, a patient is instructed to write down 12 different items (elements) in the given category. There are 4 different categories (associated with SAGE group): animals, fruits or vegetables, things that are found in kitchen (not including food) and countries of the world. In the mobile app, twelve text fields are provided for the patient input (not more than 3 words should be entered for a single name). The text field can be left blank, in case a patient does not remember any items of the category. In order to finish T11 task, 'OK' button must be pressed. Figure 3.12 illustrates the process.

In order to evaluate the patient, a similar method, as described in T7 and T8 (for Q1), is applied. Firstly, the predefined dictionaries (*TDictionary*) for storing correct answers are created by using available open educational resources (OER) on the internet. *TDictionary* is expandable, i.e., if a new item that fits the category is found, it is added to the dictionary. Please refer to the reference list below to look up the online OER used to collect the dictionaries [88], [89], [90], and [91].

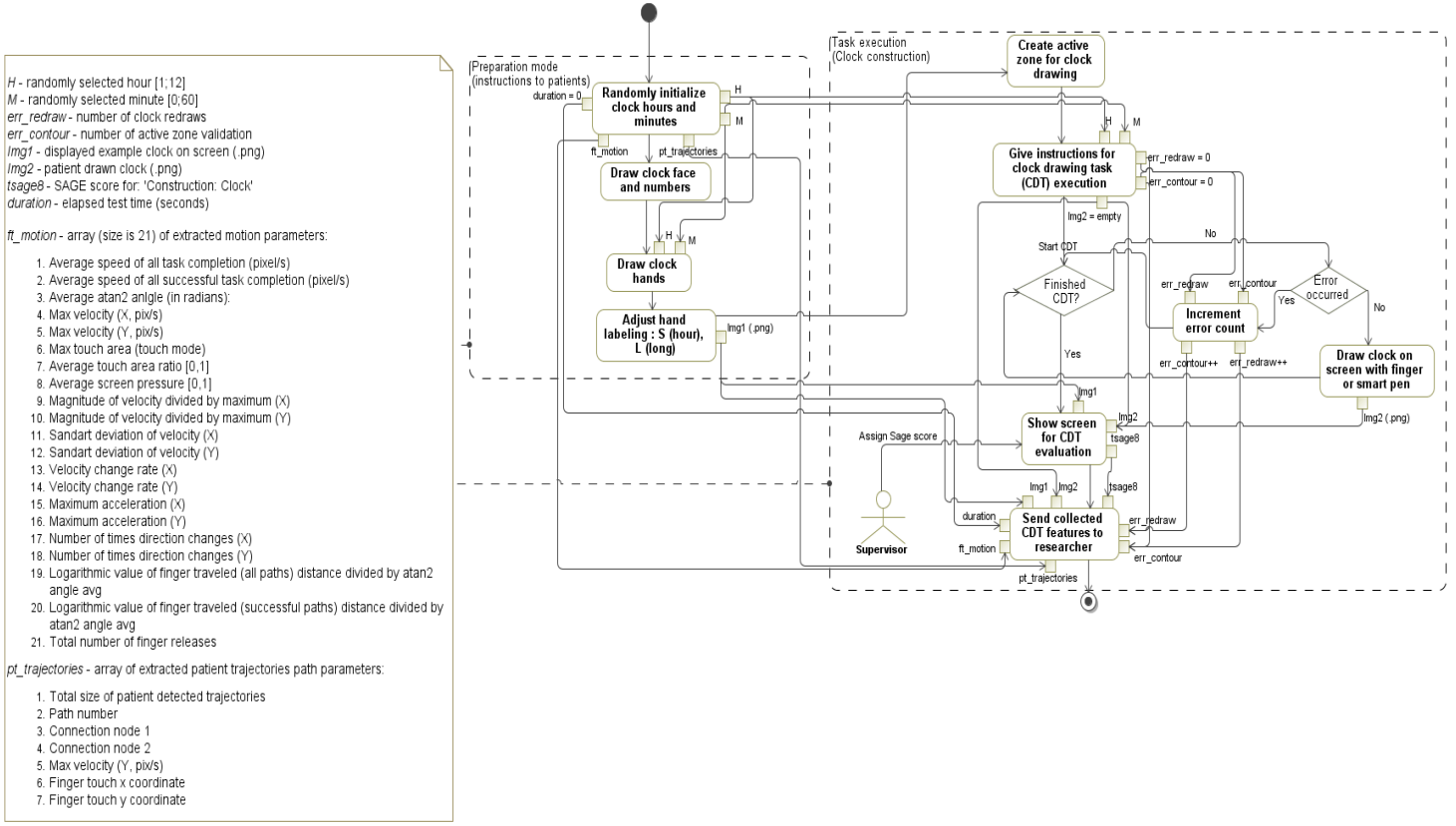


Figure 3.11. Algorithmic implementation of task T10: Construction (Clock)

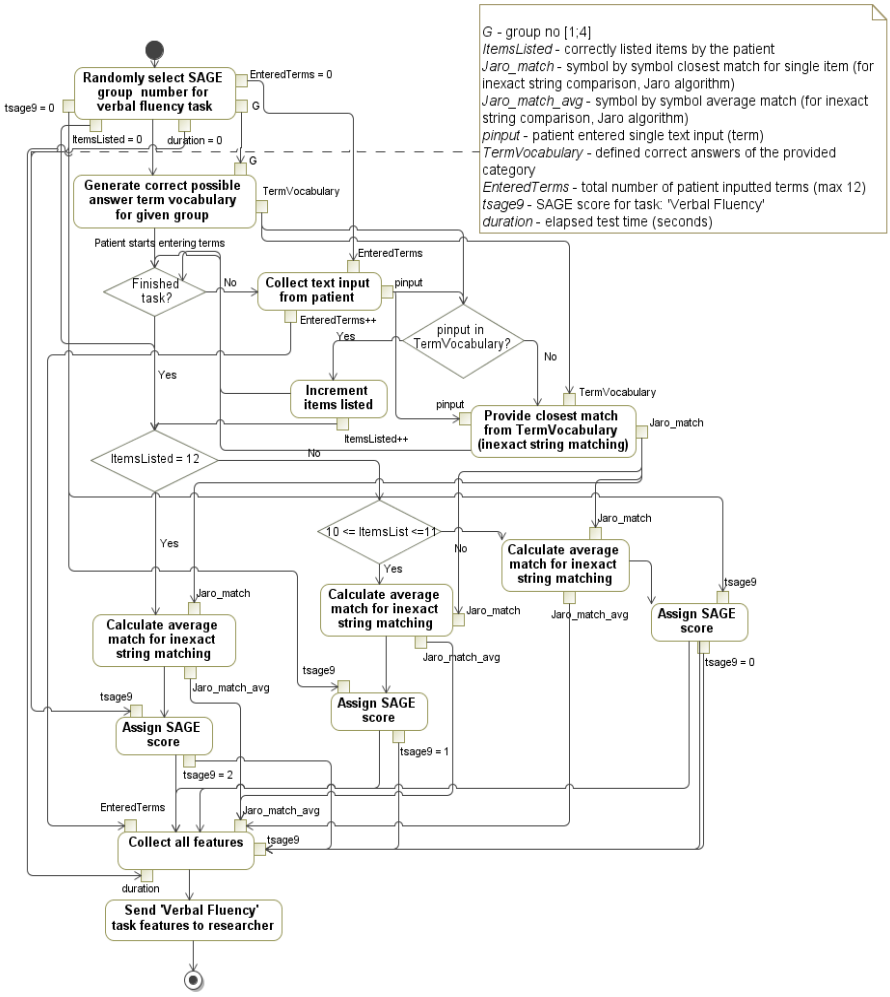


Figure 3.12. Algorithmic implementation of task T11: Verbal Fluency

The maximum SAGE score for T11 is 2 points (given when the patient enters 12 different items), 1 point (10 or 11 correct items) and 0 points (otherwise) as defined in formulas (23), (24). In case of imprecise string matching (when the input term is not found in a dictionary), the Jaro-Winkler method is used to compare symbol-by-symbol equivalent of closest found match from the dictionary. Finally, the average of all Jaro-Winkler calculations to 12 items is calculated (for a perfect match, the total average is 1).

$$tsage9 = \begin{cases} 2\,, Elem = 12 \\ 1\,,\ 10 \leq Elem \leq 11 \\ 0\,, Elem\ < 10 \end{cases} \tag{23}$$

$$Elem = \begin{cases} 1, PE \in TDictionary \\ 0, PE \notin TDictionary \end{cases} \quad (24)$$

Elem – number of correctly listed items, *PE* – patient element (text term) input, *TDictionary* – defined correct answers (using OER) of the provided category.

T11 extracted features (with alias): *t11_items_listed*, *tsage9*, *t11_jaro_closest_match_avg*, *t11_duration*.

3.2.10. T12: Executive (Modified Trials)

T12 corresponds to the 10-th question of SAGE survey. In T12, a patient is asked to follow a pattern in schema, i.e., to draw a line from one circle to another, starting at 1 and alternating numbers and letters (1 to A to 2 to B to 3 to C). Four different variants [1; 4] (tasks to solve) have been transferred from SAGE forms (one is selected randomly). As for computerization (see Figure 3.13), firstly, a patient is provided with an example schema and instructions of task execution at the bottom of the screen. In the mobile app, a schema is interpreted as a graph (nodes and edges). A node can be labelled with a letter from the English alphabet (e.g., 'A') or a number (e.g., '1'). The node is characterized with screen position coordinates (x, y), label, node number and array of connections to the other nodes. When an actual task execution begins, the screen is redrawn: new nodes (extended list to form pattern 1 to A to 2 to B to 3 to C to 4 to D to 5 to E to 6 to F) with different point, positions are generated. No lines connecting particular nodes are drawn, because this is a task for the patient.

In order to form a connection between two nodes, a patient uses finger free-drawing form (similarly as in T9 task) to create path trajectories. When the finger is released, the path is processed and successfully formed only if exactly two nodes have been touched (measured by Euclidean distance between two points).

Many erroneous situations may occur when trying to create a valid connection (especially for patients with neurological disorders). For this reason, error-tracking mechanism (*err_logical*, *err_samenode*, *err_outside*, *err_binary*) is incorporated. The assumption is that the more errors are made, the worse is the task execution. T12 task is finished when a particular button is pressed in the action bar.

1. *err_logical* – total number (counter) of wrongly connected nodes
2. *err_samenode* – total number of errors when clicking on the same node
3. *err_outside* – total number of clicks outside the node
4. *err_binary* – total number of binary connection violation for a particular node.

As for T12 task, for the patient's evaluation, the node connections of the predefined and patient's created schemas are compared. If a particular edge for node does not exist (bidirectional node analysis), logical errors are incremented. Furthermore, the final SAGE score assignment is defined in (25) formula:

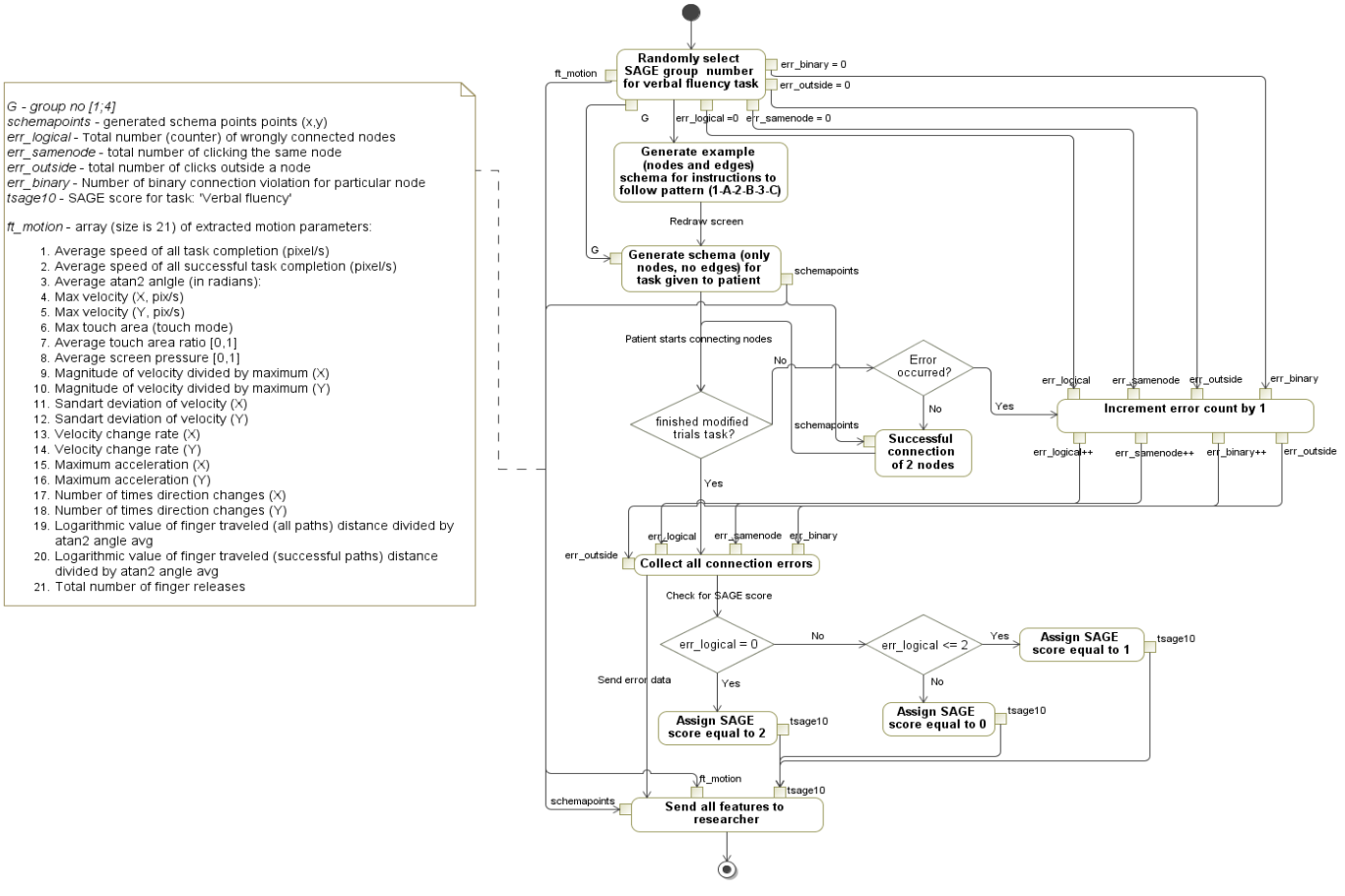


Figure 3.13. Algorithmic implementation of task T12: Modified Trials

$$tsage10 = \begin{cases} 2, Err = 0 \\ 1, Err \leq 2 \\ 0, Err > 2 \end{cases} \quad (25)$$

Err – number of logical errors (incorrect connection of graph nodes).

T12 extracted features (with alias): *t12_err_samenode*, *t12_err_nodeoutside*, t12_err_logical_connection, *t12_err_binary_connection*, *tsage10*. In addition, the graph similarity metric is calculated (Neighbour Matching algorithm) between predefined and patient's schemas. Moreover, Frechet Distance is applied to linear interpolation routine [106] to each properly formed path, e.g., perfectly straightened the line connecting node '1' to 'A' and patient's formed path between nodes '1' to 'A' etc.

3.2.11. T13: Executive (Problem Solving)

T13 corresponds to the 11-th question of SAGE survey. It is like a simulation of a simple game when having a particular number of matches (they form some geometrical shape, e.g., a triangle) and performing move actions, a new shape is formed. SAGE methodology has 4 different geometrical shape transformation (problem solving) tasks. All these variants are integrated into the proposed mobile application. The 1-st and second variants require forming 4 squares from 2 squares and 2 triangles (8 maximum operations are allowed). The 3-rd variant requires creating 4 squares from defined 5 squares by using 3 lines cross out operations. The 4-th variant requires to form 3 triangles out of the predefined 4 triangles by using 2 lines cross outs. In all variants, a problem-solving (PS) tasks should be executed in a way that no extra lines are left on the board (grid or mosaic), and each line is a part of a complete mosaic.

As for computerization (please see Figure 3.14 for details), two modes are presented to a patient in the mobile app: preparation and PS task execution. In preparation mode, an example mosaic is displayed for a patient, including instructions to complete the PS task (operations left, how to perform an operation etc.). In the mobile app, a mosaic is treated as a graph (nodes and edges), which is transformed to adjacency matrix (*AdjM1*). Furthermore, every mosaic is associated with SAGE variant [1; 4]; thus, the correct answers are as well established in the adjacency matrix form (*AdjM2*). Multiple correct answers for a particular mosaic configuration are possible (this depends on which variant is randomly selected at the beginning). It is ensured that all possible correct answers for given grid are covered. When a patient actually starts to move lines, two types of operations are allowed (insert and remove). In order to insert a line, two nodes (connection between them does not exist) needs to be clicked sequentially on the screen. When aiming to remove a line, two nodes (connection between them exists) are sequentially touched.

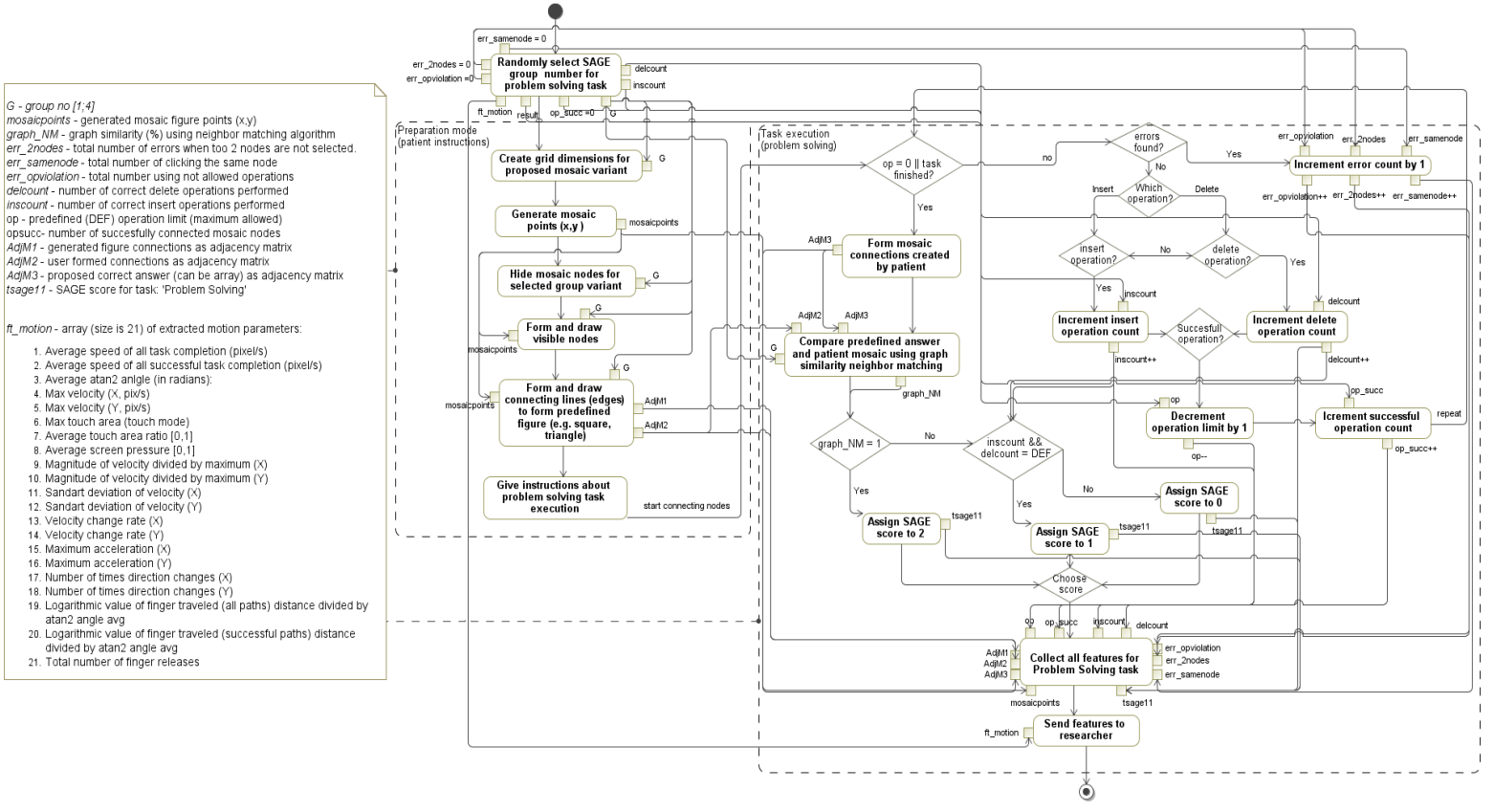


Figure 3.14. Algorithmic implementation of task T13: Problem Solving

An operation is successfully executed (left operations are decremented) when touching of two nodes occurs inside each node (measured by Euclidean distance between two points). Similarly, as in T9 and T12, many errors may occur (especially to patients with neurological disorders). Such errors are the following: touching the same node twice (*err_samenode*), touching a node outside the zone of radius (*err_2nodes*), violating allowed operations (*err_opviolation*), e.g., trying to execute the insert operation, when only delete is allowed. For this reason, the author adapted statistical error-tracking mechanism. The assumption is that the more errors are made, the worse is T13 task execution. T13 task finishes when operations left are equal to 0 or a special button is clicked on the action bar (patient *AdjM3* matrix is created).

The evaluation of patient's T13 execution is similar to the one described in T9, i.e., the Neighbour matching algorithm for the graph similarity measure is adapted. The comparison is made between the adjacency matrices (*AdjM2* and *AdjM3*) to get *graph_NM* value as defined in (26). In case *graph_NM* is not equal to 1, the total number of executed insert and delete operations are calculated (if they match the defined required operations count (DEF), 1 is assigned, 0 otherwise).

$$tsage11 = \begin{cases} 2\,,\ graph\_NM = 1 \\ 1\,, inscount \wedge delcount = DEF \\ 0\,,\ otherwise \end{cases} \tag{26}$$

graph_NM – Neighbour matching similarity measure of defined and patient's created mosaics (graphs), *DEF* – defined number of allowed operations for a particular mosaic, *delcount* – number of patient's performed delete operations, *inscount* – number of patient's performed insert operations.

T13 extracted features (with alias): *t13_operation_count*, *t13_graphs_similarity_nm*, *t13_succesfull_op*, *t13_err_2nodes*, *t13_err_op_violation*, *t13_err_samenode*, *tsage11*, *t11_duration*, finger motion tracking features.

3.2.12. T0: Memory

Last (6th and 12-th) task of the SAGE methodology: in T0, a patient is instructed to memorize a phrase that he was asked to remember before, i.e., in a particular point of time of the whole test process. SAGE proposes two phrases: 'Have you finished?', 'Are you done?' Maximum 2 points are given, when exact wording (no extra wording) is provided, 1 point if a particular word ("finished" or "done") is found in patient input text, 0 otherwise. For T0 computerization (refer to Figure 3.15), two text strings (*s1*, *s2*) are considered. If *s1* and *s2* match (ignore case) 2 points. 1 point: *s2* is split into tokens by predefined delimiters, and the extracted word search operation is executed (27) (test duration is included as well):

$$tsage\_6\_12 = \begin{cases} 2\,,\ s1 = s2 \\ 1\,,\ s2 \subset s1 \\ 0\,,\ otherwise \end{cases} \tag{27}$$

s1 – line that needs to be remembered, *s2* – text input by patient, ‚⊂' – *s2* element, i.e., the word is found in *s1*.

Finally, a total SAGE score is calculated by (28) formula:

$$SAGE\_TOTAL = \sum_{i=1}^{11} tsage_i \tag{28}$$

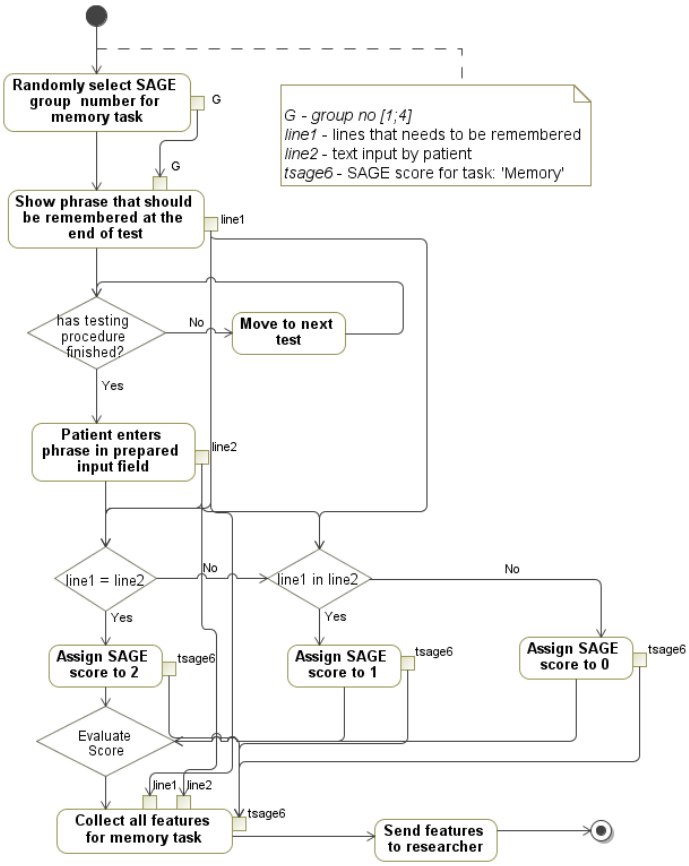


Figure 3.15. Algorithmic implementation of task T0: Memory

3.2.13. T14: Voice Recorder

In the proposed mobile app, T14 instructs a patient to read a short text of predefined poems into the mobile device microphone (Figure 3.16). The process is repeated two times, i.e., first, a poem is selected randomly; then, the remaining one is displayed. The recording begins, when a patient is ready and presses the button 'Start Recording'. Single poem recording finishes by pressing 'Stop Recording' button (a patient can make a pause if needed before the second recording). After T14 is completed, two audio files (*.mpeg4* format), together with the associated transcripts, are stored in the external storage of a mobile device.

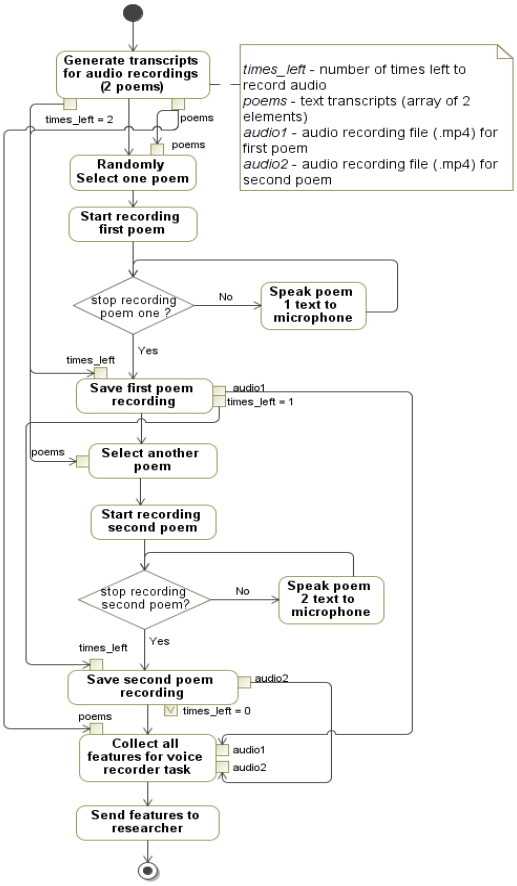


Figure 3.16. Algorithmic implementation of task T14: Voice Recorder

Stored *.mpeg4* files (*audio1* and *audio2*) are used as inputs for further audio signal processing. The author considers the following methods (listed in the paragraphs below) for speech feature extraction.

The pitch for the estimation of fundamental frequency (Hz) of an audio signal with rate *fs* can be 44.1 kHz. The pitch function estimates the fundamental frequency of the input signal at locations determined by the *WindowLength* (default size is *round* (*fs**0.052)) and *OverlapLength* (default size is *round* (*fs**0.042)) name-value pairs, *WindowLength* < *OverlapLength*. Pitch contours (*WindowLength* and *OverlapLength* ranges) are estimated by 5 methods: 'NCF' — Normalized Correlation Function [107] , 'PEF' — Pitch Estimation Filter [108], 'CEP' — Cepstrum Pitch Determination [109], 'LHS' — Log-Harmonic Summation [110], 'SRH' — Summation of Residual Harmonics [111]. For example, PEF method models signal at time *t* in the power spectral density domain with frequency *f* as defined in formula (29):

$$Y_t(f) = \sum_{k=1}^{K} a_{k,t}\delta(fs - kf) + N_t(f) \quad (29)$$

$N_t(f)$– represents the power spectral density of the unwanted noise, $a_{k,t}$ the power of the *k*-th harmonic. *K* is the number of peaks in the audio signal.

In contrast, in LHS method, the signal is modelled as indicated in (30):

$$H(s) = \sum_{n=1}^{N} h_n P(s + \log_2 nc) \quad (30)$$

nc– compression factor, $s = \log_2 f$, h_{nc} – 0.84^{nc-1} is a decreasing sequence implying that higher harmonics contribute less to the pitch than lower harmonics to the noise, $P(s) = W(s) \cdot A(s)$, $W(s)$- spectral window function, $A(s)$- logarithmic frequency abscissa, *N = 15* (number of harmonics that are taken into account).

SRH methods tracks Pitch by using calculations in (31) formula. Please consider the provided references for extra information of NCF and CEP methods.

$$SRH(f) = E(f) + \sum_{k=2}^{N} \left[E(k \cdot f) - E(\left((k - \frac{1}{2})\right) \cdot f)\right] \quad (31)$$

$E(f)$ - amplitude spectrum signal (f – frequency in the range of $[F_{min}, F_{max}]$, computed for each Hanning-windowed frame, covering several cycles of the resulting residual signal) of the *k*-th harmonic, *N* - number of harmonics that are taken into account.

The additional considered method is Mel- frequency cepstral coefficients (MFCC) [112]. MFCC returns the coefficients sampled at a frequency of f*s* Hz as well as the change in coefficients (*delta*) and the change in delta values (*deltaDelta*). *WindowLength* and *OverlapLength* default configuration setup is the same as in the Pitch method. MFCC computes a frequency analysis based on a filter bank with approximately critical band spacing of filters and bandwidths. For 4 kHz bandwidth, approximately 20 filters are used. In most implementations, a short-time Fourier analysis is done first, resulting in a discrete Fourier transform (DFT) for signal $X_t[k]$ in time *t*. DFT values are grouped together in critical bands and weighted by a triangular function [115]. (32), (33) and (34) formulas are used for MFCC calculations (R = 22, *m*-th signal sample, number of MFCC coefficients is usually 13):

$$MF_t[r] = \frac{1}{A_r} \sum_{k=L_r}^{U_r} |V_r[k] \cdot X_t[k]|^2) \quad (32)$$

$MF_t[r]$ - Mel-frequency spectrum at analysis time *t* for $r = 1,2, \ldots R$. $V_r[k]$ is the triangular weighting function for the *r* th filter, ranging from DFT index L_r to U_r.

$$A_r = \sum_{k=L_r}^{U_r} |V_r[k]|^2 \quad (33)$$

A_r - is a normalizing factor for the *r* th Mel-filter.

$$mfcc_t[m] = \frac{1}{R}\sum_{r=1}^{R} \log(MF_t[r]) \cdot \cos[\frac{2\pi}{R}\left(r + \frac{1}{2}\right)m]) \quad (34)$$

Having calculated MFCC as defined in (34), the least-squares approximation of the local slope over a region around the current time sample method is used to determine *delta* (passing MFCC) and *deltaDelta* (passing *delta*). The same rule applies for GTCC (see below).

Similarly, as MFCC, Gammatone cepstral coefficients (GTCC), including *delta* and *deltaDelta*, can be used for audio signal feature extraction. GTCC can basically be defined as a biologically inspired modification of the MFCC that uses Gammatone (GT) filters [112], [113], [114]. GT filter with its properties is defined in formula (35):

$$g(t) = Kt^{(n-1)}e^{-2\pi Bt}\cos(2\pi f_c t + \varphi), t > 0 \quad (35)$$

K is the amplitude factor, *n* is the filter order. f_c is the central frequency in Hertz, φ is the phase shift, *B* represents the duration of the impulse response.

In GTCC extraction, the audio signal is first windowed into short frames, usually about 10–50 ms (same as in MFCC). This allows signal to be stationary for such a short interval, thus facilitating the spectro-temporal signal analysis. Afterwards, GT filter bank (composed of the frequency responses of the several GT filters) is applied to the signal's fast Fourier transform (FFT), emphasizing the perceptually meaningful sound signal frequencies. Lastly, discrete cosine transform (DCT) are applied to model the human loudness perception and decorrelate the logarithmic-compressed filter outputs, thus yielding better energy compaction [114] (36):

$$gtcc_t[m] = \sqrt{\frac{2}{R}}\sum_{r=1}^{R} \log(X_t[r]) \cdot \cos\left[\frac{\pi r}{R}\left(m - \frac{1}{2}\right)m\right]), 1 \le m \le M(5) \quad (36)$$

$X_t[r]$ - is the energy of the signal in the *r* th spectral band, R - is the number of Gammatone filters, M - is the number of GTCC outputs.

Another considered method for speech feature extraction is called wavelet scattering transform (WST). It defines a locally translation invariant representation which is stable to time-warping deformations. WST extends MFCC representations by computing modulation spectrum coefficients of multiple orders through cascades of wavelet convolutions and modulus operators [115], [116]. In addition, WST overcomes MFCC in audio representations for the classification problems at time scales more than 25 *ms*.

Scattering transform recovers the information lost by a Mel-frequency averaging with a cascade of wavelet decompositions and modulus operators. Constant-Q filter banks compute a wavelet transform. A wavelet $\varphi(t)$ is band-pass filter with $\breve{\varphi}(0) = 0$ and is written in the centre frequency ω form as defined in formula (37):

$$\varphi_\omega(t) = \omega \cdot \varphi(\omega t), \breve{\varphi}_w(s) = \breve{\varphi}(\frac{s}{\omega}) \quad (37)$$

Centre frequency of $\check{\varphi}$ is normalized to 1. $\omega = 2^{k/Q}$, Q is the number of wavelets per octave, $k \in Z$. $\check{\varphi}$ is of the order of Q^{-1} to cover the whole frequency axis with these band-pass wavelet filters.

Next, 10 methods for feature extraction in audio signals and auditory spectrograms are defined. These are *spectralSlope* [117], *spectralSkewness*, *spectralSpread*, *spectralCentroid*, *spectralDecrease*, *spectralKurtosis* [118], *spectralFlux* & *spectralRolloff* [119], *spectralFlatness* [120], *spectralEntropy* [121]. Term 'bin', referred in (38) – (47) equations is a segment, e.g., *[fl,fh]* of the frequency axis that collect the amplitude, magnitude or energy from a small range of frequencies.

SpectralSlope is a measure of the slope of the spectral shape. It is calculated by using a linear approximation of the magnitude spectrum, i.e., using a linear regression approach. A linear function is modelled from the magnitude spectrum as defined in (38) equation. The spectral skewness measures the symmetry of the distribution of the spectral magnitude values around their arithmetic mean (39). *SpectralSpread*, describes the concentration of the power spectrum around the spectral centroid and is a rather technical description of spectral shape (40). The spectral centroid represents the centre of gravity (COG) of spectral energy. It is denned as the frequency-weighted sum of the power spectrum normalized by its unweighted sum (41). The spectralDecrease estimates the steepness of the decrease of the spectral envelope over frequency. The result of the spectral decrease is a value < 1. The low results indicate the concentration of the spectral energy at bin 0. The spectral decrease is not defined for audio blocks with no spectral energy (silence) (42). The spectral kurtosis measures whether the distribution of the spectral magnitude values is shaped like a Gaussian distribution or not (43). The spectral flux measures the amount of change of the spectral shape. It is defined as the average difference between consecutive STFT (Short Time Fourier Transform) frames (44). The spectral rolloff is a measure of the bandwidth of the analysed block *n* of audio samples and is specified as the frequency bin below which the accumulated magnitudes of the STFT reach a certain percentage *K* of the overall sum of magnitudes (45). The spectral flatness is the ratio of geometric mean and the arithmetic mean of the magnitude spectrum (46). Entropy can be used to capture the "peakiness" of a probability mass function (PMF). A PMF with sharp peaks has low entropy, while a PMF with flat distribution will have high entropy (47).

$$slope = \left(\sum_{k=b1}^{b2}(f_k - \mu_f)(s_k - \mu_s)\right) / \left(\sum_{k=b1}^{b2}(f_k - \mu_f)^2\right) \quad (38)$$

$$skewness = \left(\sum_{k=b1}^{b2}(f_k - \mu_1)^3 s_k\right) / (\mu_2)^3 \left(\sum_{k=b1}^{b2} s_k\right) \quad (39)$$

$$spread = \sqrt{\left(\left(\sum_{k=b1}^{b2}(f_k - \mu_1)^2 s_k\right) / \left(\sum_{k=b1}^{b2} s_k\right)\right)} \quad (40)$$

$$centroid = \left(\sum_{k=b1}^{b2}(f_k s_k)\right) / \left(\sum_{k=b1}^{b2} s_k\right) \quad (41)$$

$$decrease = \left(\sum_{k=b1+1}^{b2}\frac{s_k - s_{b1}}{k-1}\right) / \left(\sum_{k=b1+1}^{b2} s_k\right) \quad (42)$$

$$kurtosis = \left(\sum_{k=b1}^{b2}(f_k - \mu_1)^4 s_k\right) / (\mu_2)^4 \left(\sum_{k=b1}^{b2} s_k\right) \quad (43)$$

$$flux(t) = \left(\sum_{k=b1}^{b2}|s_k(t) - s_k(t-1)|^P\right)^{1/P} \quad (44)$$

$$rolloff(i) = \sum_{k=b1}^{i} s_k = K\sum_{k=b1}^{b2} s_k \quad (45)$$

$$flatness = \left(\prod_{k=b1}^{b2} s_k\right)^{\frac{1}{b2-b1}} / \left(\frac{1}{b2-b1}\left(\sum_{k=b1}^{b2} s_k\right)\right) \quad (46)$$

$$entropy = \left(-\sum_{k=b1}^{b2} s_k \log(s_k)\right) / \log(b2-b1) \quad (47)$$

f_k- is the frequency in Hz corresponding to bin *k*, μ_f is the mean frequency, s_k is the spectral value at bin *k*, μ_s is the mean spectral value, *b1* and *b2* are the band edges, in bins, over which to calculate the spectral method (e.g,. slope), μ_f is the spectral centroid, μ_2 is the spectral spread. *K* is the percentage of the total energy contained between *b1*, and *i*, *P* is norm type (2 or 1 scalar), *t* – time.

3.2.14. T15: Total Daily Energy Expenditure (TDEE)

In the computerization of T15 (check Figure 3.17 for details), firstly, patient's information is gathered: gender (G – man or woman), age (A_{tdee} - years), height (H_{tdee} in meters), weight (W_{tdee} - kilograms) and physical activity level (PAL – sedentary, light active, moderately active, very active or extremely active), body fat percentage is left optional. Based on these five parameters, Basal Metabolic Rate (BMR) for a patient is calculated by Mifflin St Jeor formula [103] as indicated in (48) and the total daily energy expenditure (TDEE) (49):

$$BMR = \begin{cases} (6.25 \cdot H_{tdee} \cdot 100) + (10 \cdot W_{tdee}) - (5 \cdot A_{tdee}) + 5 & , \ men \\ (6.25 \cdot H_{tdee} \cdot 100) + (10 \cdot W_{tdee}) - (5 \cdot A_{tdee}) - 161 & , \ women \end{cases} \quad (48)$$

$$TDEE = BMR \cdot PAL \quad (49)$$

BMR - Basal Metabolic Rate (Mifflin St Jeor formula), *PAL* – physical activity level (sedentary, light (1–3 days per week), moderate (3–5 days), heavy (6–7 days), athlete (2 times per day), H_{tdee} - height (m), W_{tdee} – weight (kg), A_{tdee} – age (years).

The next part of T15 computerization covers the tracking patient's daily gained and burned calories by using two modes: 2-nd ('Analyse daily food') and 3-

rd ('Enter performed physical activities'). In the 2-nd mode, a patient is instructed to remember what food he was eating during the day, e.g., a single food item is entered by using this approach: product name (string, picked from the predefined list with auto completion functionality), quantity (integer, in grams) and mealtime (breakfast, dinner and supper). The available product list (*foodlist*) is created by using available online educational resources (OER) as defined in online reference list element [104] and is stored in an external file of the mobile app (*foodlist* is needed to calculate calories gained from a single entered product item). A patient can enter as many food items as he remembers from one day. Update, delete functionality of a food item is available as well. When all food items are entered, the patient's formed list (*pfoodlist*) is generated. The total daily gained calories for a patient is calculated as a sum (50):

$$p\_d\_gained = \sum_{i=1}^{n_{tdee}} \frac{m_{patient} \cdot Cal_{100}}{m_{norm}} \tag{50}$$

p_d_gained – patient daily gained calories (from food), n_{tdee} – number of patient's input products, *mpatient* – input product quantity (in grams), *Cal100* – input product calorie norm (for 100 grams), *mnorm* – value of 100 (grams).

In the 3-rd mode, a patient is instructed to remember what physical activities (e.g., working with computer, doing exercises etc.) were performed during the whole day. A single physical activity is entered by using this approach: physical activity name (string, picked from the predefined list with auto completion functionality), duration (minutes) and performance time in day (morning, day, evening). The available activity list (*activitylist*, associated with the relevant metabolic equivalent task (MET) values) is created by using available online educational resources (OER) as defined in online reference list element [105] and is stored in an external file of the mobile app (*activitylist* is needed to calculate calories gained from a single entered physical activity item). Update, delete functionality of the physical activity item is available as well. When all activity items are entered, the patient formed list (*pactivitylist*) is generated. The total daily-burned calories for a patient is calculated as a sum (51):

$$p\_d\_burned = \sum_{i=1}^{m_{tdee}} \frac{MET_i \cdot W_{tdee} \cdot D_i}{60} \tag{51}$$

m_{tdee} – number of patient activities (inputted),
MET_i – item MET coefficient (Metabolic equivalent of task), D_i – *i-th* activity duration in minutes,
p_d_burned – patient's daily-burned calories (from activities), W_{tdee} – patient's weight (in kilograms).

Finally, the daily calorie balance (*p_d_balance*) for a patient is calculated in (52). In order not to lose weight, *p_d_balance* should be close to 0 and *p_d_gained* close to TDEE.

$$p\_d\_balance = p\_d\_gained - p\_d\_burned \tag{52}$$

T15 extracted features (with alias): *t15_tdee_calories*, *t15_patient_dailygained*, *t15_patient_dailylburned*, *t15_tdee_dailycaloriebalance*.

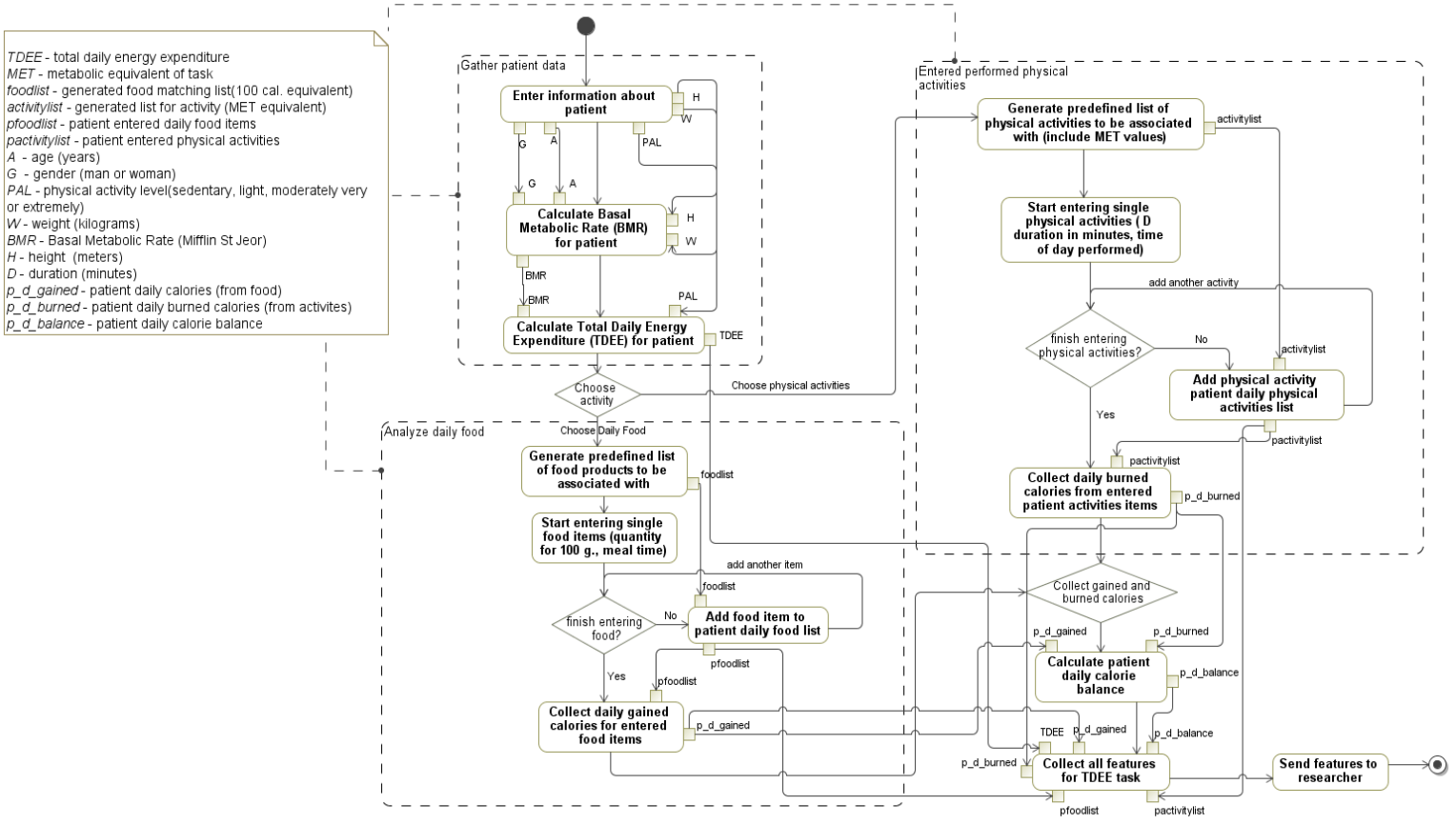


Figure 3.17. Algorithmic implementation of task T15: TDEE

3.2.15. Finger Motion-Tracking Features (FMTF)

Table 3.2. List of finger motion tracking features (for T1, T2, T3, T4, T10, T12, T13).

Notation	Description	Formula	No.
v_{avg}	Average speed of finger while it is moving on the surface (s – travelled path in time *t*)	$v_{avg} = s/\ t$	(53)
v_x	Finger velocity in x -direction (Δs_x – finger position changes in x direction)	$v_x = \Delta s_x/\ t$	(54)
v_y	Finger velocity in y -direction (Δs_y - finger position changes in y direction)	$v_y = \Delta s_y/\ t$	(55)
N_{vx}, N_{vy}	Magnitude of the rate at which pen tip changes its position (x, y directions)	$N_{vx} = v_x\ /\ v_{xmax}$ $N_{vy} = v_y\ /\ v_{ymax}$	(56)
$SD\ (v_x)$, $SD\ (v_y)$	Standard Deviation of velocity (x, y directions, n – finger movement tracking count)	$SD\ (v_x) = \frac{1}{n}\sqrt{\sum_{i=1}^{n}(v_{xi} - v_{avg})^2}$	(57)
a_x	Rate at which finger velocity changes (x direction, n - finger movement tracking count)	$a_x = D_x/\ n$	(58)
a_y	Rate at which finger velocity changes(y direction, n - finger movement tracking count)	$a_y = D_y/\ n$	(59)
a_{maxx}	Maximum finger acceleration in x-direction	a_{maxx} = max $(\sum_{i=1}^{n}(v_{x(i+1)} - v_{xi}))$	(60)
a_{maxy}	Maximum finger acceleration in y-direction	a_{maxy} = max $(\sum_{i=1}^{n}(v_{y(i+1)} - v_{yi}))$	(61)
D_x	Finger position change count in x direction (N_{xi} =1, if $N_{xi} < 0$ and $N_{xi+1} > 0$)	$D_x = \sum_{i=0}^{n} N_{xi}$	(62)
D_y	Finger position change count in y direction (N_{yi} =1, if $N_{yi} < 0$ and $N_{yi+1} > 0$)	$D_y = \sum_{i=0}^{n} N_{yi}$	(63)
φ_{ftmf}	Average *atan2* angle (radians) (between positive x-axis of a plane and point given by coordinates (x, y))	$\varphi_{ftmf} = \mathrm{atan}(y, x)$	(64)
p_n	Logarithmic value of finger travelled path divided by φ_{ftmf}	p_n = log (s / φ)	(65)
p_{avg}	Average finger screen pressure [0;1]	$p_{avg} = s/\ t$	(66)
S_{avg}	Average finger touch area ratio [0;1] (S – finger touch area)	$S_{avg} = S/\ n$	(67)

3.3. Summary of proposed materials and methods

To sum up, the third chapter of this dissertation provides a detailed description of feature extraction algorithms, which are integrated into developed neural impairment test suite (NITS) mobile app for Android OS. In total, 238 features are extracted (refer to Appendice C.) from all 16 tasks. Such tasks were implemented to target four major groups (tremor, cognitive, speech and energy expenditure) of neural impairments, occurring to patients suffering from central nervous system disorders (CNSD). The knowledge for proposed task design was gathered from self-administered gerocognitive examination (SAGE) methodological instrument to identify cognitive impairments (MCI or early dementia) in CNSD. T5, T6, T7, T8, T9, T10, T11, T12 T13 and T0 tasks correspond to the items from the original SAGE questionnaire. Based on the scientific proof of related work findings, the author of this dissertation extended SAGE methodology with six extra tasks for detection of tremor (T1, T2, T3, T4), speech (T14) and energy expenditure (T15) impairments.

Table 3.3 summarizes the feature extraction methods used for neural impairment detection in this dissertation. The computerization of proposed tasks enables automatic health state self-assessment of CNSD patients at home (or medical institution) while using only one smart-interface (mobile app) and shifting from traditional paper based evaluation. In addition, the computerized tasks can track multiple neural impairments, e.g., in the 3-D figure construction task, hand tremors and visuospatial cognitive responses are considered, whereas the standard SAGE question only focus on a single impairment. Table 3.3 shows that not only automatic assignment of the predefined SAGE score (e.g., *tsage7*) is computerized, but the additional algorithms (e.g., Frechet distance, Jaro method, neighbour match graph similarity, FMTF etc.) are adapted by the author of the dissertation as custom routines (or built-in Android SDK functions) based on their mathematical representation. All such listed factors result in novel approach of health state monitoring for CNSD patient.

Table 3.3. Summary of feature extraction methods for targeting neural impairments

Task	Summary of feature extraction methods	Targeted neural impairments (group)
T1 T2 T3	Euclidean distance, *delta* Reaction Time, *rt* Duration FMTF (finger motion-tracking)	Hand movement stability, pressure, accuracy (tremor), visuospatial response speed (cognitive), memory loss (cognitive), speed of thought (cognitive)
T4	Frechet distance, $F(A, B)$ Spiral point match, *mp* Duration FMTF (finger motion-tracking)	Hand movement stability, pressure, accuracy (tremor), visuospatial response (cognitive)

T5 T6	Current date match, *tsage1* Duration	Memory loss, speed of thought (cognitive)
T7	Jaro distance, *JA* (coefficient average) Picture match, *tsage2* Duration	Hand movement stability, pressure, accuracy (tremor), visuospatial abilities, memory loss (cognitive), speed of thought (cognitive)
T8	Item similarity matching, *tsage3* Mathematical subtraction, *tsage4* Mathematical division, *tsage5* Jaro distance, *JA* Duration	Hand movement stability, pressure, accuracy (tremor), memory loss, speed of thought (cognitive)
T9	Neighbour match graph similarity, *GM, tsage7* Parallel line angle calculation, *tsage7* 3-D figure construction speed Number of path trajectories made Edge construction errors statistics Duration FMTF (finger motion-tracking)	Hand movement stability, pressure, accuracy (tremor), visuospatial abilities, memory loss (cognitive)
T10	Analog clock drawing speed Number of path trajectories made Duration Clock drawing error statistics, *tsage8* FMTF (finger motion-tracking)	
T11	Matched items in category, *tsage9* Jaro distance, *JA* Duration	Hand movement stability, pressure, accuracy (tremor), visuospatial abilities, memory loss (cognitive), memory loss (cognitive), speed of thought (cognitive)
T12	Frechet distance, $F(A, B)$ Neighbour match graph similarity, *GM, tsage10* Edge connection errors, *tsage10,* Duration, (seconds) FMTF (finger motion-tracking)	Hand movement stability, pressure, accuracy (tremor), visuospatial abilities, memory loss (cognitive), visuospatial abilities (cognitive), memory loss (cognitive)
T13	Neighbour match graph similarity, *GM, tsage11* Number of erroneous operations,	

	tsage11 Edge connection errors, *tsage11*[1] Duration FMTF (finger motion-tracking)	
T14	Pitch contours Mel- frequency cepstral coefficients (MFCC) Gammatone cepstral coefficients (GTCC) Wavelet scattering transform (WST) Spectral (slope, skewness, spread etc.)	Slurred, imprecise or slower articulation, low volume or weak voice, abnormally long pauses between words etc. (speech)
T15	Basal Metabolic Rate (BMR), Mifflin St Jeor equation, Daily calories gained, cal. Daily calories burned, cal. Daily calories balance, cal.	Weight loss or overweight (energy expenditure)
T0	Matching two text phrases, *tsage6* Duration, seconds	Memory loss, speed of thought (cognitive)

In conclusion, the extracted features play a key role in data analysis and pattern recognition to decision support systems. Such process is covered in the fourth section (experimental research) of this dissertation, i.e., collecting data with the proposed NITS tool from test subjects (both healthy and CNSD patients) as well as implementing methods for grouping individuals into healthy or sick classes.

[1] If SAGE score is listed multiple times for a particular method in a task, it means that the final score is evaluated by accumulating the results of each required method (as a temporal result).

4. EXPERIMENTAL RESEARCH

This section describes the experimental work of the dissertation. Firstly, it covers the data collection process: involved test subjects, data formalization aspects, procedure for the evaluation of CNSD patient health state, number of records gathered in the dataset (dissertation author contribution) and mobile devices that are used for testing. Secondly, the methods for solving sick vs. healthy binary classification problem are defined. These algorithms adapt supervised learning classifiers for clinical decision support. Seventeen supervised learning classifiers (including boost methods) of five types (functions, Bayesian, trees, rules and lazy) are considered in the experimental research. Moreover, the classifier ensemble methods are analysed for the design of a hybrid model, which increases the accuracy of performance as compared to the standalone classifiers. Finally, the proposed models are validated with four experiments to classify test subjects to sick or healthy groups, based on the targeted impairment (e.g., tremor, cognitive, speech, energy expenditure or combination of each).

4.1. Data collection: participants, procedure and dataset

A total number of 15 test subjects from Lithuania were involved in the investigation. 7 patients with neurological disorders (3 Huntington: 2 of them at the age of forties, one of them Juvenile of 18 years, 1 Parkinson (74 years old), 1 after a stroke (60 years), 1 MCI (40 years), 1 cerebral palsy (20 years), other 8 were healthy subjects (age varied from 20 to 78 years). All participants performed the same sets of tasks (16 in total), considering that health state of neurological patients was in the early stage, e.g., HD patients can be described as the early clinical form of HD (I or II in Shoulson-Fahn evaluation system) [17]. The main symptoms of such patients are hand tremors, balance disorders, body and muscle stagnancy, decision-making difficulties [79], problems to focus attention, memory loss, development of early negative energy balance [80] etc. In total, 5 iterations (data collection rounds) were gathered by visiting listed patients multiple times with direct supervision for test execution procedure by the author of the dissertation.

A permit[2] was issued for the data collection formalization and ethical aspects. The informed consent paper form is signed (two copies) with every test subject when conducting the test for the first time. Such document allows using research data for scientific purposes as well as for processing and transferring data to the other countries, maintaining the confidentiality of gathered data. In addition, the license for data collection agreement is presented in the first launch of proposed mobile application (electronic form). Only reliable data, which is approved by the data collection permit of Kaunas University of Technology (KTU) and signed with a test subject on agreement document, is used in this work. Other sources are ignored.

[2] The permit No. IFEP201706-3 for using human subject related materials was issued by the Faculty of Informatics at Kaunas University of Technology.

The data collection procedure is described as follows: researcher, i.e., the creator of a mobile app helps a patient to perform test for the first time by explaining the working principle of each task. In the later stages, if health state allows, a patient works individually (or helped by the family members, medical personnel) and uses his own mobile device. However, a supervised approach of procedure is recommended in order to ensure a fair execution of the test, i.e., without cheating or faking the results. The duration of the test with a single patient is ~20 minutes. Multiple test attempts are required to evaluate the possible progression of the patient's neurological disorder. The second attempt should take into account time scale, e.g., approx. 2–3 weeks (or up to 1 month) after following the execution of the test.

The dataset consist of 150 records. All data was collected in February–August, 2019. Each row represents 238 columns (extracted features). In total, 5 iterations (revision rounds, i.e., patient visitations, each iteration stores ~30 entries in the dataset) were collected from all test subjects during the investigation. All visitations were carried out by using face-to-face communication with patients, providing direct supervision for the test execution. Moreover, the collected data was labelled by using a healthy vs. sick (0 or 1) objective assessment criteria for health status of the test subject. Number 0 indicates that a test subject is healthy, whereas 1 – sick, i.e., has a neurological disorder. In the process of data collection, class label is specified in the mobile application before starting the actual testing procedure and clicking a checkbox on the user profile screen.

The data collection procedure was carried out with Lenovo YOGA YT3-X50L tablet (10.1" screen, resolution 1280 x 800 pixels), SAMSUNG S7 smartphone (5.1", 2560 x 1440) and OnePlus 5 (5.5", 1920 x 1080). The preferred device for carrying out tests is the one with bigger screen size (in this case YOGA YT3-X50L tablet), because patients who have tremor impairments have more difficulties touching particular screen positions accurately. Moreover, the elder test subjects have weakened vision and less experience in using mobile devices.

4.2. Classification models for clinical decision support

As in the author's previous work that was published in Hindawi Journal of Healthcare Engineering [122], a feed-forward backpropagation (FFBP) neural network for total functional capacity (TFC) prediction as the best evaluation with regression *R* value not less than 0.98 and mean squared error (MSE) values of 0.08 was proposed. There, only 2 features were considered (*t1_delta_avg*, *t1_reaction_time_avg*). In a recent publication in IEEE Journal of Biomedical and Health Informatics [123], the features (without FMTF) from T1, T2 and T3 were combined resulting in binary back-propagation neural network (BPNN) classifier with achieved accuracy of 86.4% (F-measure 0.859) for the recognition of early, prodromal symptoms to patients with central nervous system motor disorders.

4.2.1. Supervised learning classifiers

For the decision support making of health state evaluation, the author targets a wide range of existing machine learning (ML) methods. Four main types of ML algorithms can be defined for dealing with classification (categorization of certain instances of types) problems [164] [165]: supervised (input data has a known label), semi-supervised (not all data instances has labels), unsupervised (input data has no label) and reinforcement learning (labels are gathered from the environment in an iterative fashion). Three latter approaches are excluded, i.e., leaving with supervised learning, as labelled data (0 or 1) was collected from patients with direct supervision for the test execution.

In addition, the addressed classification problematics in this dissertation is considered to be of innovative and non-linear origin because of uniquely formed tremor, cognitive and energy expenditure feature combinations for model inputs. In such case, the author of this dissertation chooses a strategy to investigate a number of the most commonly used supervised learning algorithms and search for an appropriate solution. Firstly, five groups of supervised learning methods are examined: functions, Bayesian, trees, rules and lazy classifiers [147] [148]. Function group classifiers can be defined by using mathematical equations. Bayesian group classifiers are based on Bayes Theorem of Probability [169] and allocate the element value to a population from one of the categories that is available. Tree group classifiers apply graph theory that makes use of branching methodology to exemplify all possible outcomes of a decision, based on a certain condition. Supervised learning based on rules is a method to represent knowledge of the system by a set of expressions (e.g., IF: THEN). Lazy classifiers store the training instances and do real work only when the actual classification starts.

The author of this dissertation analyses the most commonly used supervised learning algorithms as referenced in [170] and [171] as well examining the extra methods (to get a better intuition for the proposed classification problem), which deal with nominal (categorical) attributes, i.e., patient health state evaluator (sick or healthy). The considered classifiers are: Support Vector Machines (SVM) [124] [125], Artificial Neural Networks (ANN) with multilayer perceptron (MLP) [127], K-nearest neighbours (K-NN) [126], John Platt's sequential minimal optimization algorithm for training a support vector classifier (SMO), [128] [129] [130]. In addition, these methods are investigated: Linear Discriminant Analysis (LDA) [131], Fisher's Linear Discriminant Analysis (FLDA) [132], Deep Learning Networks (DNN) [133], Random Forests [134], Bayes Nets [135], Naive Bayes [136]. Moreover, the author adapts Decision Tree (J48) [137], Stochastic Gradient Descent (SGD), [138] Logistic Model Trees (LMT) [139] [140], Decision Stump [141], Voted Perceptron [142], Logistic Regression [163], Decision Tables [166] and Locally Weighted Learning (LWL) [167] [168] classifiers.

Moreover, the boosting algorithms [144] are adapted for classifier ensemble learning. For attribute selection (correlation) methods, 4 of them are investigated: Principal Component Analysis (*PCA*) [145] for dimensionality reduction, *CorrelationAttributeEval* [146] for the evaluation of attribute relationships to target class, *WrapperSubsetEval* [143] and *ClassifierAttributeEval* (to evaluate the worth of an attribute by using a user-specified classifier). The attribute to search methods using *CfsSubsetEval* (evaluation of the worth of a subset of attributes by considering the individual predictive ability of each feature along with the degree of redundancy between them) algorithm are *Best-First* (Forward, Backward, Greedy) and *Ranker* [147], [148].

4.2.2. Classifier ensemble algorithm (hybrid model)

The hybrid classification model combines strengths of the different knowledge representation types, e.g., applying fusion technique of function and Bayesian methods. The motivation for designing a hybrid classification model is based on the expectance that classifier fusion strategy will provide more accurate classification decision [149]. The insights referenced in [172] define that instead of looking for the best set of features and classifiers, the best set of classifiers and their combination methods should be considered as a priority.

Three reasons are listed why the classifier ensemble approach has advantages over a single classifier [173]: 1) statistical, 2) computational, 3) representational. The statistical approach addresses the presumption that aggregating a number of different classifiers will bring the resultant classifier closer to the one that best (optimal) fits the problem (as well as performs good on the training set) than a randomly chosen classifier. The computational reason assumes that the training process of each individual classifier starts somewhere in the space of possible classifiers and ends closer to the optimal classifier. Thus, the aggregation procedure leads to better approximation to optimal classifier than a single one. The representational approach deals with scenarios when the considered classifier space does not contain an optimal classifier, i.e., if only linear algorithms are considered, but the best classifier is of non-linear origin; then, the classifier fusion can approximate any decision boundary with any predefined accuracy [149].

As for the hybridization approach in supervised learning algorithms, the method of fusing class label outputs is required. In this dissertation, *average of probability* (69) combination rule is considered [174] [175]. In each case, the final ensemble decision $H(X)$ is the class i that receives the largest computed probability $PMAX_i(X)$ (68).

$$H(X) = \max_{i=1..n}(PMAX_i(X)) \tag{68}$$

$$PMAX_i(X) = \frac{1}{m}\sum_{j=1}^{m} P_j(\,(w_i \mid x) \tag{69}$$

n – number of classes (targets), m – number of classifiers used in ensemble (fusion) methods, $P_j(\,(w_i \mid x)$ – probability of i-th data instance to belong to class x, evaluated by the j-th classifier.

In this work, the hybrid model is applied for tremor, cognitive and energy expenditure impairment classification as defined in E2. The motives for selecting E2 for combining classifiers is that in this experiment, the standalone classifiers are treated to be more precise evaluators (with all integrated features) and produce more accurate results (as compared to E1). Moreover, E3 and E4 experiments are provided as supplementary material for triggering alarms only about speech impairments.

Seventeen different supervised learning classifiers are targeted in this dissertation. For hybrid model design, such scenario includes considering 17! (355687428096000) of possible combinations in fusion process. A combination is meaningful if chosen classifier methods in a single ensemble give increased accuracy as compared to a particular standalone classifier. Table 4.1 shows a heuristic approach of randomly choosing 50 different combinations and comparing accuracy results. The order of classifiers in the proposed ensemble is not important. The author notes that such investigation is only a pilot study; however, multiple combinations were found that improve the classification accuracy results (in this context treated as local optimum findings). Refer to Table 4.3 and Table 4.4 of this section for detailed statistical evaluation of single classifiers and proposed hybrid model. The best-achieved accuracy of 96.12% was obtained in nine combinations (4, 34, 38, 40, 41, 42, 43, 44 and 47). Adding more classifiers to single ensemble does not necessarily improve the performance of the model, i.e., combination 3 illustrates that adding all 17 methods slightly decrease the accuracy. However, a smaller number of classifiers in the highest accuracy ensemble is preferred as a model-training time is prone to be faster.

Table 4.1. Investigation of selecting classifier combinations for proposed hybrid model. Plus (+) sign indicates a combination of separate classifiers (each of methods in usage process is wrapped in [] brackets). The accuracy is provided for the integrated feature set (N=10 cross validation procedure). CC – number of classifiers used in ensemble. The bolded combination number indicates combinations with the best achieved accuracy = 96.12%.

No.	CC	Classifier combination	Accuracy %
1	7	[AdaBoostM1(Decision Stump)] + [AdaBoostM1 (Random Forest)] + [AdaBoostM1 (MLP)] + [AdaBoostM1 (SMO)] + [AdaBoostM1 (kNN)] + [AdaBoostM1] (LWL)] + [AdaBoostM1 (BayesNet)]	95.34
2	4	[AdaBoostM1(Decision Stump)] + [AdaBoostM1 (Random Forest)] + [AdaBoostM1(J48)] + [AdaBoostM1 (LMT)]	93.79
3	17	[AdaBoostM1(Decision Stump)] + [AdaBoostM1 (Random Forest)] + [AdaBoostM1 (MLP)] + [AdaBoostM1 (SMO)] + [SGD] [AdaBoostM1 (kNN)] + [AdaBoostM1] (LWL)] + [AdaBoostM1] (J48)] + [AdaBoostM1] (Naïve Bayes)]+ [AdaBoostM1 (BayesNet)] + [FLDA(CfsSubseEval)] + [SVM (linear) with PCA] +[SVM (sigmoid) with PCA (2 times)] +	93.02

		[SVM (linear) with PCA] + [DNN (LSTM)] + [Decision Table] + [Voted Perceptron with PCA]	
4	12	[AdaBoostM1(Decision Stump)] + [AdaBoostM1 (Random Forest)] + [AdaBoostM1 (MLP)] + [AdaBoostM1 (SMO)] + [AdaBoostM1 (kNN)] + [AdaBoostM1] (LWL)] + [AdaBoostM1 (BayesNet)] + [SVM (sigmoid) with PCA (2 times)] + [SVM (linear) with PCA] +[DNN (LSTM)] + [Voted Perceptron with PCA]	96.12
5	2	[AdaBoostM1 (BayesNet)] + [AdaBoostM1] (Naïve Bayes)]	93.02
6	2	[AdaBoostM1 (kNN)] + [AdaBoostM1] (LWL)]	93.79
7	9	[AdaBoostM1 (MLP)] + [AdaBoostM1 (SMO)] + [SGD] + [FLDA(CfsSubseEval)] + [SVM (linear) with PCA] +[SVM (sigmoid) with PCA (2 times)] + [SVM (linear) with PCA] + [DNN (LSTM)] + [Voted Perceptron with PCA]	90.67
8	4	[FLDA(CfsSubseEval)] + [SVM (linear) with PCA] + [SVM (sigmoid) with PCA (2 times)] + [SVM (linear) with PCA] +	95.34
9	3	[SVM (linear) with PCA] + [SVM (sigmoid) with PCA (2 times)] + [SVM (linear) with PCA] +	92.24
10	4	[FLDA(CfsSubseEval)] + [AdaBoostM1 (kNN)] + [AdaBoostM1 (Random Forest)] + [SVM (sigmoid) with PCA]	95.34
11	3	[AdaBoostM1 (MLP)] + [SVM (sigmoid) with PCA] + [SVM (linear) with PCA]	93.79
12	2	[AdaBoostM1 (MLP)] + [AdaBoostM1 (Random Forest)]	93.02
13	2	[FLDA(CfsSubseEval)] + [AdaBoostM1 (Random Forest)]	94.57
14	2	[SVM (sigmoid) with PCA]+ [AdaBoostM1 (Random Forest)]	95.34
15	2	[SVM (sigmoid) with PCA]+ [SGD]	94.57
16	3	[SVM (sigmoid) with PCA]+ [SGD] + [DNN (LSTM)]	93.79
17	2	[SVM (sigmoid) with PCA]+ [DNN (LSTM)]	95.34
18	3	[SVM (sigmoid) with PCA]+ [SVM (linear) with PCA] + [DNN (LSTM)]	92.24
19	4	[AdaBoostM1(Decision Stump)] + [DNN (LSTM)] + [SVM (sigmoid) with PCA]+ [SVM (linear) with PCA]	93.02
20	2	[AdaBoostM1(Decision Stump)] + [Voted Perceptron with PCA]	80.62
21	2	[AdaBoostM1(Random Forest)] + [DNN (LSTM)]	65.11
22	2	[DNN (LSTM)] + [AdaBoostM1(Naïve Bayes)]	85.27
23	4	[DNN (LSTM)] + [AdaBoostM1(Naïve Bayes)] + [SGD] + [Decision Table]	93.02
24	5	[DNN (LSTM)] + [AdaBoostM1(Naïve Bayes)] + [SGD] + [Decision Table] + [FLDA(CfsSubseEval)]	93.02

25	3	[AdaBoostM1(Naïve Bayes)] + [Decision Table] + [FLDA(CfsSubseEval)]	87.59
26	2	[AdaBoostM1(Bayes Net)] + [AdaBoostM1(MLP)]	93.79
27	4	[SVM (sigmoid) with PCA (2 times)] + [SVM (linear) with PCA] [DNN (LSTM)] + [Voted Perceptron with PCA]	93.02
28	8	[AdaBoostM1(Decision Stump)] + [AdaBoostM1 (Random Forest)] + [AdaBoostM1 (MLP)] + [AdaBoostM1 (SMO)] + [AdaBoostM1 (kNN)] + [AdaBoostM1] (LWL)] + [AdaBoostM1 (BayesNet)] + [DNN (LSTM)]	95.34
29	9	[AdaBoostM1(Decision Stump)] + [AdaBoostM1 (Random Forest)] + [AdaBoostM1 (SMO)] + [AdaBoostM1 (kNN)] + [AdaBoostM1] (LWL)] + [AdaBoostM1 (BayesNet)] + [SVM (sigmoid) with PCA (2 times)] + [SVM (linear) with PCA] + [Voted Perceptron with PCA]	95.34
30	9	[AdaBoostM1(Decision Stump)] + [AdaBoostM1 (SMO)] + [AdaBoostM1 (kNN)] + [AdaBoostM1] (LWL)] + [AdaBoostM1 (BayesNet)] + [SVM (sigmoid) with PCA (2 times)] + [SVM (linear) with PCA] + [Voted Perceptron with PCA] + [FLDA(CfsSubseEval)]	95.34
31	4	[AdaBoostM1 (BayesNet)] + [SVM (sigmoid) with PCA (2 times)] + [SVM (linear) with PCA] + [Voted Perceptron with PCA]	95.34
32	4	[AdaBoostM1 (SMO)] + [DNN (LSTM)] + [SVM (sigmoid) with PCA (2 times)] + [SVM (linear) with PCA]	94.57
33	3	[AdaBoostM1 (SMO)] + [SVM (sigmoid) with PCA (2 times)] + [SVM (linear) with PCA]	94.57
34	4	[AdaBoostM1 (SMO)] + [SVM (sigmoid) with PCA (2 times)] + [SVM (linear) with PCA] + [FLDA(CfsSubseEval)]	96.12
35	5	[AdaBoostM1 (SMO)] + [SVM (sigmoid) with PCA (2 times)] + [SVM (linear) with PCA] + [FLDA(CfsSubseEval)] + [SGD]	93.79
36	5	[AdaBoostM1 (SMO)] + [SVM (sigmoid) with PCA (2 times)] + [SVM (linear) with PCA] + [FLDA(CfsSubseEval)] + [Logistic Regression]	94.57
37	5	[AdaBoostM1 (SMO)] + [SVM (sigmoid) with PCA (2 times)] + [SVM (linear) with PCA] + [AdaBoostM1 (Naïve Bayes)] + [Voted Perceptron with PCA]	94.57
38	5	[AdaBoostM1 (SMO)] + [SVM (sigmoid) with PCA (2 times)] + [SVM (linear) with PCA] + [AdaBoostM1 (LWL)] + [Voted Perceptron with PCA]	96.12
39	5	[AdaBoostM1 (SMO)] + [SVM (sigmoid) with PCA (2 times)] + [SVM (linear) with PCA] + [AdaBoostM1 (Bayes Net)] + [Voted Perceptron with PCA]	95.34
40	5	[AdaBoostM1 (SMO)] + [SVM (sigmoid) with PCA (2 times)] + [SVM (linear) with PCA] + [AdaBoostM1 (MLP)] + [Voted Perceptron with PCA]	96.12
41	5	[AdaBoostM1 (SMO)] + [SVM (sigmoid) with PCA (2 times)] + [SVM (linear) with PCA] + [AdaBoostM1 (LMT)] + [Voted Perceptron with PCA]	96.12
42	5	[AdaBoostM1 (SMO)] + [SVM (sigmoid) with PCA (2 times)] + [SVM	96.12

		(linear) with PCA] + [AdaBoostM1 (Random Forest)] + [Voted Perceptron with PCA]	
43	4	[SVM (sigmoid) with PCA (2 times)] + [SVM (linear) with PCA] + [AdaBoostM1 (J48)]+ [Voted Perceptron with PCA]	96.12
44	6	[SVM (sigmoid) with PCA (2 times)] + [SVM (linear) with PCA] + [AdaBoostM1 (J48)]+ [Voted Perceptron with PCA] + [AdaBoostM1 (SMO)] + [Voted Perceptron with PCA]	96.12
45	5	[SVM (sigmoid) with PCA (2 times)] + [AdaBoostM1 (J48)]+ [Voted Perceptron with PCA] + [AdaBoostM1 (SMO)] + [Voted Perceptron with PCA]	94.57
46	2	[AdaBoostM1 (SMO)] + [Voted Perceptron with PCA]	93.02
47	3	[AdaBoostM1 (SMO)] + [Voted Perceptron with PCA] + [AdaBoostM1 (Bayes Net)]	96.12
48	3	[AdaBoostM1 (SMO)] + [Voted Perceptron with PCA] + [AdaBoostM1 (Naïve Bayes)]	93.02
49	3	[AdaBoostM1 (SMO)] + [Voted Perceptron with PCA] + [AdaBoostM1 (Random Forest)]	95.34
50	3	[AdaBoostM1 (SMO)] + [Voted Perceptron with PCA] + [FLDA(CfsSubseEval)]	95.34

For choosing the best-fit combination, the model speed evaluation metric is considered as well. The goal is to find the highest accuracy ensemble, which is trained with the minimum time expenditure. Such investigation is provided in Table 4.2. The most accurate combinations (4, 34, 38, 40, 41, 42, 43, 44 and 47) from Table 4.1 are taken for further speed analysis. The hypothesis that smaller number of classifiers used in the ensemble will improve the speed performance is confirmed (combination no. 47).

Table 4.2. Investigation of speed performance for combinations with the best achieved accuracy =96.12%. CC – number of classifiers used in ensemble. Speed is provided in seconds. The bolded combination number indacates the best combination.

No.	CC	Classifier combination	Speed (s)
4	12	[AdaBoostM1(Decision Stump)] + [AdaBoostM1 (Random Forest)] + [AdaBoostM1 (MLP)] + [AdaBoostM1 (SMO)] + [AdaBoostM1 (kNN)] + [AdaBoostM1] (LWL)] + [AdaBoostM1 (BayesNet)] + [SVM (sigmoid) with PCA (2 times)] + [SVM (linear) with PCA] +[DNN (LSTM)] + [Voted Perceptron with PCA]	40.53
34	4	[AdaBoostM1 (SMO)] + [SVM (sigmoid) with PCA (2 times)] + [SVM (linear) with PCA] + [FLDA(CfsSubseEval)]	0.71
38	5	[AdaBoostM1 (SMO)] + [SVM (sigmoid) with PCA (2 times)] + [SVM (linear) with PCA] + [AdaBoostM1 (LWL)] + [Voted Perceptron with PCA]	15.54
40	5	[AdaBoostM1 (SMO)] + [SVM (sigmoid) with PCA (2 times)] + [SVM (linear) with PCA] + [AdaBoostM1 (MLP)] + [Voted Perceptron with PCA]	25.57

41	5	[AdaBoostM1 (SMO)] + [SVM (sigmoid) with PCA (2 times)] + [SVM (linear) with PCA] + [AdaBoostM1 (LMT)] + [Voted Perceptron with PCA]	2.14
42	5	[AdaBoostM1 (SMO)] + [SVM (sigmoid) with PCA (2 times)] + [SVM (linear) with PCA] + [AdaBoostM1 (Random Forest)] + [Voted Perceptron with PCA]	0.88
43	4	[SVM (sigmoid) with PCA (2 times)] + [SVM (linear) with PCA] + [AdaBoostM1 (J48)]+ [Voted Perceptron with PCA]	0.62
44	6	[SVM (sigmoid) with PCA (2 times)] + [SVM (linear) with PCA] + [AdaBoostM1 (J48)]+ [Voted Perceptron with PCA] + [AdaBoostM1 (SMO)] + [Voted Perceptron with PCA]	1.02
47	3	[AdaBoostM1 (SMO)] + [Voted Perceptron with PCA] + [AdaBoostM1 (Bayes Net)]	0.59

The innovative aspects of a designed hybrid model is author's proposition of uniquely combined classifier set for achieving improved accuracy results as compared to the single models. Another innovative aspect of the implemented hybrid model is the usage of ensemble learning based classifiers (e.g., AdaBoostM1) as individual model components for building a resulting ensemble classifier. Configuration setup of 47-th combination (Figure 4.1) is fusing AdaBoostM1 (SMO), AdaBoostM1 (BayesNet) and Voted Perceptron + PCA) with Vote method (Average of Probabilities combination rule) [149]. In such scenario, each sub model generates predictions (*p_classifier_name*) on the provided test data (probabilities of assigning each instant to a class), then taking the mean of the probability distributions for each of the base classifiers, allowing each sub-model to vote on what the outcome should be.

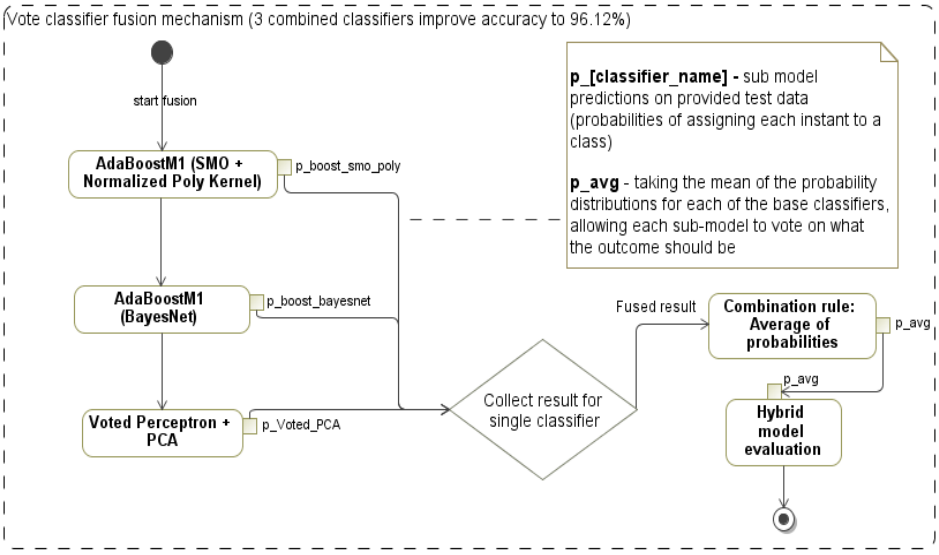


Figure 4.1. Schema of proposed hybrid classification model (3 classifiers)

4.3. Conducted experiments for solving sick vs. healthy classification problem

In this dissertation, a fusion strategy is provided, i.e., all extracted materials (features) are considered for healthy vs. sick (two target classes, 0 vs. 1) test subject evaluation. Thus, 4 experiments (all of them are adapted on own built-in dataset) were carried out:

1. Experiment1 (E1): feature set is distributed to individual tasks only (14 different classifiers);
2. Experiment2 (E2): all 238 features are combined (integrated) and fed to a classifier, the combinations of classifiers are used to propose a hybrid model;
3. Experiment3 (E3): audio files (from T14 task) are used for further processing to extract features with a combination of methods (Pitch, MFCC, GTCC and spectral, e.g., skewness) and classify samples with deep learning networks (in particular, the recurrent neural network (RNN) with Long short-term memory (LSTM));
4. Experiment4 (E4): audio files (from T14 task) are used for further processing to extract features with WST method and classify samples with support vector machine (SVM).

E1 and E2 were implemented with Weka tool for data mining [153], whereas E3 and E4 by using MATLAB Audio Toolbox R2019a software.

4.3.1. E1: Sick vs. Healthy classification models for individual tasks

Such experiment was designed for triggering alarms (alerts), based only on a single task that is executed by a test subject (sick or healthy). In addition, E1 is innovative for uniquely composed feature combinations (subsets) for the determination of health state evaluation. Table 4.3 provides the best achieved results for each individual task. Higher accuracy indicates that a model can distinguish between two target classes better. For example, LMT classifier, combined with PCA, results in 91.50% accuracy for T9 task. Other classifiers achieve slightly lower results (the lowest are *Spelling* and *T11* = 74.52%), which means that either extra indicators (evaluation criteria) are needed, or more data should be fed for training a model. If a classifier is in brackets of the other classifier, *WrapperSubsetEval* method was applied for training the selected attributes. The model building speed is provided in seconds on CPU. All trained models were tested by using N-Cross (N=10) validation procedure [151], [152, [150]. For a complete feature list that is used for the proposed classifiers, refer to Appendice C.

Table 4.3. Healthy vs. Sick classification results of individual tasks

Task: Feature Count	Attribute Selection Method	Classifier Accuracy % (N-cross validation), N =10	Model Speed (s)
T1: **9**	PCA (Principal Component Analysis)	J48 = 84.90%, SVM (RBF) = 84.90%	Instant
T2: **10**	WrapperSubsetEval	J48 (Naive Bayes Multinomial) = 81.13%	0.07
T3: **28**	WrapperSubsetEval	Random Forest (k - NN) = 77.35%	1.72
T4 (spiral contour following): **22**	WrapperSubsetEval	Logistic Regression (LDA) = 84.90%, ANN (Random Forest) = 82.07%	1.49 36.06
T4 (spiral contour drawing): **22**	WrapperSubsetEval	ANN (k - NN) = 87.73%, Random Forest (FLDA) = 86.79%	2.16 1.24
T9: **30**	PCA, CorrelationAttributeEval	LMT = 91.50%, ANN = 90.56%, Random Forest =90.56%	0.04, 0.07 0.02
T10: **24**	CorrelationAttributeEval, WrapperSubsetEval	k-NN = 90.56%, ANN (k-NN) = 89.62%	Instant 2.36
T11: **2**	CorrelationAttributeEval	Random Forest =74.52%	0.02
T12: **33**	CorrelationAttributeEval, WrapperSubsetEval	Random Forest= 83.09%, ANN (k-NN) = 82.07%	0.03 5.84
T13: **25**	WrapperSubsetEval	J48 (FLDA) = 83.96%	0.50
T15: **4**	CorrelationAttributeEval	SVM (Radial Basis Function) = 78.3%	Instant
Spelling (T7,T8, T11): **3**	WrapperSubsetEval	LMT (Random Forest) = 74.52%	2.76
SAGE (T6, T7, T8, T9, T10, T11, T12, T13, T0): **10**	PCA, CorrelationAttributeEval	LMT = 84.90% SMO = 84.90%	0.01 0.03
Duration (all tests): **16**	WrapperSubsetEval	FLDA (FLDA) = 89.62%	0.66

4.3.2. E2: Sick vs. Healthy classification models for integrated feature set

The dataset of all collected records for E2 is split into 129 for training (N-Cross validation procedure) by omitting records of randomly chosen 3 patients (1 Juvenile Huntington Disease, 1 Parkinson Disease, 1 MCI) and 3 healthy test subjects. These omitted records were supplied as 2 individual test sets (1 for healthy (H), 1 for sick(S)) for predictions on new (unseen) data samples (healthy vs. sick classification). Boosting algorithm *AdaBoostM1* [144] is considered for tuning classifier accuracy. The evaluation metrics for designed models are True Positive Rate (*TPR*), False Positive Rate (*FPR*), *Precision*, *Recalll* (same calculation as *TPR*), *F1* score (F-Measure) [155]. Moreover, Matthews correlation coefficient (*MCC*), [154], *AUC* (area under Receiver Operating Characteristics (*ROC*), i.e., a plot of the true positive fraction vs. false positive fraction (= 1 – TPR) for all potential cut-offs for a representation), *PRC* (precision-recall plot, better adapted for imbalanced datasets). In equations (70–74), the term *TP* refers to true positive, i.e., the number of test subjects with neurological disorder who have a positive test result. *TN* (true negative) is the number of subjects without disease who have a negative test result. *FP* (False positive) is a number of subjects without disease who have a positive test result. *FN* (False negative) is a number of subjects with disease who have a negative test result.

$$\mathrm{TPR} = \frac{\mathrm{TP}}{(\mathrm{TP} + \mathrm{FN})} \tag{70}$$

$$\mathrm{FPR} = \frac{\mathrm{FP}}{(\mathrm{FP} + \mathrm{TN})} \tag{71}$$

$$\mathrm{Precision} = \frac{\mathrm{TP}}{(\mathrm{TP} + \mathrm{FP})} \tag{72}$$

$$\mathrm{F1} = 2 \cdot \frac{\mathrm{Precision} \cdot \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}} \tag{73}$$

$$\mathrm{MCC} = \frac{\mathrm{TP} \cdot \mathrm{TN} - \mathrm{FP} \cdot \mathrm{FN}}{\sqrt{(\mathrm{TP} + \mathrm{FP})(\mathrm{TP} + \mathrm{FN})(\mathrm{TN} + \mathrm{FP})(\mathrm{TN} + \mathrm{FN})}} \tag{74}$$

Table 4.4. shows a comparison of the best-achieved results by using TPR, FPR, Precision, F1, MCC metrics with 17 classifiers (as well as their combinations with boost algorithms or attribute selection methods) for experiment E2. The results are presented as weighted average values for both sick and healthy target classes.

AdaBoostM1 algorithm is applied for particular classifiers only if it takes effect on the overall classifier accuracy performance.

For statistical evaluation of proposed classifiers (refer to Table 4.5), 5 error metrics are used: Kappa coefficient *KP* [156], mean absolute error (MAE), root mean squared error (RMSE), relative absolute error (RAE), root relative squared error (RRSE) as denoted in equations for individual model *i* (75-80). For *KP*, the ideal case is value = 1; whereas, MAE, RMSE, RAE and RRSE ideal value is 0. CCI indicates the number of correctly classified instances, ICI – number of incorrectly classified instances. C0_C is the number of correctly classified healthy test subjects (target class = 0), C0_I – number of incorrectly classified healthy test subjects (target class = 0), C1_C – number of correctly classified sick test subjects (target class = 1), C1_I – number of incorrectly classified sick test subjects (target class = 1).

The designed hybrid model shows an improvement of statistical evaluation metrics. The accuracy (N=10 cross validation procedure applied) = 96.124%, TPR = 0.961, FPR = 0.054, Precision = 0.962, F1 = 0.961, MCC = 0.919, ROC = 0.994, PRC = 0.994. However, the model building is slower, i.e., 51.98 seconds. The error statistics: K = 0.919, MAE =0.161, RMSE = 0.217, RAE = 28.274%, RRSE = 42.163%, CCI = 124, ICI = 5, C0_C = 78, C0_I = 1, C1_C = 46, C1_I = 4.

$$KP = \frac{p_0 - p_e}{1 - p_e} = 1 - \frac{1 - p_0}{1 - p_e} \tag{75}$$

p_0- is the relative observed agreement among raters, p_e- is the hypothetical probability of chance agreement, using the observed data to calculate the probabilities of each observer who randomly seeing each category.

$$MAE_i = \frac{1}{n}\sum_{j=1}^{n}\left|P_{(i,j)} - T_j\right| \tag{76}$$

$$RMSE_i = \sqrt{\frac{1}{n}\sum_{j=1}^{n}\left(P_{(i,j)} - T_j\right)^2} \tag{77}$$

$$RAE_i = \left(\sum_{j=1}^{n}\left|P_{i,j} - T_j\right|\right) / \left(\sum_{j=1}^{n}\left|T_j - \overline{T_j}\right|\right) \tag{78}$$

$$RRSE_i = \sqrt{\left(\left(\sum_{j=1}^{n}\left(P_{(i,j)} - T_j\right)^2\right) / \left(\sum_{j=1}^{n}\left(T_j - \overline{T_j}\right)^2\right)\right)} \tag{79}$$

$$\overline{T_j} = \frac{1}{n}\sum_{j=1}^{n} T_j \tag{80}$$

$P_{i,j}$- is the value predicted by the individual model *i* for sample *j* (out of *n* observations), T_j is the target value for sample j.

Table 4.4. Healthy vs. Sick classification results for the unified classifier (accuracy metrics, N=10 cross validation procedure)

Classifier	Classifier Type	Accuracy %	TPR	FPR	Precision	F1	MCC	ROC	PRC	Speed (s)
AdaBoostM1 (Decision Stump)	Boost+Tree	93.023	0.930	0.081	0.930	0.930	0.850	0.986	0.987	0.11
AdaBoostM1 (RandomForest)	Boost+Tree	94.573	0.946	0.078	0.948	0.945	0.887	0.990	0.990	0.13
AdaBoostM1 (J48)	Boost+Tree	86.046	0.860	0.169	0.860	0.859	0.703	0.816	0.784	0.01
Logistic Regression	Function	89.922	0.899	0.100	0.901	0.900	0.791	0.972	0.975	0.03
AdaBoostM1 (LMT)	Boost+Tree	91.472	0.915	0.083	0.917	0.915	0.823	0.981	0.983	1.41
SGD	Function	91.472	0.915	0.091	0.915	0.915	0.821	0.912	0.881	0.07
AdaBoostM1 (NaiveBayes)	Boost+ Bayes	88.372	0.884	0.103	0.891	0.885	0.776	0.921	0.925	0.26
AdaBoostM1 (ANN-MLP)	Boost+Function	92.480	0.922	0.086	0.922	0.922	0.837	0.971	0.971	19.46
AdaBoostM1 (SMO)	Boost+Function	92.248	0.922	0.078	0.923	0.923	0.838	0.967	0.967	0.36
AdaBoostM1 (kNN)	Boost+Lazy	94.573	0.946	0.071	0.946	0.945	0.886	0.933	0.918	0.03
AdaBoostM1 (LWL)	Boost+Lazy	91.472	0.915	0.091	0.915	0.915	0.821	0.972	0.973	13.62
AdaBoostM1 (BayesNet)	Boost+Bayes	93.798	0.938	0.083	0.939	0.937	0.869	0.958	0.962	0.31
SVM (sigmoid) + PCA	Function	91.472	0.915	0.091	0.915	0.915	0.821	0.912	0.881	0.24
SVM (linear) + PCA	Function	92.248	0.922	0.086	0.922	0.922	0.837	0.918	0.890	0.23
(LDA) FLDA+ CfsSubsetEval	Function	92.248	0.922	0.100	0.923	0.922	0.836	0.976	0.978	2.94
Decision Table	Rules	86.821	0.868	0.172	0.869	0.866	0.720	0.905	0.879	0.31
DNN (LSTM)	Function	94.573	0.946	0.056	0.946	0.946	0.886	0.987	0.987	2.80
Voted Perceptron + PCA	Function	93.023	0.930	0.081	0.930	0.930	0.853	0.937	0.920	0.30
Hybrid (proposed by the author)	Fusion of classifiers	96.124	0.961	0.047	0.961	0.961	0.918	0.983	0.984	0.59

Figure 4.2 illustrates the error visualization and ROC curve for SVM (linear kernel) classifier combined with PCA: blue 'X' indicates the correctly classified healthy individuals (equivalent to target class 0), red 'X' is correctly classified patients with neurological disorders (equivalent to target class 1). Blue square indicates incorrectly classified healthy persons, red square incorrectly classified patients with neurological disorders.

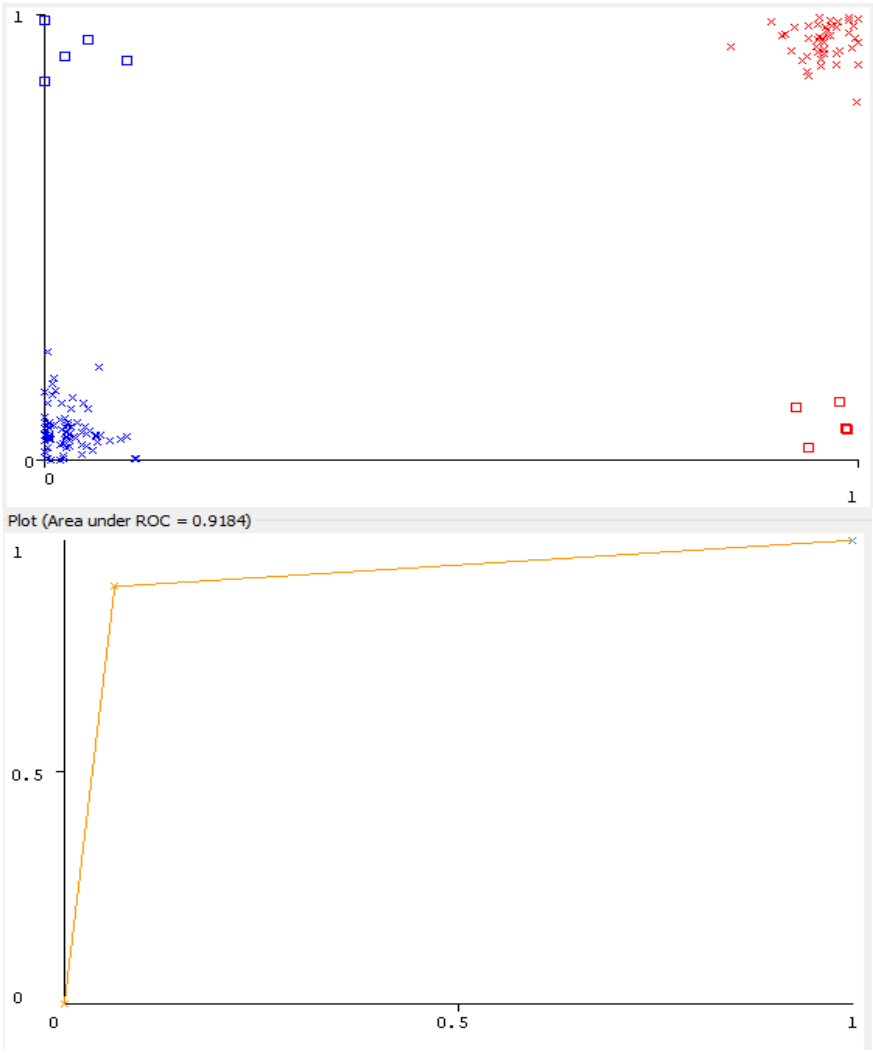


Figure 4.2. Statistical evaluation of the unified classifier. Illustration model: SVM (linear kernel) + PCA. Error visualization (top side, 10 incorrect classifications), ROC curve =0.918

Table 4.5. Healthy vs. Sick classification results for the unified classifier (all extracted features combined for the evaluation of error statistics)

Classifier	KP	MAE	RMSE	RAE %	RRSE %	CCI	ICI	C0_C	C0_I	C1_C	C1_I
AdaBoostM1 (Decision Stump)	0.852	0.07	0.225	15.695	46.204	120	9	75	4	45	5
AdaBoostM1 (RandomForest)	0.883	0.243	0.284	51.290	58.373	122	7	78	1	46	4
AdaBoostM1 (J48)	0.701	0.145	0.369	29.570	75.910	111	18	72	7	39	11
Logistic Regression	0.790	0.099	0.3053	21.013	62.664	116	13	71	8	45	5
AdaBoostM1 (LMT)	0.822	0.080	0.263	17.008	54.160	118	11	72	7	46	4
SGD	0.821	0.081	0.292	17.945	59.934	118	11	73	6	45	5
AdaBoostM1 (NaiveBayes)	0.762	0.127	0.339	26.906	69.602	114	15	68	11	46	4
AdaBoostM1 (ANN-MLP)	0.836	0.086	0.255	18.131	52.356	119	10	74	5	45	5
AdaBoostM1 (SMO)	0.771	0.108	0.329	22.839	67.614	115	14	72	7	43	7
AdaBoostM1 (kNN)	0.884	0.061	0.231	13.008	47.436	122	7	77	2	45	5
AdaBoostM1 (LWL)	0.821	0.096	0.271	20.323	55.733	118	11	73	6	45	5
AdaBoostM1 (BayesNet)	0.867	0.063	0.244	13.283	50.185	121	8	77	2	44	6
SVM (sigmoid) + PCA	0.821	0.085	0.292	17.945	59.933	118	11	73	6	45	5
SVM (linear) + PCA	0.836	0.077	0.278	16.313	57.147	119	10	74	5	45	5
(LDA) FLDA+ CfsSubsetEval	0.834	0.484	0.484	102.024	99.526	119	10	76	3	43	7
Decision Table	0.715	0.243	0.333	51.233	68.445	112	17	74	5	38	12
DNN (LSTM)	0.886	0.068	0.204	14.485	41.936	122	7	75	4	47	3
Voted Perceptron + PCA	0.852	0.071	0.264	15.077	54.297	120	9	75	4	45	3
Hybrid (proposed by the author)	0.918	0.074	0.198	15.641	40.741	124	5	77	2	47	3

Next, the investigation of classifying new data samples was executed (as displayed in Table 4.6). For the prediction evaluation, two metrics were chosen: prediction confidence (PrC, $0 \leq PrC \leq 1$), i.e., the probability for a classifier to output instance value, and error count (EC), which indicates how many false predictions (alarms) occurred on sick and healthy test sets. As Table 4.5 shows, not a single classifier performed without an error ($0 \leq EC \leq 2$) on the healthy test set, but on the sick test set, the classifiers AdaBoostM1 (RandomForest), AdaBoostM1 (MLP), AdaBoostM1 (SMO), AdaBoostM1 (kNN), SVM (linear) + PCA, FLDA, DNN (LSTM) and Voted Perceptron + PCA work without error (EC=0). This implies that probably a single healthy subject was feeling nervous (anxious), or the proposed models need extra tuning.

Table 4.6. Evaluating predictions on unseen data. Two individual test sets are considered: 10 samples are taken from healthy test subjects and 10 samples from sick test subjects. *PrC_0* on each of 10 samples (Healthy Test Set, target class=0). *PrC_1* on each of 10 samples (Sick Test Set, target class=1). *EC_0* (Healthy Test Set, target class = 0), *EC_1* (Sick Test Set, target class=1)

Classifier	PrC_0	PrC_1	EC_0	EC_1
AdaBoostM1 (Decision Stump)	1 1 1 1 1 0.999 1 1 1 1	0.974 1 0.945 0.934 0.999 1 0.533 1 0.959 0.999	1	1
AdaBoostM1 (RandomForest)	0.810 0.80 0.96 0.91 0.94 0.78 0.95 0.98 0.94 0.95	0.820 0.770 0.620 0.650 0.780 0.780 0.740 0.600 0.770 0.830	1	0
AdaBoostM1 (J48)	1 (for all samples)	1 (for all samples)	1	0
Logistic Regression	1 (for all samples, except one = 0.997)	1 (for all samples)	1	1
AdaBoostM1 (LMT)	1 (for all samples)	1 (for all samples)	1	0
SGD	(for all samples)	(for all samples)	1	0

AdaBoostM1 (NaiveBayes) -------	1 1 1 1 1 1 1 1 0.934 0.958	1 1 0.998 1 1 1 1 0.758 1 1	2	0
AdaBoostM1 (ANN-MLP)	1 0.998 1 1 1 0.991 1 1 0.999 1	1 1 1 0.999 1 1 0.999 0.984 1 0.990	1	0
AdaBoostM1 (SMO)	1 (for all samples)	1 (for all samples, except one = 0.964)	1	0
AdaBoostM1 (kNN)	0.992 (for all samples)	0.992 (for all samples)	1	0
AdaBoostM1 (LWL)	1 (for all samples)	1 1 1 1 1 1 0.722 1 1 1	1	1
AdaBoostM1 (BayesNet)	1 (for all samples)	0.981 0.988 0.981 1 1 1 1 1 1 1	1	1
SVM (sigmoid) + PCA	1 (for all samples)	1 (for all samples)	1	2
SVM (linear) + PCA	1 (for all samples)	1 (for all samples)	1	0

(LDA) FLDA + CfsSubsetEval	0.524 0.521 0.518 0.533 0.530 0.517 0.537 0.523 0.514 0.518	0.531 0.541 0.531 0.516 0.528 0.517 0.523 0.515 0.521 0.509	1	0
DNN (LSTM)	0.999 0.995 1 1 1 0.986 1 1 1 1	1 1 1 1 1 1 1 0.958 1 1	1	0
Decision Table	0.909 0.833 0.800 0.500 0.800 0.611 0.667 0.8 0.8 0.909	0.611 0.611 0.611 0.750 0.923 0.750 0.5 0.923 0.923 0.923	1	4
Voted Perceptron + PCA	1 (for all samples)	1 (for all samples)	2	0
Hybrid (proposed by the author)	1 1 1 1 1 0.910 1 1 1 1	0.994 0.671 0.994 0.667 1 1 1 0.988 1 1	1	0

As for prediction on new test sample, the results are sustained (in comparison to the classifier from Table 4.5, which generates the smallest EC for healthy and sick test sets), i.e., EC=1 (healthy set), EC=0 (sick test set).

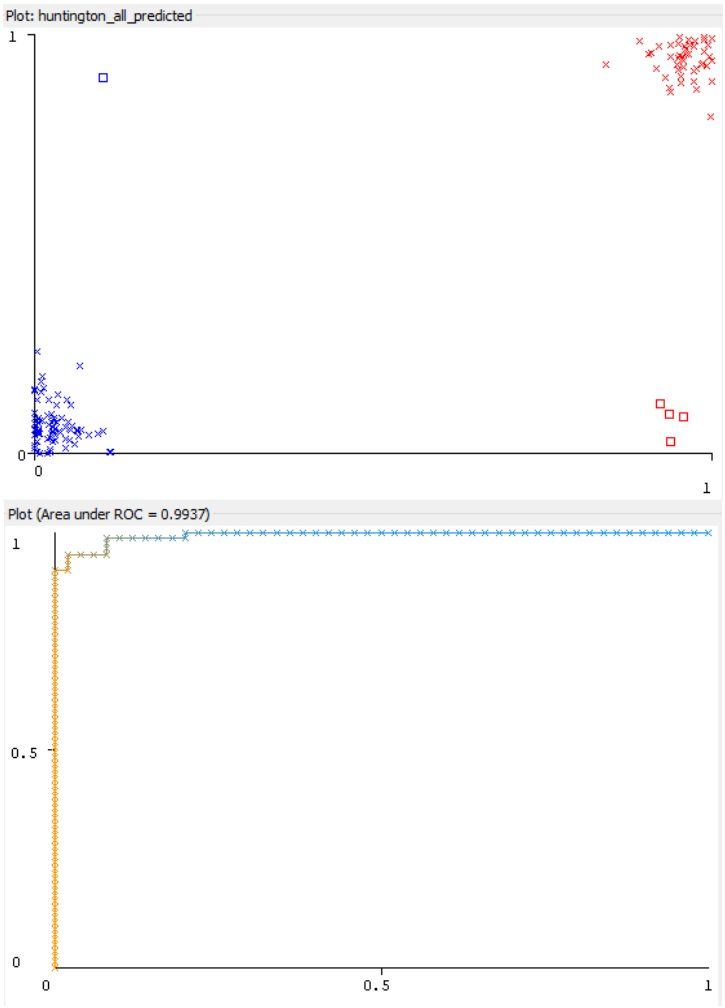


Figure 4.3. Statistical evaluation of hybrid model. Error visualization (top side, 5 incorrect classifications), ROC curve (bottom side) increases to 0.994

4.3.3. E3: Sick vs. Healthy classification by using recurrent neural network (RNN) with long short-term memory (LSTM)

The purpose of experiment E3 is to classify voice recordings, taken from T14 task, into sick and healthy instances. The procedure starts by creating a root folder and 2 subfolders, naming '*train'* and '*test'* correspondingly. E3 takes the initial audio materials as 64 kbps sample rate files (*.mp3* format).

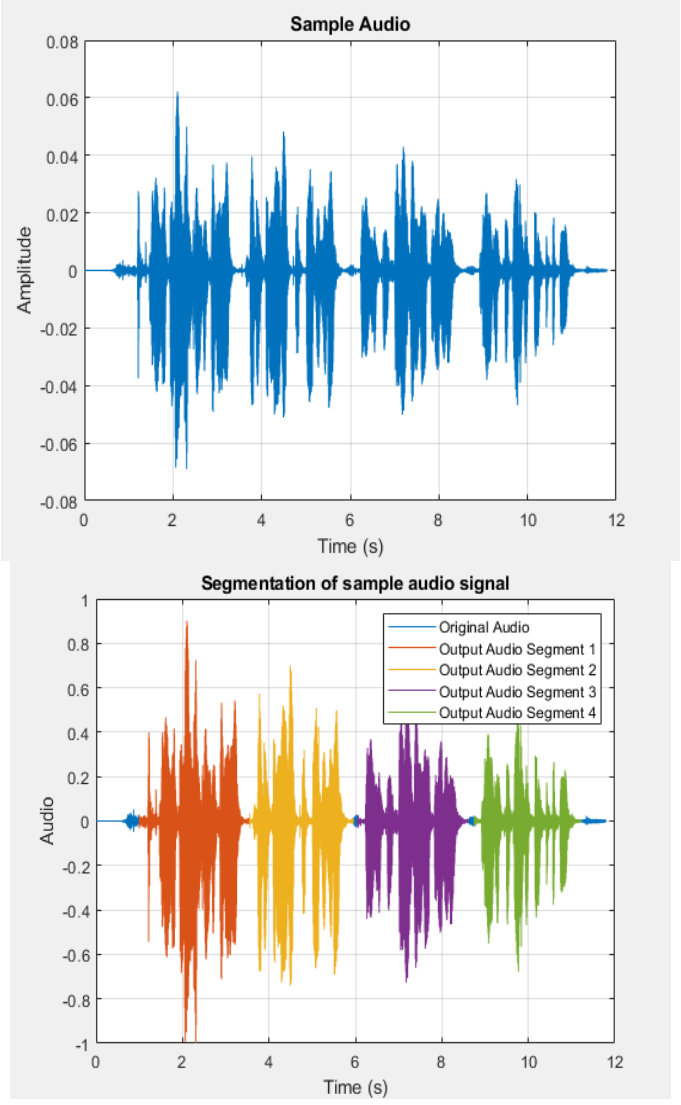


Figure 4.4. The statistical plot of the original audio signal (top side) with silence segments (taken a random sample from a test subject) by using amplitude and time scales. The bottom side illustrates the detected speech segments of the signal (number of segments can change, depending on the speaker)

Two *.csv* files (one for training set, another for testing set) are prepared for storing the summarized information about the collected files by using this format: linkage to the stored audio file in disk (e.g., *'hd_test/audio.00128.mp3'*), transcript of a read poem, sick or healthy indicator and recording duration (varies from 9 ms, spoken by a healthy subject, to 47 seconds, spoken by a sick subject). The data is transferred to a single object in memory *audioDataStore* (MATLAB environment), which contains all required info for further processing of audio files: filename (with extension), labels, counts for each label (number of healthy and sick instances). The silence segments can appear in the original audio; thus, the approaches are needed to remove such segments (see Figure 4.4).

In order to eliminate the silence segments that do not contain useful information that is pertaining to the health status indicator of the speaker, the isolation of the speech segment method is applied (see Figure 4.5). This method uses the thresholding approach. First, 2 features (*signalEnergy*, *centroid*) over non-overlapping frames of the audio data are calculated. Next, the energy and spectral centroid for each frame is evaluated; centroid threshold ($T\_C = 5000\ Hz$) and energy threshold (T_E) are calculated afterwards. The speech regions where the feature values fall below or above their respective thresholds are disregarded. On the contrast, the speech region is active in cases as shown in (81) equation:

$$\text{isSpeechRegion} = signalEnergy \geq T\_E\,, centroid \leq T\_C \tag{81}$$

In the implementation, *IsSpeechRegion* is further characterized by *regionStartPos* (indices of frames where a speech-to-silence or silence-to-speech transition occurs), *RegionLengths* (length of all-silence or all-speech regions), start and end indices (*SI*, *EI*) for each speech region. Once the active speech regions are detected, the intersecting speech segments are merged and fed for the feature extraction mechanism (*segments*).

The speech signal changes over time, but is stationary on short time scales; thus, their processing is often done in windows of 20–40 ms. For each speech segment, a periodic hamming window [157] with 80% overlap is used and then concatenated into sequences (each vector contains 92 features, each sequence 40 feature vectors). The features used in E3 are the following: GTCC, MFCC, pitch, slope, skewness, spread, flux, rolloff, decrease, flatness, kurtosis and entropy (refer to 3.2.13 section for details). These 12 features are concatenated together and can be combined in numerous ways. A feature can be removed from the sequence or swapped with other if necessary.

With this configuration setup, the next step is feature transferring to tall array *T* (this provides a way to work with data backed by an audio data store (*audioDataStore*) that can have millions or billions of rows) on the GPU. These feature sequences are re-evaluated (*featureSequences*) and normalized (mean and standard deviation for each coefficient is computed). Such normalized GPU features are ready to be supplied for training deep neural network (DNN).

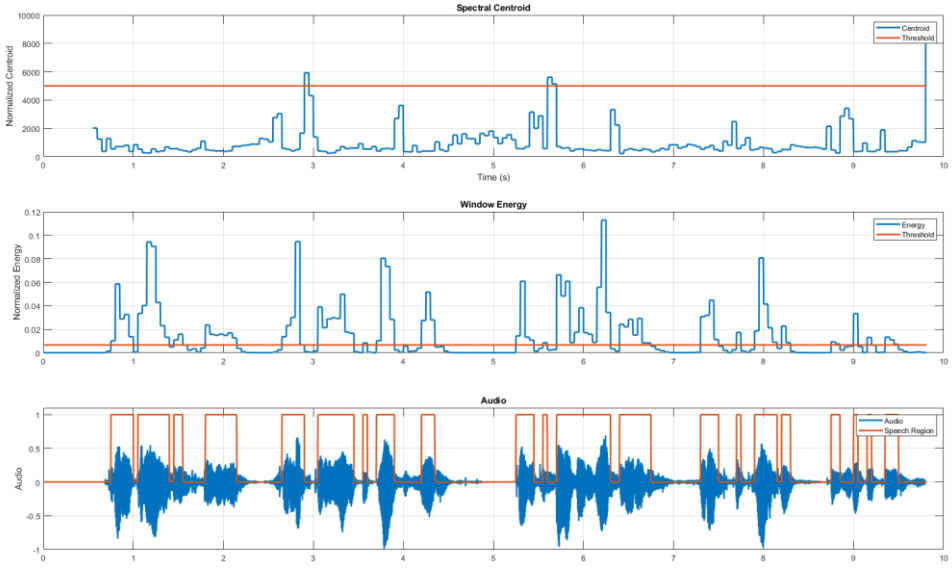


Figure 4.5. Spectral centroid applied to sample audio signal with threshold T_C (top), signal energy with threshold T_E (middle part) and active voice detection speech segment in orange colour (bottom)

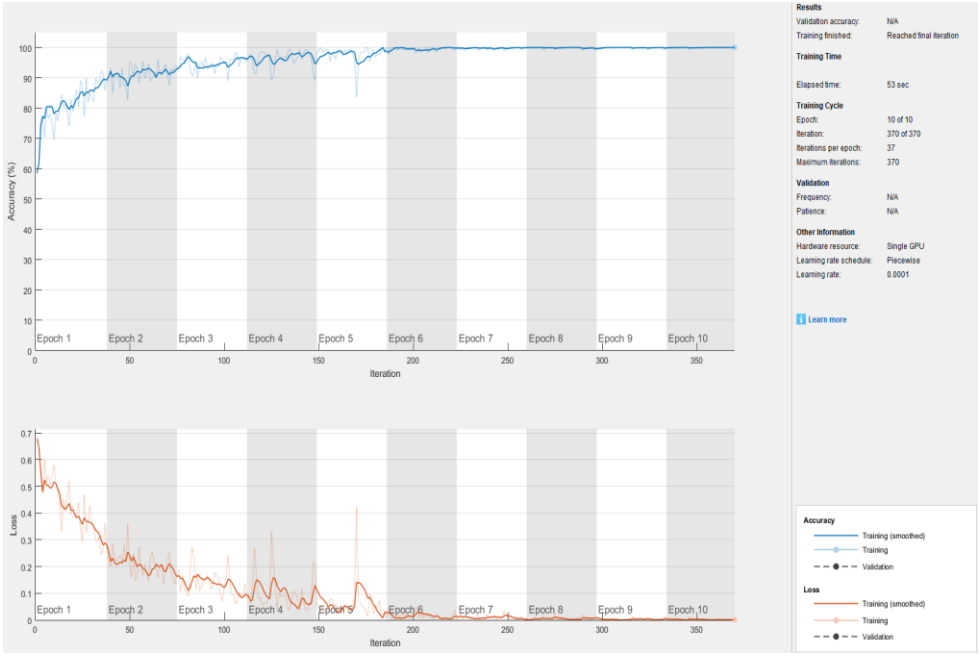


Figure 4.6. Training of BiLSTM for sick vs. healthy classification problem: network converges (achieves training accuracy (blue curve) is close to 100%, loss (orange curve) is close to 0), 370 training iterations, 10 epochs, took 53 seconds

Bidirectional Long Short-Term Memory (BiLSTM) neural network [158], [159] is used in E3. Parallel Pool is invoked (Number of Workers used is 4) for training BSTM to speed up the training process on a single GPU (Figure 4.6).

In E3, the training dataset consist of 234 and test set of 35 (recall that each test subject tells the poem multiple times) voice recordings. LSTM can learn long-term dependencies between time steps of sequence data (forward and backward directions). BiLSTM training options: algorithm (optimizer) is *RMSProp* (root mean square propagation) [160], *MaxExpochs*=10, *MiniBatchSize*=128, Shuffle on every epoch, learning rate drop factor=0.1, learning rate schedule *'piecewise'*. BiLSTM architecture: 2 fully connected layers of 100 neurons, followed by a softmax [161] layer and a classification (output) layer. BiLSTM design can be modified if necessary.

Figure 4.7 illustrates sick vs. healthy classification results on the provided dataset and applying Major Vote method (rule) for tuning classifier performance. All instances from the training set were correctly classified, whereas in a test set, 2 healthy instances were incorrectly classified, thus giving the accuracy of 86.63%.

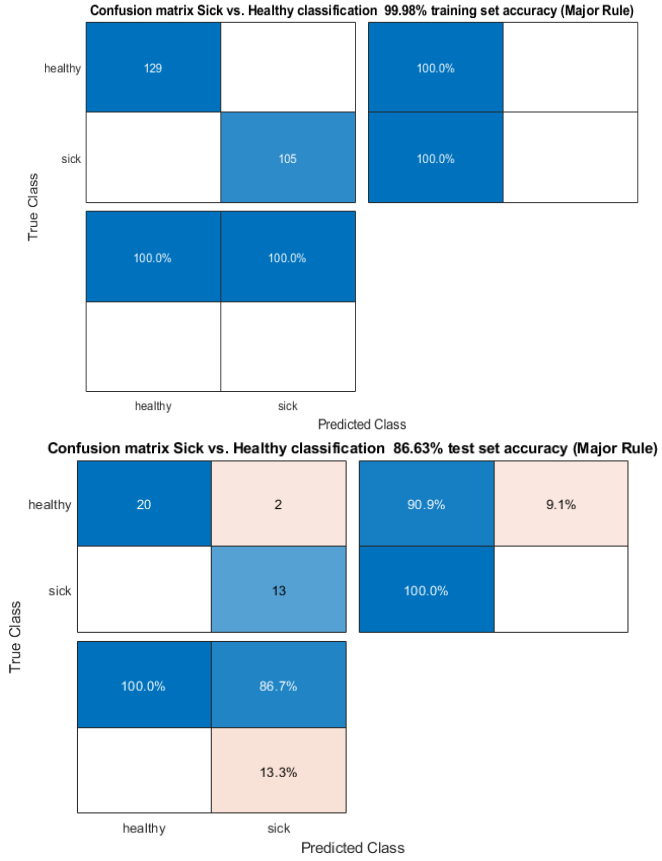


Figure 4.7. Achieved results of E3 classification by using BiLSTM network and Majority Vote method: Taining accuracy is 100%, the test set 86.63%

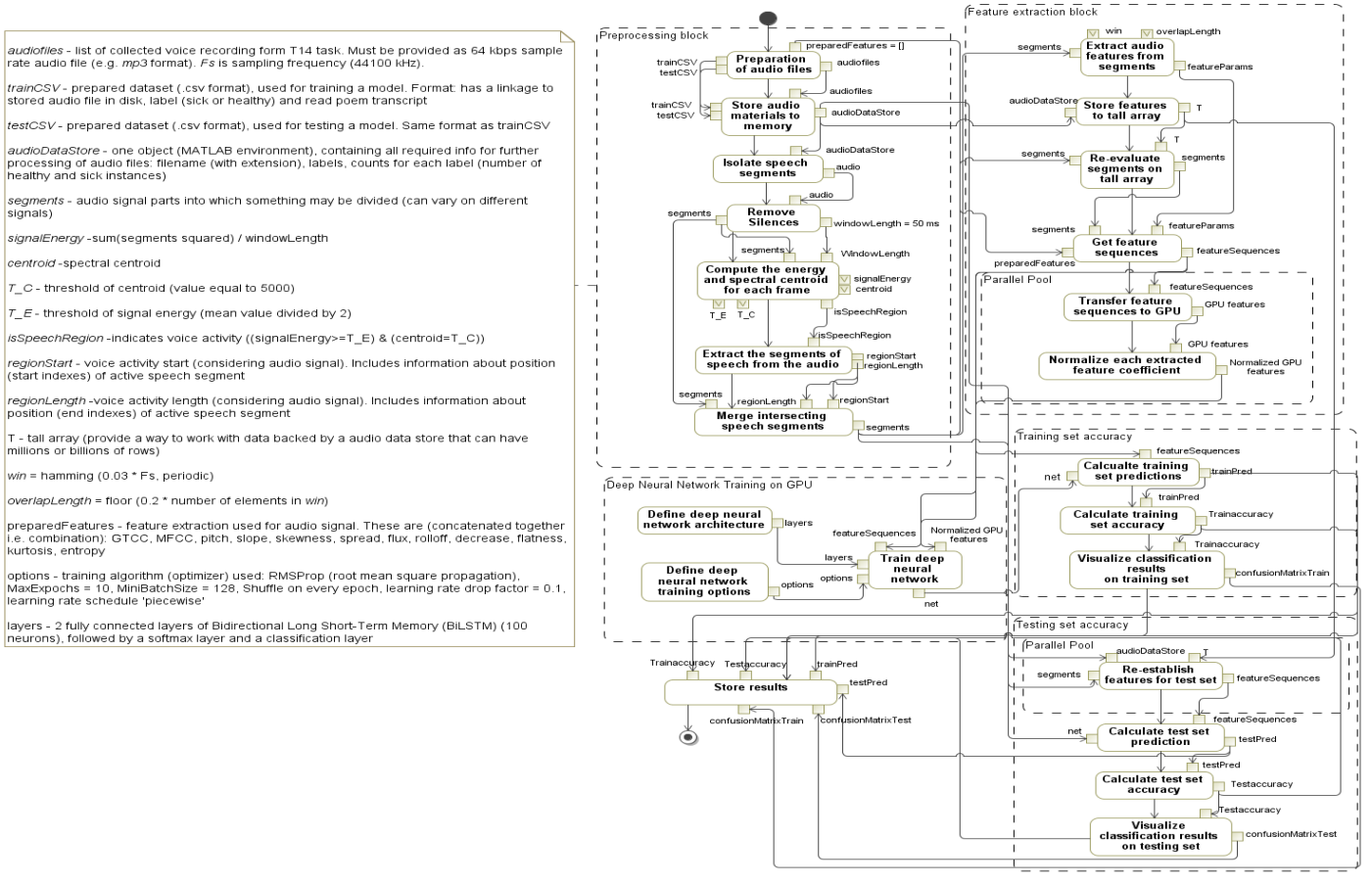


Figure 4.8. Implementation of building a speech impairment model for healthy vs. sick classification problem by using Bidirectional Long Short-Term Memory (BiLSTM) network and MFCC, GTCC, pitch and several spectral shape descriptor methods

4.3.4. E4: Sick vs. Healthy classification by using wavelet scattering transform method (WST)

The purpose of E4 is the same as in E3, i.e., to solve binary classification problem (sick vs. healthy) based on voice recordings, collected from T14 task. WST method applies Gabor (analytic Morlet) wavelet [162]. Such wavelets use low pass scaling function to generate low-variance representations of real-valued time series data. The initial preparation steps of E4 include the creation of a root folder and 2 sub folders naming '*healthy'* and '*sick'* correspondingly. The names of the subfolders must match the names of the output target classes. In E4, the author's collected dataset is used: train set (70%) consists of 190 records (107 healthy and 83 sick), test set (30%) of 81 (46 healthy and 35 sick). The audio files are provided as 1411 kbps sample rate *.wav* audio files at 22050 Hz.

The next phase is the design of wavelet (*sf*). The signal length is a natural logarithm value of 2^{19}. For WST configuration, only 3 parameters are provided: the duration of the time invariance, the number of wavelet filter banks (band-pass filters that separate the input signal into multiple components, each one carrying a single frequency sub-band of the original signal) and the number of wavelets per octave. Two wavelet filter banks are used: first (*fb1*) and second (*fb2*) (see Figure 4.9). The first filter bank has 8 wavelets per octave, and the second filter bank has 1 wavelet per octave. The time invariance scale (*InvarianceScale*) is set to 0.5 seconds. As Figure 4.10 (top side) image shows, *InvarianceScale* parameter that is plotted on the coarsest-scale does not exceed the invariant scale of the wavelet scattering decomposition, i.e., is indicator of low variance.

Using a similar approach as in E3, in E4, the audio materials are transferred to a single object in memory *ads*. Train and test data are converted to tall array *Ttrain* and *Ttest*. Then, scattering train features (*scatteringTrain)* and scattering test features (*scatteringTest*) are created by applying log transformation of each audio file and subsamples, the number of scattering windows by 8. Moreover, the scattering features are combined together to a matrix by using MATLAB Parallel Pool (Number of Workers=4) on a single GPU, resulting in feature sets *TrainFeatures* and *TestFeatures* (each row of the matrix is 1 time window across the *N=341* paths in the scattering transform of each audio signal).

TrainFeatures and *TestFeatures* are used to fit the data for support vector machine (SVM) model with polynomial kernel (order=3). SVM tuning was applied by using the Majority Vote method (as in E3), which proved to achieve 100% accuracy of the supplied test data (see bottom side image of Figure 4.10).

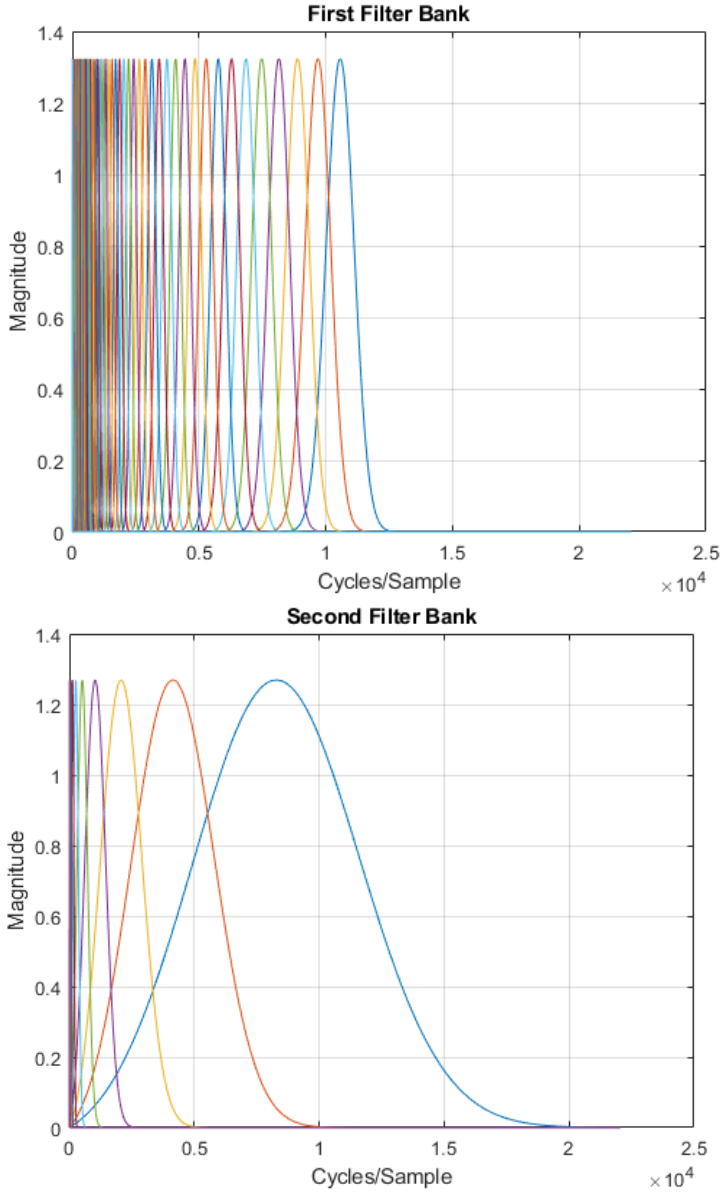


Figure 4.9. Filter banks for the designed wavelet: first filter (8 per octave), second filter (1 per octave)

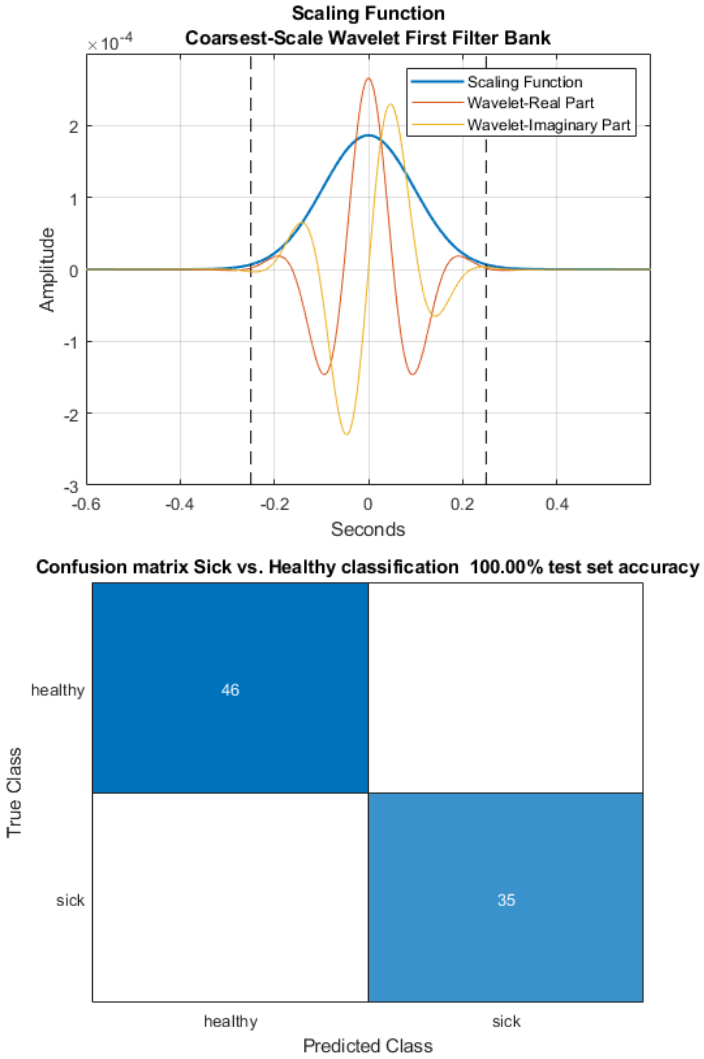


Figure 4.10. Invariance is set to 0.5 seconds (top side). The achieved classification results (bottom side) for E4 on test set using WST method is 100% accuracy

4.4. Experiments summary

All four conducted experiments were designed to solve sick vs. healthy binary classification (17 supervised learning classifiers were considered) problem, which targets patients with central nervous system disorders (CNSD). The purpose of E1 is to solve the proposed problem, based only on a single task execution (feature distribution method), whereas in E2, all extracted features from tremor, cognitive, energy expenditure tasks (T0, T1, T2, T3, T4, T5, T6, T7, T8, T9, T10, T11, T12, T13, T15) are considered. In addition, the purpose of experiments E3 and E4 is the classification of voice recordings, taken from T14 task, into sick and healthy groups.

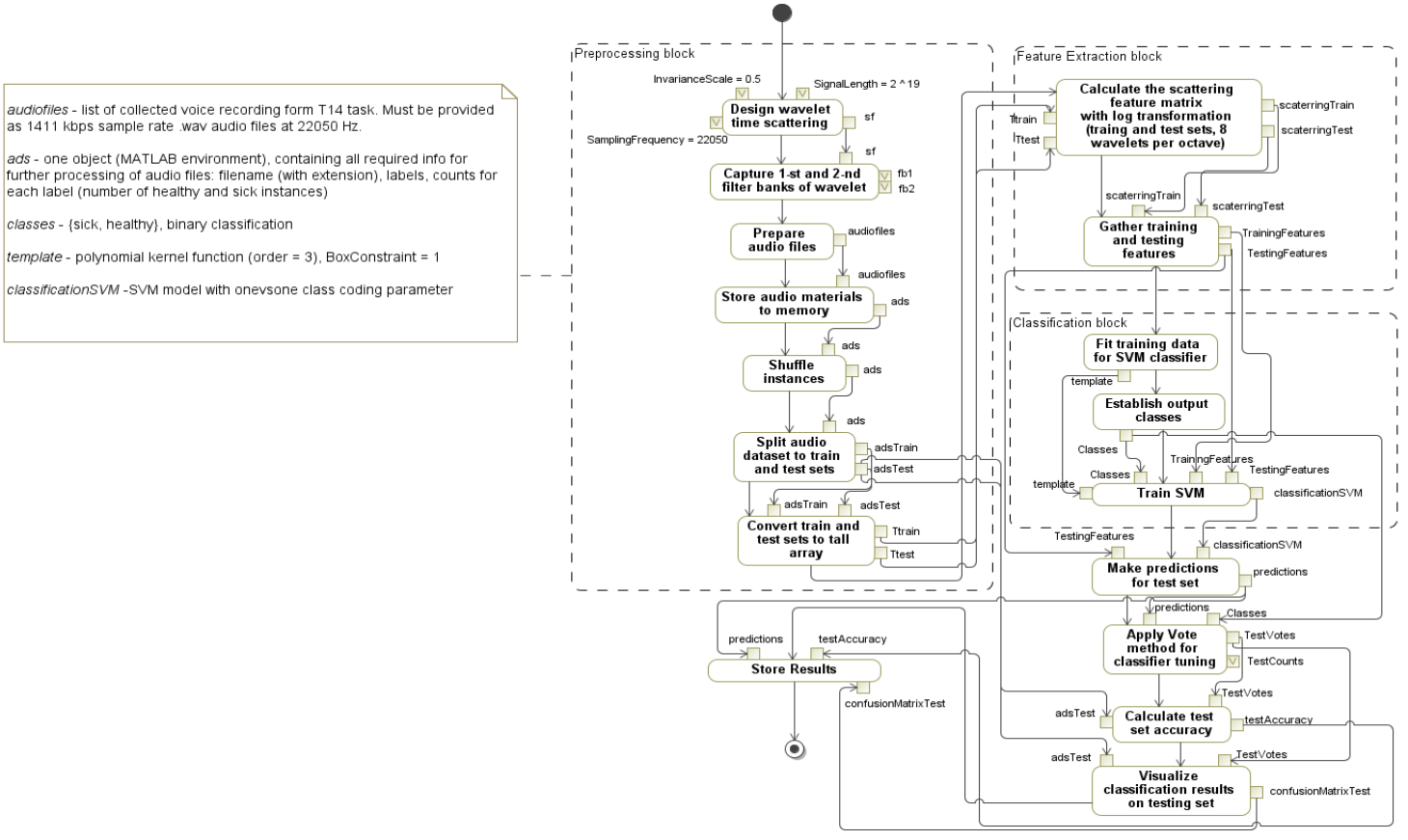


Figure 4.11. Implementation of building a speech impairment model for healthy vs. sick classification problem by using WST and SVM (polynomial kernel, 3-rd order)

Table 4.7 provides a summary of requirements in order to replicate E1, E2, E3 and E4 experiments. Refer to Appendix C for a complete list of features used for E1 and E2. Appendix D provides details on how to design Weka *.arff* files. The author notes that the requirements for extra experiments to collect data from CNSD patients are Android SDK v. 6.0 or up, supported by tablets with sizes (10.1" screen, resolution 1280 x 800 pixels) and smartphones (5.1", 2560 x 1440), (5.5", 1920 x 1080).

Table 4.7. Requirements for proposed experiment setup

	Data Format	**Software**	**Hardware**
E1	*.arff* files (each model uses a separate file with distributed features)	Classification tool: *Weka v. 3.8* Compiler: *Java ≥ 1.7* Operating System: *Windows, Mac, Linux*.	*CPU* (no further requisites are necessary)
E2	*.arff* file (one file with 238 features)		
E3	64 kbps sample rate *.mp3* format audio files	Classification tool: *MATLAB Audio Toolbox R2019a (*requires *DSP System Toolbox, Signal Processing Toolbox)* Compiler: *Microsoft Visual C++ 2015 or up.* Operating System: *Windows, Mac, Linux*.	*CUDA*-enabled *NVIDIA GPU*s with compute capability 3.0 or higher
E4	1411 kbps sample rate *.wav* audio files		

As for the experimental results and discussion, in E1, the splitting features for individual tasks cause misclassification (highest achieved accuracy is 91.50%). Such models are likely to cause errors in predicting new data samples. E2 gives more promising results, i.e., especially when adapting classifier fusion mechanism to implement a hybrid model (accuracy increases to 96.12%). This factor suggests an execution of the full testing procedure (all tasks) for a better determination whether a test subject is sick or healthy.

In addition, E3 and E4 experiments for speech impairment detection provide strong expectations of proper sick vs. healthy classification models: *BiLSTM* method achieves 86.63% and *WST* 100% accuracy.

Extra validation approaches for the proposed models could be acquired, i.e., extending the dataset with additional patient monitoring data, targeting not yet investigated diseases (e.g., Alzheimer's) in various progression statuses.

5. CONCLUSIONS AND FUTURE WORK

5.1. Conclusions

1. The related work analysis provides a comprehensive study of existing data mining and machine learning methods for health state assessment in patients with central nervous system disorders (CNSD). The minority of these solutions targets Huntington Disease (HD) patients in described investigations. In addition, CNSD patient monitoring approaches are very different and have limitations, e.g., one does not use a mobile device, the second adapts extra, the third uses party wearable devices that usually increase the stress factor for patients and are uncomfortable. None of the related solutions proposes an extended methodology that applies a combination of feature extraction methods to target tremor, cognitive, speech and energy expenditure impairments that are occurring in CNSD patients as it is presented in this dissertation.
2. The restrictions of existing solutions have been addressed by gathering a priori knowledge of the medical staff and developing a computerized version of SAGE (Self-administered cognitive testing) methodology. The proposed system is an actual framework that requires only one smart non–invasive interface, i.e., mobile device or tablet for CNSD patient's neural impairment screening. The framework supports 16 tasks: 10 of them are based on SAGE methodology, the others adapt finger tapping, spiral contour following (drawing), voice recording and energy expenditure tasks as an extension. In total, more than 230 features are automatically extracted in the framework, i.e., finger motion tracking, task duration, distance estimation of geometrical shapes, graph similarity evaluation, image collection from clock drawing (CDT) task, SAGE scores, voice recordings, daily calorie balances etc. An additional innovative aspect of this dissertation is the creation of newly designed dataset, which opens a gateway for using it in a machine learning repositories (e.g., UCI) by other computer scientists.
3. The practical significance of this dissertation has been evaluated by the implementation of Neural Impairment Test Suite (NITS) mobile application, which provides feedback on the patient's health status in a graphical form and compares the results with self and healthy subject's estimates. NITS is available on Google Play web portal. More than 500 users have already installed NITS app at the time of writing this dissertation. The author notes, however, that the proposed mobile app is designed as a curiosity tool to self-evaluate behaviour changes of people in a risk group for the suspected symptoms of neural impairments (e.g., those of Huntington, Parkinson diseases, MCI, dementia) and does not replace professional medical assessment. Moreover, only reliable data, which is approved by the data collection permit of Kaunas University of Technology (KTU) and signed with a test subject in the agreement document, is used in this work. Other, unknown sources are ignored.

4. Four experiments were carried out for solving sick vs. healthy binary classification problem. The collected dataset was used to validate the proposed experiments. Tremor, cognitive, energy expenditure features are considered in experiments E1 and E2. E1 applies innovative feature distribution method, i.e., for each proposed individual methodological task, the features are grouped separately. Furthermore, three extra unique feature combinations were composed for designed classification models: proper spelling, SAGE points and task durations in seconds. E1 showed the best results for task T9 with LWL classifier = 91.50% (lowest achieved accuracy was 74.52). In experiment E2, the combination of all 238 features into a single classifier resulted in 3% increased accuracy model, compared to the best performance in E1 with AdaBoostM1 (Random Forest) = 94.57% classifier. In addition, all 12 proposed classifiers for E2 experiment achieved at least 91.47% accuracy. E3 and E4 experiments deal with speech impairments. E3 applies bidirectional recurrent neural network with long short-term memory (BiLSTM) and achieved an accuracy of 86.63% for the test set (100% for training model and network converge). E4 uses wavelet scattering transform (WST) method to achieve the accuracy of 100% for the test set. WST overcomes BiLSTM, considering the fact that collected voice recording from CNSD patients were significantly long, i.e., up to 47 seconds.
5. In a classifier ensemble (hybrid model) design, multiple combinations with ~2% increased accuracy, i.e., 96.12%, were found by using heuristic approach. The fusing of 3 classifiers: AdaBoostM1 (SMO), AdaBoostM1 (BayesNet) and Voted Perceptron + PCA) with Vote method (Average of Probabilities combination rule), results in best-fit combination based on the highest accuracy value and minimum time expenditure (0.59 seconds) for model training. The hybridization approach targets tremor, cognitive and energy expenditure feature set, leaving out the audio features from standalone experiments E3 and E4, as they require special software environment for voice signal processing and extra computational resources.

5.2. Future Work

The built models cover a very wide range of possible disorders, occurring to those in Huntington, Parkinson or MCI disease patients: some can have a tremor in hands or body, others may have memory loss, voice problems or weight loss. This implies a high potential for the determination of patient's health state deterioration status. The designed theoretical models (based on experiments E1, E2, E3 and E4) can be integrated into the proposed NITS mobile app for the prediction of alerts (alarms) to a patient about his disease progression before symptoms occur visually.

Acknowledgements

The author would like to thank the president of Lithuania Huntington disease association Dr. Zivile Navikiene for contacting family members of HD patients and disseminating proposed research in the official website and Facebook channel of the association to help carry out the experiments described in this dissertation as well as for her practical support and advices.

Human Studies

The research was approved by the Institutional Review Board of the Faculty of Informatics, Kaunas University of Technology.

6. REFERENCES

Table 6.1. Systematic overview of literature references

Reference No.	Group
[10], [12], [13], [14], [15], [16], [17], [50], [78], [79], [97], [98], [99], [100], [101], [102], [103], [104], [105], [154], [176], [177], [178], [201], [202], [203], [204], [205], [206], [209], [210], [225], [234], [236], [237], [238], [240], [241], [242, [243], [244], [257], [260]	Bioinformatics research and medical methodologies
[3], [5], [22], [25], [26], [27], [28], [34], [35], [37], [38], [43], [44], [47], [54], [63], [73], [84], [85], [86], [93], [106], [107], [108], [109] [110], [111], [112], [113], [114], [115], [116], [117], [118], [119], [120], [121], [124], [125], [126], [127], [128] [129], [130], [131], [132], [133], [134], [135], [136], [137], [138], [139], [140], [141], [142], [143], [144], [145], [146], [147], [148], [149], [158], [159], [162], [163], [166], [167], [168], [172], [173], [192], [199], [212], [217], [219], [220], [224], [249], [250], [251], [258], [259]	Computer science algorithms (methods), models
[150], [151], [152], [155], [156], [157], [169]	Statistical analysis of methods
[1], [2], [6], [7],[11], [23], [24], [29], [30], [31], [33], [36], [39], [40], [41], [45], [46], [48], [49], [51], [52], [53], [55], [56], [57], [58], [59], [60], [61], [62], [64], [65], [66], [67], [69], [70], [71], [72], [74], [75], [80], [81], [82], [83], [92], [94], [95], [96], [122], [123], [174], [179], [180], [181], [182], [183], [184], [185], [186], [187], [188], [189], [190], [191], [193], [194], [195], [197], [198], [200], [207], [208], [213], [214], [215], [216], [218], [221], [222], [223], [226], [227], [228], [229], [230], [231], [232], [233], [239], [248], [252], [253], [254], [255], [256]	Applications of computer science related research
[18], [147], [148], [153], [196]	Tools
[76], [87], [88], [89], [90], [91]	OER (Open Educational Resources)
[8], [9], [32], [42], [68], [77], [160], [161], [164], [165], [170], [171], [211], [235], [245], [246], [247],	Computer science related reviews or surveys
[19], [20], [21]	Dissemination
[4], [175]	Theoretical studies

1. F. Patou, M. Dimaki, W. E. Svendsen, K. Kjaegaard, and J. Madsen, “A smart mobile lab-on-chip-based medical diagnostics system architecture designed for evolvability,” in 2015 Euromicro Conference on Digital, System Design. IEEE, aug 2015.
2. C. Doukas, T. Pliakas, and I. Maglogiannis, “Mobile healthcare information management utilizing cloud computing and android os,” in Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE. IEEE, 2010, pp. 1037–1040.
3. W.-T. Sung and Y.-C. Chiang, “Improved particle swarm optimization algorithm for android medical care iot using modified parameters,” Journal of medical systems, vol. 36, no. 6, pp. 3755–3763, 2012.
4. A. Kailas, C.-C. Chong, and F. Watanabe, “From mobile phones to personal wellness dashboards,” IEEE pulse, vol. 1, no. 1, pp. 57–63, 2010.
5. K. Huckvale, S. Adomaviciute, J. T. Prieto, M. K.-S. Leow, and J. Car, “Smartphone apps for calculating insulin dose: a systematic assessment,” BMC medicine, vol. 13, no. 1, p. 106, 2015.
6. Y. E. Gelogo and H.-K. Kim, “Integration of wearable monitoring device and android smartphone apps for u-healthcare monitoring system,” International Journal of Software Engineering & Its Applications, vol. 9, no. 4, 2015.
7. A. Nugroho, N. Nuryani, I. Yahya, A. D. Sutomo, B. Haijito, and A. Lestari, “Premature ventricular contraction detection using artificial neural network developed in android application,” in Electric Vehicular Technology and Industrial, Mechanical, Electrical and Chemical Engineering (ICEVT & IMECE), 2015 Joint International Conference. IEEE, 2015, pp. 212–214.
8. A. Jutel and D. Lupton, “Digitizing diagnosis: a review of mobile applications in the diagnostic process,” Diagnosis, vol. 2, no. 2, pp. 89–96, 2015.
9. E. Rovini, C. Maremmani, and F. Cavallo, “Automated systems based on wearable sensors for the management of parkinson’s disease at home: A systematic review,” Telemedicine and e-Health, jul 2018. [Online]. Available: https://doi.org/10.1089/tmj.2018.0035
10. G. K. Wenning, S. Kiechl, K. Seppi, J. Müller, B. Högl, M. Saletu, G. Rungger, A. Gasperi, J. Willeit, and W. Poewe, “Prevalence of movement disorders in men and women aged 50–89 years (bruneck study cohort): a population-based study,” The Lancet Neurology, vol. 4, no. 12, pp. 815–820, dec 2005. [Online]. Available: https://doi.org/10.1016/s1474-4422(05)70226-x
11. Bennasar, M. et al. (2016). Huntington's Disease Assessment Using Tri Axis Accelerometers. In Procedia Computer Science, vol. 96, pages 1193-1201. https://doi.org/10.1016/j.procs.2016.08.163.
12. Wiecki, T. M. et al. (2016). A Computational Cognitive Biomarker for Early-Stage Huntington’s Disease. PLOS journal. [Online]. https://doi.org/10.1371/journal.pone.0148409.

13. Warby C. S., Visscher H., Collins A. J. et al. (2011). HTT haplotypes contribute to differences in Huntington disease prevalence between Europe and East Asia. *Article European Journal of Human Genetics*. EJHG, 2011, January 19, volume 19, pages 561-566.
14. Coordination center of Huntington in Lithuania at Vilnius Hospital of Santara clinic. Internet access:

    http://www.santa.lt/index.php?option=com_content&view=article&id=2041&catid=178&Itemid=129, 2015.
15. Cambray S., Cattaneo E. (2013). Huntington's disease: how could stem cells help? *EuroStemCell project from European Union's Horizon 2020 research and innovation program*. Stem Cells fact sheets. Online: http://www.eurostemcell.org/stem-cell-factsheets.
16. Hsu A. (2004). The Inheritance of Huntington's disease. *Hopes*. Huntington's outreach project for education, at Stanford University. Online: http://web.stanford.edu/group/hopes/cgi-bin/hopes_test/the-inheritance-of-huntingtons-disease-text-and-audio.
17. Nguyen, P.H.H, & Cenci, A.M. (Eds.). (2015). Behavioral Neurobiology of Huntington's disease and Parkinson's disease. Springer-Verlag Berlin Heidelberg. *Current Topics in Behavioral Neurosciences*, 2013, volume 22, pages 3–29, ISBN: 978-3-662-46343-7, ISSN: 1866-3370.
18. Lauraitis, A. (2018). Neural Impairment Test Suite. Google Play. [Online]. https://play.google.com/store/apps/details?id=com.alauraitis.test_suite&hl=en
19. Amanda Hollenbeck (2019). Smartphone App Can Help Detect Early Symptoms of Rare Disease. In Laboratory Equipment. [Online]:

    https://www.laboratoryequipment.com/article/2019/06/smartphone-app-can-help-detect-early-symptoms-rare-disease.
20. Iqra Mumal (2019). New Smartphone App Able to Recognize Early Symptoms of Huntington's Disease. In Huntington's Disease News. [Online]:

    https://huntingtonsdiseasenews.com/2019/05/28/new-smartphone-app-recognizes-early-huntingtons-symptoms/
21. LRT (2019). Rytdienos pranešimas, padėsiantis gyventi ilgiau: atpažinti retą ligą gali padėti ir mobilioji programėlė. In Lithuanian Television and Radio portal [Online]:

    https://www.lrt.lt/naujienos/mokslas-ir-it/11/1060638/rytdienos-pranesimas-padesiantis-gyventi-ilgiau-atpazinti-reta-liga-gali-padeti-ir-mobilioji-programele
22. Engin, M., Demirag, S., Engin, Z., E., Celebi, G., Ersan, F., Asena, F., Colakoglu, Z. (2007). The classification of human tremor signals using artificial neural network (2007). Expert Systems with Applications, 33(3), pp. 754-761.
23. Wu D., Warwick K., Ma Z., Gasson M.N., Burgess J.G., Pan S., Aziz T.Z. (2010). Prediction of Parkinson's disease tremor onset using a radial basis function neural network based on particle swarm optimization. Int. J. Neur. Syst., 20(02), 109.

24. Cole B.T., Roy S.H., De Luca C.J., Nawab S. H. (2010). Dynamic neural network detection of tremor and dyskinesia from wearable sensor data. In: Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE, 2010:6062-5.
25. Chandrashekhar A., Jain S., Kumar Jha V. (2014). Design and Analysis of Data Mining Based Prediction Model for Parkinson's disease. International Journal of Computer Science Engineering (IJCSE), Vol. 3 No.03.
26. Furuhashi T., Tano S., Jacobsen H-A. (2012). Deep Fusion of Computational and Symbolic Processing. Publish in Physica, Fuzziness and Soft Computing, vol. 59, 256 p.
27. Kazienko P., Lughofer E., Trawinski, B. (2013). Hybrid and Ensemble Methods in Machine Learning. Published in Journal of Universal Computer Science, 19(4), 457- 461.
28. Fuller R. (2013). Introduction to Neuro-Fuzzy Systems. Published in Springer Science & Business Media (2013), volume 2 of Advances in Intelligent and Soft Computing, 289 p.
29. Geman O. (2013). Parkinson's disease Assessment using Fuzzy Expert System and Nonlinear Dynamic. In Advances in Electrical and Computer Engineering, 13(1), pp. 41-46.
30. Obi J.C., Imainvan A.A. (2011). Decision Support System for the Intelligent Identification of Alzheimer using Neuro Fuzzy logic. International Journal on Soft Computing (IJSC), Vol.2, No.2, DOI: 10.5121/ijsc.2011.2203.
31. Iram S., Fergus P., Al–Jumeily D., Hussain A., Randles M. (2015). A classifier fusion strategy to improve the early detection of neurodegenerative diseases. International Journal of Artificial Intelligence and Soft Computing (IJAISC), 5(1), 23-44.
32. Viharos Z.J., Kis K.B. (2015). Survey on Neuro-Fuzzy systems and their applications in technical diagnostics and measurement. Measurement, 67, 126-136. https://doi.org/10.1016/j.measurement.2015.02.001
33. Chiuchisan I., Geman, O. (2014). An Approach of a Decision Support and Home Monitoring System for Patients with Neurological Disorders using Internet of Things Concepts. WSEAS Transactions on Systems, 13, 460-469.
34. Yang G., Lin Y., Bhattacharya P. (2008). Multimodality inferring of human cognitive states based on integration of neuro-fuzzy network and information fusion techniques. EURASIP Journal on Advances in Signal Processing, 2008:371621. DOI: 10.1155/2008/371621
35. Sateesh Babu G., Suresh S. (2013). Parkinson's disease prediction using gene expression – A projection based learning meta-cognitive neural classifier approach. Expert Systems with Applications, 40(5), 1519–1529.
36. Geman O. (2014). Towards an inclusive Parkinson's screening system. In 18th International Conference on System Theory, Control and Computing (ICSTCC), 470-475.

37. Ramani, G.R, Sivagami, G. (2011). Parkinson Disease Classification using Data Mining Algorithms. International Journal of Computer Applications 32(9):17-22.
38. Rustempasic I., Can M. (2013). Diagnosis of Parkinson's disease using Fuzzy C-Means Clustering and Pattern Recognition. In SouthEast Europe Journal of Soft Computing, 2, 42–49.
39. Gelsvartas J., Simutis R., Maskeliunas R. (2016). User Adaptive Text Predictor for Mentally Disabled Huntington's Patients. Comp. Int. and Neurosc. 2016: 3054258:1-3054258:6.
40. Lauraitis A., Maskeliūnas R. (2017) Investigation of Predicting Functional Capacity Level for Huntington Disease Patients. In: Damaševičius R., Mikašytė V. (eds) Information and Software Technologies. ICIST 2017. Communications in Computer and Information Science, vol 756, pp. 142-149. Springer, Cham.
41. C. R. Pereira, D. R. Pereira, G. H. Rosa, V. H. Albuquerque, S. A. Weber, C. Hook, and J. P. Papa, "Handwritten dynamics assessment through convolutional neural networks: An application to parkinson's disease identification," Artificial Intelligence in Medicine, vol. 87, pp. 67–77, may 2018.
42. C. R. Pereira, D. R. Pereira, S. A. Weber, C. Hook, V. H. C. de Albuquerque, and J. P. Papa, "A survey on computer-assisted parkinson's disease diagnosis," Artificial Intelligence in Medicine, sep 2018.
43. D. Gupta, A. Julka, S. Jain, T. Aggarwal, A. Khanna, N. Arunkumar, and V. H. C. de Albuquerque, "Optimized cuttlefish algorithm for diagnosis of parkinson's disease," Cognitive Systems Research, vol. 52, pp. 36–48, dec 2018.
44. D. Gupta, S. Sundaram, A. Khanna, A. E. Hassanien, and V. H. C. de Albuquerque, "Improved diagnosis of parkinson's disease using optimized crow search algorithm," Computers & Electrical Engineering, vol. 68, pp. 412–424, may 2018.
45. V. Papaioannou, G.-P. K. Economou, and A. Tsakalidis, "A robust smart device app assisting medical diagnosis," Artificial Intelligence Research, vol. 5, no. 2, p. 82, 2016.
46. A. Bourouis, M. Feham, M. A. Hossain, and L. Zhang, "An intelligent mobile based decision support system for retinal disease diagnosis," Decision Support Systems, vol. 59, pp. 341–350, 2014.
47. C. Liu, H. Sun, N. Du, S. Tan, H. Fei, W. Fan, T. Yang, H. Wu, Y. Li, and C. Zhang, "Augmented lstm framework to construct medical self-diagnosis android," in Data Mining (ICDM), 2016 IEEE 16th International Conference on. IEEE, 2016, pp. 251–260.
48. P. R. Bhandari, S. P. Yadav, S. A. Mote, D. P. Rankhambe, U. Scholar, and P. APCOER, "Predictive system for medical diagnosis with expertise analysis," International Journal of Engineering Science, vol. 4652, 2016.

49. C. D. Billones, O. J. L. D. Demetria, D. E. D. Hostallero, and P. C. Naval, “Demnet: A convolutional neural network for the detection of alzheimer’s disease and mild cognitive impairment,” in Region 10 Conference (TENCON), 2016 IEEE. IEEE, 2016, pp. 3724–3727.
50. R. Myers and A. Falek, “Quantification of muscle tremor of huntington’s disease patients and their offspring in an early detection study.” Biological psychiatry, vol. 14, no. 5, pp. 777–789, 1979.
51. H. Dai, P. Zhang, and T. Lueth, “Quantitative assessment of parkinsonian tremor based on an inertial measurement unit,” Sensors, vol. 15, no. 10, pp. 25 055–25 071, sep 2015. [Online]. Available: https://doi.org/10.3390/s151025055
52. D. Wu, K. Warwick, Z. Ma, M. N. Gasson, J. G. Burgess, S. Pan, and T. Z. Aziz, “Prediction of parkinson’s disease tremor onset using a radial basis function neural network based on particle swarm optimization,” International journal of neural systems, vol. 20, no. 02, pp. 109–116, 2010
53. B. T. Cole, S. H. Roy, C. J. De Luca, and S. H. Nawab, “Dynamic neural network detection of tremor and dyskinesia from wearable sensor data,” in Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE. IEEE, 2010, pp. 6062–6065.
54. C. Azad, S. Jain, and V. K. Jha, “Design and analysis of data mining based prediction model for parkinson’s disease,” Issues, vol. 1, no. 1, pp. 181–189.
55. A. Bermeo, M. Bravo, M. Huerta, and A. Soto, “A system to monitor tremors in patients with parkinson’s disease,” in Engineering in Medicine and Biology Society (EMBC), 2016 IEEE 38th Annual International Conference of the. IEEE, 2016, pp. 5007–5010.
56. P. Zham, S. P. Arjunan, S. Raghav, and D. K. Kumar, “Efficacy of guided spiral drawing in the classification of parkinson's disease,” IEEE Journal of Biomedical and Health Informatics, vol. 22, no. 5, pp. 1648–1652, sep 2018.
57. E. J. Smits, A. J. Tolonen, L. Cluitmans, M. van Gils, R. C. Zietsma, R. W. K. Borgemeester, T. van Laar, and N. M. Maurits, “Graphical tasks to measure upper limb function in patients with parkinson's disease: Validity and response to dopaminergic medication,” IEEE Journal of Biomedical and Health Informatics, vol. 21, no. 1, pp. 283–289, jan 2017.
58. K. L. de Ipina, J. Solé-Casals, M. Faúndez-Zanuy, P. Calvo, E. Sesa, J. Roure, U. M. de Lizarduy, B. Beitia, E. Fernández, J. Iradi, J. Garcia- Melero, and A. Bergareche, “Automatic analysis of archimedes’ spiral for characterization of genetic essential tremor based on shannon’s entropy and fractal dimension,” Entropy, vol. 20, no. 7, p. 531, jul, 2018. [Online]. Available: https://doi.org/10.3390/e20070531
59. N. Zhi, B. Jaeger, A. Gouldstone, R. Sipahi, and S. Frank, “Toward monitoring parkinson’s through analysis of static handwriting samples: A quantitative analytical framework,” IEEE Journal of Biomedical and Health Informatics, pp. 1–1, 2016.

60. M. Moetesum, I. Siddiqi, N. Vincent, and F. Cloppet, “Assessing visual attributes of handwriting for prediction of neurological disorders - a case study on parkinson’s disease,” Pattern Recognition Letters, 2018.
61. J. Bermeo, M. Huerta, M. Bravo, and A. Bermeo, “Artificial neural network applied like qualifier of symptoms in patients with parkinson’s disease by evaluating the movement of upper-limbs activities.” World Congress on Medical Physics and Biomedical Engineering 2018, pp. 409–414.
62. A. Zhang, A. Cebulla, S. Panev, J. Hodgins, and F. D. la Torre, “Weakly-supervised learning for parkinson's disease tremor detection,” in 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). IEEE, jul 2017.
63. E. Abdulhay, N. Arunkumar, K. Narasimhan, E. Vellaiappan, and V. Venkatraman, “Gait and tremor investigation using machine learning techniques for the diagnosis of parkinson disease,” Future Generation Computer Systems, vol. 83, pp. 366–373, 2018.
64. S. Aich, K. Younga, K. L. Hui, A. A. Al-Absi, and M. Sain, “A nonlinear decision tree based classification approach to predict the parkinson’s disease using different feature sets of voice data.” IEEE 2018 20th International Conference on Advanced Communication Technology (ICACT).
65. G. Massimiliano, P. Giampaoloa, C. Daniela, K. Schruers, R. Duara, and A. D. Loewenstein, “A clinically-translatable machine learning algorithm for the prediction of alzheimer’s disease conversion in individuals with mild and premild cognitive impairment,” Journal of Alzheimer’s Disease, vol. 61, pp. 1555–1573, 2018.
66. A. Lins, M. Muniz, A. Garcia, A. Gomes, R. Cabral, and C. Bastos- Filhob, “Using artificial neural networks to select the parameters for the prognostic of mild cognitive impairment and dementia in elderly individuals,” Computer Methods and Programs in Biomedicine, vol. 152, pp. 93–104, 2017.
67. M. Perez, W. Jin, D. Le, N. Carlozzi, P. Dayalu, A. Roberts, and E. M. Provost, “Classification of huntington disease using acoustic and lexical features.” Interspeech, 2018, pp. 1898–1902.
68. D. Lamichhane and M. Armstrong, “Systematic review of parkinson’s disease related mobile applications,” Neurology, vol. 84, no. S14, p. 7.293, 2015.
69. A. Sahyoun, K. Chehab, O. Al-Madani, F. Aloul, and A. Sagahyroon, “ParkNosis: Diagnosing parkinsons disease using mobile phones,” in 2016 IEEE 18th International Conference on e-Health Networking, Applications and Services (Healthcom). IEEE, sep 2016. [Online].Available: https://doi.org/10.1109/healthcom.2016.7749491.
70. W. Lee, A. Evans, and D. R. Williams, “Validation of a smartphone application measuring motor function in parkinson’s disease,” Journal of Parkinson’s Disease, vol. 6, no. 2, pp. 371–382, May 2016. [Online]. Available: http://doi.org/10.3233/JPD-150708

71. B. Klimova, “Mobile devices and mobile applications used in parkinson’s disease,” in Mobile Web and Intelligent Information Systems. Springer International Publishing, 2017, pp. 137–143
72. D. Roalf, P. Rupert, D. Mechanic-Hamilton, L. Brennan, J. Duda, D. Weintraub, J. Trojanowski, D. Wolk, and P. Moberg, “Quantitative assessment of finger tapping characteristics in mild cognitive impairment, alzheimer’s disease, and parkinson’s disease,” Journal of Neurology, vol. 265, pp. 1365–1375, 2018.
73. J. Hyoseon, L. Woongwoo, P. Hyeyoung, L. H. Ji, K. S. Kyong, K. H. Byul, J. Beomseok, and P. K. Suk, “High-accuracy automatic classification of parkinsonian tremor severity using machine learning method,” Physiological Measurement, vol. 11, 2017.
74. S. Liu, C. Miao, J. M. McKeown, J. Jun, S. Zhiqi, and L. Cyril, “Pumpkin garden: A mobile game platform for monitoring parkinson’s disease symptoms.” ITAP 2018 (Human Aspects of IT for the Aged Population. Applications in Health, Assistance, and Entertainment), 2018, pp. 546–560.
75. A. Vianello, L. Chittaro, S. Burigat, and R. Budai, “Motorbrain: A mobile app for the assessment of users’ motor performance in neurology,” Computer Methods and Programs in Biomedicine, vol. 143, pp. 35–47, 2017.
76. (2018) Uci machine learning repository. [Online]. Available: https://archive.ics.uci.edu/ml/index.php
77. D. Lupton and A. Jutel, “’it’s like having a physician in your pocket! A critical analysis of self-diagnosis smartphone apps,” Social Science & Medicine, vol. 133, pp. 128–135, 2015.
78. D. W. Scharre, S.-I. Chang, R. A. Murden, J. Lamb, D. Q. Beversdorf, M. Kataki, H. N. Nagaraja, and R. A. Bornstein, “Self-administered gerocognitive examination (SAGE),” Alzheimer Disease & Associated Disorders, vol. 24, no. 1, pp. 64–71, jan 2010.

    [Online]. Available: https://doi.org/10.1097/wad.0b013e3181b03277
79. Gaba AM, Zhang K, Marder K, Moskowitz CB, Werner P, Boozer CN (2005). Energy balance in early-stage Huntington disease. *Am J Clin Nutr*. 2005;81(6):1335-41.
80. Marleen R. van Walsem et al. Assistive Technology for Cognition and Health-related Quality of Life in Huntington’s Disease (2016). *J Huntingtons Dis*. 2016; 5(3): 261–270.
81. Memedi, M., et al. (2015). Automatic Spiral Analysis for Objective Assessment of Motor Symptoms in Parkinson’s disease. *Sensors*. 15(9), 23727-23744. DOI: 10.3390/s150923727.
82. Sadikov, A., et al. (2014). ParkinsonCheck smart phone app. *Frontiers in Artificial Intelligence and Applications.* DOI: 10.3233/978-1-61499-419-0-1213.
83. Surangsrirat, D., Thanawattano, C. (2012). Android application for spiral analysis in Parkinson's disease. *Proceedings of IEEE Southeastcon.* DOI: 10.1109/SECon.2012.6196943.

84. Eiter, T., Mannila, H. (1994). Computing Discrete Frechet Distance. Technische Universit¨at Wien. [Online]. Available:
http://www.kr.tuwien.ac.at/staff/eiter/et-archive/cdtr9464.pdf
85. Winkler W. E. (1990). String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage, Proceedings of the Section on Survey Research Methods, American Statistical Association, 354–359.
86. Nikolic, M. (2010). Measuring Similarity of Graphs and their Nodes by Neighbor Matching. *Computer Science, Artificial Intelligence*. arXiv:1009.5290. [Online]. Available: https://arxiv.org/pdf/1009.5290.pdf.
87. Math Planet (2019). Angles and parallel lines. [Online]. Available: https://www.mathplanet.com/education/pre-algebra/introducing-geometry/angles-and-parallel-lines
88. OER. Fruits from A-Z. (2019). Half Your Plate [Online]. Available: http://www.halfyourplate.ca/fruits-and-veggies/fruits-a-z/
89. OER. Animals. (2019). Vocabulary [Online]. Available: https://a-z-animals.com/animals/
90. OER. Countries and Regions of the World from A to Z. (2019). Vocabulary [Online]. Available: https://www.nationsonline.org/oneworld/countries_of_the_world.htm
91. OER. Things in kitchen (2019). Vocabulary [Online]. Available: https://www.englisch-hilfen.de/en/words/kitchen.htm
92. E.Vaiciukynas et al. (2017). Detecting Parkinson's disease from sustained phonation and speech signals. In NCBI (National Center for Biotechnology Information advances science and health), vol 12(10), DOI: 10.1371/journal.pone.0185613.
93. Khan T, Westin J, Dougherty M (2014). Cepstral separation difference: A novel approach for speech impairment quantification in Parkinson's disease. Biocybernetics and Biomedical Engineering. 2014;34(1):25–34. doi: 10.1016/j.bbe.2013.06.001
94. Tsanas A, Little MA, Fox C, Ramig LO (2014). Objective Automatic Assessment of Rehabilitative Speech Treatment in Parkinson's Disease. IEEE Transactions on Neural Systems and Rehabilitation Engineering. 2014;22(1):181–190. doi: 10.1109/TNSRE.2013.2293575
95. An G, Brizan DG, Ma M, Morales M, Syed AR, Rosenberg A (2015). Automatic Recognition of Unified Parkinson's Disease Rating from Speech with Acoustic, i-Vector and Phonotactic Features. In: 16th Annual Conference of the International Speech Communication Association (INTERSPEECH). Dresden, Germany; 2015.
96. Caesarendra W, Putri FT, Ariyanto M, Setiawan JD (2015). Pattern Recognition Methods for Multi Stage Classification of Parkinson's Disease Utilizing Voice Features. In: 2015 IEEE International Conference on Advanced Intelligent Mechatronics (AIM). Busan, Korea: IEEE; 2015. p. 802–807.

97. N. A. Aziz et al. Weight loss in Huntington disease increases with higher CAG repeat number (2008). Neurology November 4, 2008 vol. 71 no. 19 1506-1513.
98. Goodman AO, Murgatroyd PR, Medina-Gomez G, Wood NI, Finer N, Vidal-Puig AJ, Morton AJ, Barker RA (2008). The metabolic profile of early Huntington's disease--a combined human and transgenic mouse study. Exp Neurol. 2008;210(2):691-8.
99. Gaba AM, Zhang K, Marder K, Moskowitz CB, Werner P, Boozer CN. Energy balance in early-stage Huntington disease (2005). Am J Clin Nutr. 2005;81(6):1335-41.
100. Richard E. Pratley et al. (2000). Higher sedentary energy expenditure in patients with Huntington's disease, Annals of neurology. Volume 47, Issue 1, January 2000, Pages 64–70
101. Gil Polo C, Cubo Delgado E, Mateos Cachorro A, Rivadeneyra Posadas J, Mariscal Pérez N, Armesto Formoso D. (2015) Energy Balance in Huntington's Disease. Ann Nutr Metab. 2015;67(4):267-73.
102. Fanny Mochel et al (2007). Early Energy Deficit in Huntington Disease: Identification of a Plasma Biomarker Traceable during Disease Progression. PLoS ONE 2(7): e647.
103. Amirkalali, B., Hosseini, S., Heshmat, R., Larijani B. (2008). Comparison of Harris Benedict and Mifflin-ST Jeor equations with indirect calorimetry in evaluating resting energy expenditure. *Indian J Med Sci*. 62(7):283-90. PMID: 18688113.
104. Weight Loss Tips - Diet, Nutrition, & Exercise Advice and Plans (2019). WeightLossForAll [Online]. Available: http://www.weightlossforall.com/
105. Metabolic Equivalent of Task Values for Activities in American Time Use Survey and 2002 Census Occupational Classification System (2019). *National Cancer Institute*. [Online]. Available:

https://epi.grants.cancer.gov/physical/MET/?major%255B%255D=05#atus
106. Hazewinkel, Michiel, ed. (2001) [1994], "Linear interpolation", Encyclopedia of Mathematics, Springer Science+Business Media B.V. / Kluwer Academic Publishers, ISBN 978-1-55608-010-4
107. Atal, B.S. "Automatic Speaker Recognition Based on Pitch Contours." The Journal of the Acoustical Society of America. Vol. 52, No. 6B, 1972, pp. 1687–1697.
108. Gonzalez, Sira, and Mike Brookes. "A Pitch Estimation Filter robust to high levels of noise (PEFAC)."
109. Noll, Michael A. "Cepstrum Pitch Determination." The Journal of the Acoustical Society of America. Vol. 31, No. 2, 1967, pp. 293–309.
110. Hermes, Dik J. "Measurement of Pitch by Subharmonic Summation." The Journal of the Acoustical Society of America. Vol. 83, No. 1, 1988, pp. 257–264.

111. Drugman, Thomas, and Abeer Alwan. "Joint Robust Voicing Detection and Pitch Estimation Based on Residual Harmonics." Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH. 2011, pp. 1973–1976.

112. Rabiner, Lawrence R., and Ronald W. Schafer. Theory and Applications of Digital Speech Processing. Upper Saddle River, NJ: Pearson, 2010.

113. Shao, Yang, Zhaozhang Jin, Deliang Wang, and Soundararajan Srinivasan. "An Auditory-Based Feature for Robust Speech Recognition." IEEE International Conference on Acoustics, Speech and Signal Processing. 2009.

114. Valero, X., and F. Alias. "Gammatone Cepstral Coefficients: Biologically Inspired Features for Non-Speech Audio Classification." IEEE Transactions on Multimedia. Vol. 14, Issue 6, 2012, pp. 1684–1689

115. Andén, J., and S. Mallat. "Deep Scattering Spectrum." IEEE Transactions on Signal Processing. Vol. 62, Number 16, 2014, pp. 4114–4128.

116. Mallat, S. "Group Invariant Scattering." Communications in Pure and Applied Mathematics. Vol. 65, Number 10, 2012, pp. 1331–1398.

117. Lerch, Alexander. An Introduction to Audio Content Analysis Applications in Signal Processing and Music Informatics. Piscataway, NJ: IEEE Press, 2012

118. Peeters, G. "A Large Set of Audio Features for Sound Description (Similarity and Classification) in the CUIDADO Project." Technical Report; IRCAM: Paris, France, 2004.

119. Scheirer, E., and M. Slaney. "Construction and Evaluation of a Robust Multifeature Speech/Music Discriminator." IEEE International Conference on Acoustics, Speech, and Signal Processing. Volume 2, 1997, pp. 1221–1224.

120. Johnston, J.d. "Transform Coding of Audio Signals Using Perceptual Noise Criteria." IEEE Journal on Selected Areas in Communications. Vol. 6, Number 2, 1988, pp. 314–323.

121. Misra, H., S. Ikbal, H. Bourlard, and H. Hermansky. "Spectral Entropy Based Feature for Robust ASR." 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing.

122. Lauraitis, A., Maskeliūnas. R., Damaševičius, R., Polap, D., Wozniak, M. (2019). A smartphone application for automated decision support in cognitive task based evaluation of central nervous system motor disorders. *IEEE Journal of Biomedical and Health Informatics.* Preprint (Early Access). Electronic ISSN: 2168-2208. DOI 10.1109/JBHI.2019.2891729.

123. Lauraitis, A., Maskeliūnas. R., Damaševičius, R. (2018). ANN and Fuzzy Logic Based Model to Evaluate Huntington Disease Symptoms. Journal of Healthcare Engineering. Volume 2018, Article ID 4581272, 10 pages. DOI: https://doi.org/10.1155/2018/4581272.

124. Yasser EL-Manzalawy (2005). WLSVM. http://www.cs.iastate.edu/~yasser/wlsvm/

125. Chih-Chung Chang, Chih-Jen Lin (2001). LIBSVM - A Library for Support Vector Machines. URL http://www.csie.ntu.edu.tw/~cjlin/libsvm/

126. D. Aha, D. Kibler (1991). Instance-based learning algorithms. Machine Learning. 6:37-66.
127. Ware, M. 2000. WEKA Documentation. University of Waikoto.
128. J. Platt: Fast Training of Support Vector Machines using Sequential Minimal Optimization. In B. Schoelkopf and C. Burges and A. Smola, editors, Advances in Kernel Methods - Support Vector Learning, 1998.
129. S.S. Keerthi, S.K. Shevade, C. Bhattacharyya, K.R.K. Murthy (2001). Improvements to Platt's SMO Algorithm for SVM Classifier Design. Neural Computation. 13(3):637-649.
130. Trevor Hastie, Robert Tibshirani: Classification by Pairwise Coupling. In: Advances in Neural Information Processing Systems, 1998.
131. Fisher, R. A. (1936). "The Use of Multiple Measurements in Taxonomic Problems". Annals of Eugenics. 7 (2): 179–188. doi:10.1111/j.1469-1809.1936.tb02137.x. hdl:2440/15227.
132. Mika, S; Rätsch, G.; Weston, J.; Schölkopf, B.; Müller, KR (1999). Fisher discriminant analysis with kernels. Neural Networks for Signal Processing. IX. pp. 41–48. CiteSeerX 10.1.1.35.9904. doi:10.1109/NNSP.1999.788121. ISBN 978-0-7803-5673-3
133. Deeplearning4j Development Team. (2017) Deeplearning4j: Open-source distributed deep learning for the JVM, Apache Software Foundation License 2.0. http://deeplearning4j.org
134. Leo Breiman (2001). Random Forests. Machine Learning. 45(1):5-32.
135. Ben Gal I (2007). "Bayesian Networks" (PDF). In Ruggeri F, Kennett RS, Faltin FW (eds.). Encyclopedia of Statistics in Quality and Reliability. John Wiley & Sons. doi:10.1002/9780470061572.eqr089. ISBN 978-0-470-01861-3.
136. George H. John, Pat Langley: Estimating Continuous Distributions in Bayesian Classifiers. In: Eleventh Conference on Uncertainty in Artificial Intelligence, San Mateo, 338-345, 1995.
137. Ross Quinlan (1993). C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Mateo, CA.
138. Bottou L. (2012) Stochastic Gradient Descent Tricks. In: Montavon G., Orr G.B., Müller KR. (eds) Neural Networks: Tricks of the Trade. Lecture Notes in Computer Science, vol 7700. Springer, Berlin, Heidelberg.
139. Niels Landwehr, Mark Hall, Eibe Frank (2005). Logistic Model Trees. Machine Learning. 95(1-2):161-205.
140. Marc Sumner, Eibe Frank, Mark Hall: Speeding up Logistic Model Tree Induction. In: 9th European Conference on Principles and Practice of Knowledge Discovery in Databases, 675-683, 2005.

141. Iba, Wayne; and Langley, Pat (1992); Induction of One-Level Decision Trees, in ML92: Proceedings of the Ninth International Conference on Machine Learning, Aberdeen, Scotland, 1–3 July 1992, San Francisco, CA: Morgan Kaufmann, pp. 233–240.
142. Y. Freund, R. E. Schapire: Large margin classification using the perceptron algorithm. In: 11th Annual Conference on Computational Learning Theory, New York, NY, 209-217, 1998.
143. Ron Kohavi, George H. John (1997). Wrappers for feature subset selection. Artificial Intelligence. 97(1-2):273-324.
144. Yoav Freund, Robert E. Schapire: Experiments with a new boosting algorithm. In: Thirteenth International Conference on Machine Learning, San Francisco, 148-156, 1996.
145. Jolliffe I.T. Principal Component Analysis, Series: Springer Series in Statistics, 2nd ed., Springer, NY, 2002, XXIX, 487 p. 28 illus. ISBN 978-0-387-95442-4
146. Benesty J., Chen J., Huang Y., Cohen I. (2009) Pearson Correlation Coefficient. In: Noise Reduction in Speech Processing. Springer Topics in Signal Processing, vol 2. Springer, Berlin, Heidelberg
147. I. Witten, E. Frank, L. Trigg, M. Hall, G. Holmes, and S. Cunningham. Weka: Practical machine learning tools and techniques with java implementations. 1999
148. Ian H. Witten and Eibe Frank. Data mining: Practical machine learning tools and techniques.
149. Ludmila I. Kuncheva (2004). Combining Pattern Classifiers: Methods and Algorithms. John Wiley and Sons, Inc.
150. Devijver, Pierre A.; Kittler, Josef (1982). Pattern Recognition: A Statistical Approach. London, GB: Prentice-Hall. ISBN 0-13-654236-0.
151. Geisser, Seymour (1993). Predictive Inference. New York, NY: Chapman and Hall. ISBN 978-0-412-03471-8.
152. Kohavi, Ron (1995). "A study of cross-validation and bootstrap for accuracy estimation and model selection". Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence. San Mateo, CA: Morgan Kaufmann. 2 (12): 1137–1143. CiteSeerX 10.1.1.48.529.
153. David, S. K., Saeb, A. T., Rafiullah, M., & Rubeaan, K. (2019). Classification Techniques and Data Mining Tools Used in Medical Bioinformatics. In S. Strydom, & M. Strydom (Eds.), Big Data Governance and Perspectives in Knowledge Management(pp. 105-126). Hershey, PA: IGI Global. doi:10.4018/978-1-5225-7077-6.ch005
154. Matthews, B. W. (1975). "Comparison of the predicted and observed secondary structure of T4 phage lysozyme". Biochimica et Biophysica Acta (BBA) - Protein Structure. 405 (2): 442–451.
155. Sasaki, Y. (2007). "The truth of the F-measure" (PDF).

156. McHugh M. L. (2012). Interrater reliability: the kappa statistic. Biochemia medica, 22(3), 276–282.
157. Essenwanger, O. M. (Oskar M.) (1986). Elements of statistical analysis. Elsevier. ISBN 0444424261. OCLC 152410575.
158. Schuster, Mike, and Kuldip K. Paliwal. "Bidirectional recurrent neural networks." Signal Processing, IEEE Transactions on 45.11 (1997): 2673-2681.2. Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan.
159. Sepp Hochreiter; Jürgen Schmidhuber (1997). "Long short-term memory". Neural Computation. 9 (8): 1735–1780. doi:10.1162/neco.1997.9.8.1735. PMID 9377276.
160. Tieleman, Tijmen and Hinton, Geoffrey (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning.
161. Sako, Yusaku (2018-06-02). "Is the term "softmax" driving you nuts?". Medium.
162. Lee, Tai S. (October 1996). "Image Representation Using 2D Gabor wavelets" (PDF). IEEE Transactions on Pattern Analysis and Machine Intelligence. 18 (10): 959–971. doi:10.1109/34.541406.
163. le Cessie, S., van Houwelingen, J.C. (1992). Ridge Estimators in Logistic Regression. Applied Statistics. 41(1):191-201.
164. Fumo, David (2017). Types of Machine Learning Algorithms You Should Know. In magazine *Towards Data Science Sharing concepts, ideas, and codes*. [Online]: https://towardsdatascience.com/types-of-machine-learning-algorithms-you-should-know-953a08248861.
165. Brownlee, Jason (2019). A Tour of The Most Popular Machine Learning Algorithms. In magazine *Machine Learning Algorithms*. Tour of The Most Popular Machine Learning Algorithms. [Online] : https://machinelearningmastery.com/a-tour-of-machine-learning-algorithm
166. Ron Kohavi: The Power of Decision Tables. In: 8th European Conference on Machine Learning, 174-189, 1995.
167. Eibe Frank, Mark Hall, Bernhard Pfahringer: Locally Weighted Naive Bayes. In: 19th Conference in Uncertainty in Artificial Intelligence, 249-256, 2003.
168. C. Atkeson, A. Moore, S. Schaal (1996). Locally weighted learning. AI Review.
169. Florens, Jean-Pierre. *Elements of Bayesian statistics*. Routledge, 2019.
170. Gang Board portal. *Top 10 Machine Learning Algorithms*, 2019. [Online] https://www.gangboard.com/blog/top-10-machine-learning-algorithms/#6.
171. Oleksii Kharkovyna. *Top 10 Machine Learning Algorithms for Data Science*, 2019. *Towards Data Science Magazine*. [Online]: https://towardsdatascience.com/top-10-machine-learning-algorithms-for-data-science-cdb0400a25f9.

172. T. K. Ho. Multiple classifier combination: Lessons and the next steps. In A. Kandel and H. Bunke, editors, Hybrid Methods in Pattern Recognition. World Scientific Publishing, 2002, pp. 171– 19.

173. T. G. Dietterich. Ensemble methods in machine learning. In J. Kittler and F. Roli, editors, Multiple Classifier Systems, volume 1857 of Lecture Notes in Computer Science, Cagliari, Italy, 2000, Springer, pp. 1 –15.

174. Dehzangi, Abdollah, Somnuk Phon-Amnuaisuk, and Omid Dehzangi. "Enhancing protein fold prediction accuracy by using ensemble of different classifiers." Australian Journal of Intelligent Information Processing Systems 26.4 (2010): 32-4.

175. Kuncheva, Ludmila I. "A theoretical study on six classifier fusion strategies." *IEEE Transactions on pattern analysis and machine intelligence* 24.2 (2002): 281-286.

176. Scharre, Douglas W., et al. "Digitally translated Self-Administered Gerocognitive Examination (eSAGE): relationship with its validated paper version, neuropsychological evaluations, and clinical assessments." Alzheimer's research & therapy 9.1 (2017): 44.

177. Nasreddine, Ziad S., et al. "The Montreal Cognitive Assessment, MoCA: a brief screening tool for mild cognitive impairment." Journal of the American Geriatrics Society 53.4 (2005): 695-699.

178. Huisingh, Carrie, et al. "Relationship between Areas of Cognitive Functioning on the Mini-Mental State Examination and Crash Risk." Geriatrics 3.1 (2018): 10.

179. Tsiouris, Kostas M., et al. "PD_Manager: an mHealth platform for Parkinson's disease patient management." Healthcare technology letters 4.3 (2017): 102-108.

180. Roalf, David R., et al. "Quantitative assessment of finger tapping characteristics in mild cognitive impairment, Alzheimer's disease, and Parkinson's disease." Journal of neurology 265.6 (2018): 1365-1375.

181. Morimoto, Chie, et al. "Temporal processing instability with millisecond accuracy is a cardinal feature of sensorimotor impairments in autism spectrum disorder: analysis using the synchronized finger-tapping task." Journal of autism and developmental disorders 48.2 (2018): 351-360.

182. Rose, Dawn, et al. "Music and Metronomes Differentially Impact Motor Timing in People with and without Parkinson's Disease: Effects of Slow, Medium, and Fast Tempi on Entrainment and Synchronization Performances in Finger Tapping, Toe Tapping, and Stepping on the Spot Tasks." Parkinson's Disease 2019 (2019).

183. Payne, Joshua S., et al. "Study protocol for a randomized pilot study of a computer-based, non-pharmacological cognitive intervention for motor slowing and motor fatigue in Parkinson's disease." Pilot and feasibility studies 4.1 (2018): 190.

184. Binder, Julius. Implementation and evaluation of two distinct electronic motion detection devices for the assessment of abnormal movements in Huntington's disease. Diss. Universität Ulm, 2019.
185. Muhamed, Siti Anizah, et al. "Objective Evaluation of Bradykinesia in Parkinson's Disease using Evolutionary Algorithms." BIOSIGNALS. 2018.
186. Đurić-Jovičić, Milica, et al. "Finger and foot tapping sensor system for objective motor assessment." Vojnosanitetski pregled 75.1 (2018): 68-77.
187. Suzumura, Shota, et al. "Differences among patients with Alzheimer's disease, older adults with mild cognitive impairment and healthy older adults in finger dexterity." Geriatrics & gerontology international 18.6 (2018): 907-914.
188. Janzen, Thenille Braun, Marion Haase, and Michael H. Thaut. "Rhythmic priming across effector systems: A randomized controlled trial with Parkinson's disease patients." Human movement science 64 (2019): 355-365.
189. Passos, Leandro A., et al. "Parkinson disease identification using residual networks and optimum-path forest." 2018 IEEE 12th International Symposium on Applied Computational Intelligence and Informatics (SACI). IEEE, 2018.
190. Poon, Christine, et al. "Derivation and Analysis of Dynamic Handwriting Features as Clinical Markers of Parkinson's Disease." Proceedings of the 52nd Hawaii International Conference on System Sciences. 2019.
191. Pham, Hung N., et al. "Multimodal Detection of Parkinson Disease based on Vocal and Improved Spiral Test." 2019 International Conference on System Science and Engineering (ICSSE). IEEE, 2019.
192. Gupta, Deepak, et al. "Optimized cuttlefish algorithm for diagnosis of Parkinson's disease." Cognitive systems research 52 (2018): 36-48.
193. Pereira, Clayton R., et al. "Handwritten dynamics assessment through convolutional neural networks: An application to Parkinson's disease identification." Artificial intelligence in medicine 87 (2018): 67-77.
194. Chen, Kai-Hsiang, et al. "The difference in visuomotor feedback velocity control during spiral drawing between Parkinson's disease and essential tremor." Neurological Sciences 39.6 (2018): 1057-1063.
195. Sharma, Prerna, et al. "Diagnosis of Parkinson's disease using modified grey wolf optimization." Cognitive Systems Research 54 (2019): 100-115.
196. Danna, Jérémy, et al. "Digitalized spiral drawing in Parkinson's disease: A tool for evaluating beyond the written trace." Human movement science 65 (2019): 80-88.
197. Bernardo, Lucas S., et al. "Handwritten pattern recognition for early Parkinson's disease diagnosis." Pattern Recognition Letters 125 (2019): 78-84.
198. Moetesum, Momina, et al. "Assessing visual attributes of handwriting for prediction of neurological disorders—A case study on Parkinson's disease." Pattern Recognition Letters 121 (2019): 19-27.

199. DelMastro, Heather M., et al. "Quantification characteristics of digital spiral analysis for understanding the relationship among tremor and clinical measures in persons with multiple sclerosis." Journal of neuroscience methods 307 (2018): 254-259.

200. M Isenkul, B Sakar, O Kursun, "Improved spiral test using digitized graphics tablet for monitoring Parkinson's disease", *Proc. International Conference on e-Health and Telemedicine*, pp. 171-175, 2014.

201. Craufurd, David et al. Behavioral Changes in Huntington Disease. Neuropsychiatry, Neuropsychology, & Behavioral Neurology: October December 2001 - Volume 14 - Issue 4 - pp 219-226.

202. K. Duff et al. Mild cognitive impairment in prediagnosed Huntington disease. Neurology August 10, 2010 vol. 75 no. 6 500-507.

203. Nehl, Carissa et al. Cognitive and Psychiatric Aspects of Huntington Disease Contribute to Functional Capacity. Journal of Nervous & Mental Disease: January 2004 - Volume 192 - Issue 1 - pp 72-74.

204. Pascale Riba, et al. Psychiatric and Cognitive Difficulties as Indicators of Juvenile Huntington Disease Onset in 29 Patients. Arch Neurol. 2007;64(6):813-819. doi:10.1001/archneur.64.6.81.

205. Marleen R. van Walsem et al. Assistive Technology for Cognition and Health-related Quality of Life in Huntington's Disease. J Huntingtons Dis. 2016; 5(3): 261–270.

206. Deborah Lynn Harrington et al. Cognitive domains that predict time to diagnosis in prodromal Huntington disease. J Neurol Neurosurg Psychiatry. 2012 Jun; 83(6): 10.1136/jnnp-2011-301732.

207. Lunven, Marine, et al. "F32 Cognitive decline in Hntington's disease (HD) in computerized arithmetic task." (2018): A51-A51.

208. Alloni, Anna, et al. "Evaluation of an ontology-based system for computerized cognitive rehabilitation." *International journal of medical informatics 115* (2018): 64-72.

209. Flannery, Samuel L., et al. "Computerized testing in Parkinson's disease: Performance deficits in relation to standard clinical measures. *Journal of clinical and experimental neuropsychology*. 40.10 (2018): 1062-1073.

210. Maggio, Maria G., et al. "Personal computer-based cognitive training in Parkinson's disease: a case study." Psychogeriatrics 18.5 (2018): 427-429.

211. Wu, Jia-Yun, et al. "Computerized Cognitive Assessment System for Dementia Screening Application." World Congress on Medical Physics and Biomedical Engineering 2018. Springer, Singapore, 2019.

212. Wiloth, Stefanie, et al. "Motor-cognitive effects of a computerized game-based training method in people with dementia: a randomized controlled trial." Aging & mental health 22.9 (2018): 1130-1141.

213. Betrouni, Nacim, et al. "Electroencephalography-based machine learning for cognitive profiling in Parkinson's disease: Preliminary results. *Movement Disorders*, 34.2 (2019): 210-217.
214. Gillivan-Murphy, Patricia, Nick Miller, and Paul Carding. "Voice tremor in Parkinson's disease: an acoustic study." *Journal of Voice* 33.4 (2019): 526-535.
215. Gaballah, Amr, et al. "Objective and Subjective Speech Quality Assessment of Amplification Devices for Patients with Parkinson's Disease." *IEEE Transactions on Neural Systems and Rehabilitation Engineering* (2019).
216. Hauptman, Yermiyahu et al. Identifying distinctive acoustic and spectral features in Parkinson's disease. INTERSPEECH 2019.
217. Wu, Kebin, et al. "Learning acoustic features to detect Parkinson's disease." *Neurocomputing* 318 (2018): 102-108.
218. Perez, Matthew, et al. "Classification of Huntington Disease Using Acoustic and Lexical Features." *Interspeech*. 2018.
219. Sakar, C. Okan, et al. "A comparative analysis of speech signal processing algorithms for Parkinson's disease classification and the use of the tunable Q-factor wavelet transform." *Applied Soft Computing* 74 (2019): 255-263.
220. Oung, Qi Wei, et al. "Empirical Wavelet Transform Based Features for Classification of Parkinson's Disease Severity." *Journal of medical systems* 42.2 (2018): 29.
221. Ali, Humair, et al. "Sound Classification of Parkinsonism for Telediagnosis." *Technical Journal* 24.01 (2019): 90-97.
222. Kuresan, Harisudha, Dhanalakshmi Samiappan, and Sam Masunda. "Fusion of WPT and MFCC feature extraction in Parkinson's disease diagnosis." *Technology and Health Care* Preprint (2019): 1-10.
223. Burk, Brittany R., and Christopher R. Watts. "The effect of Parkinson disease tremor phenotype on cepstral peak prominence and transglottal airflow in vowels and speech." *Journal of Voice* 33.4 (2019): 580-e11.
224. Jeancolas, Laetitia, et al. "Comparison of Telephone Recordings and Professional Microphone Recordings for Early Detection of Parkinson's Disease, using Mel-Frequency Cepstral Coefficients with Gaussian Mixture Models." *Proc. Interspeech 2019* (2019): 3033-3037.
225. Cersosimo, Maria G., et al. "Weight Loss in Parkinson's Disease: The Relationship with Motor Symptoms and Disease Progression." *BioMed research international* 2018 (2018).
226. Verbeken, Sandra, et al. "Computer training of attention and inhibition for youngsters with obesity: A pilot study." *Appetite* 123 (2018): 439-447.
227. Singh, Balbir, and Hissam Tawfik. "A Machine Learning Approach for Predicting Weight Gain Risks in Young Adults." *2019 10th International Conference on Dependable Systems, Services and Technologies (DESSERT)*. IEEE, 2019.

228. Kubben, Pieter L., et al. "TREMOR12: an open-source mobile app for tremor quantification." Stereotactic and functional neurosurgery 94.3 (2016): 182-186.
229. Pan, Di, et al. "A mobile cloud-based Parkinson's disease assessment system for home-based monitoring." JMIR mHealth and uHealth 3.1 (2015): e29.
230. Kuosmanen, Elina, et al. "Mobile-based Monitoring of Parkinson's Disease." Proceedings of the 17th International Conference on Mobile and Ubiquitous Multimedia. ACM, 2018.
231. Villasana, María Vanessa, et al. "CoviHealth: Novel approach of a mobile application for nutrition and physical activity management for teenagers." Proceedings of the 5th EAI International Conference on Smart Objects and Technologies for Social Good. ACM, 2019.
232. Tirasirichai, Benjarat, et al. "Bloom Balance: Calorie Balancing Application With Scientific Validation." 2018 15th International Joint Conference on Computer Science and Software Engineering (JCSSE). IEEE, 2018.
233. Merchant, Kaiz, and Yash Pande. "ConvFood: A CNN-Based Food Recognition Mobile Application for Obese and Diabetic Patients." Emerging Research in Computing, Information, Communication and Applications. Springer, Singapore, 2019. 493-502.
234. Cohen, Shani, et al. "Characterizing patient compliance over six months in remote digital trials of Parkinson's and Huntington disease." *BMC medical informatics and decision making* 18.1 (2018): 138.
235. Arora S, Venkataraman V, Zhan A, Donohue S, Biglan KM, Dorsey ER, et al. Detecting and monitoring the symptoms of Parkinson's disease using smartphones: a pilot study. Park Relat Disord [internet]. Elsevier Ltd. 2015;21:650–3 Available from: https://doi.org/10.1016/j.parkreldis.2015.02.026.
236. Ferreira JJ, Godinho C, Santos AT, Domingos J, Abreu D, Lobo R, et al. Quantitative home-based assessment of Parkinson's symptoms: the SENSE-PARK feasibility and usability study. BMC Neurol [internet]. BMC Neurol. 2015;15:1–7 Available from: https://doi.org/10.1186/s12883-015-0343-z.
237. De Lima ALS, Hahn T, Evers LJW, De Vries NM, Cohen E, Afek M, et al. Feasibility of large-scale deployment of multiple wearable sensors in Parkinson's disease. PLoS One. 2017;12:1–15.
238. Bot BM, Suver C, Neto EC, Kellen M, Klein A, Bare C, et al. The mPower study, Parkinson disease mobile data collected using ResearchKit. Sci Data. 2016;3:160011 Available from: http://www.nature.com/articles/sdata201611.
239. Zhan A, Little MA, Harris DA, Abiola SO, Dorsey ER, Saria S, et al. High Frequency Remote Monitoring of Parkinson's Disease via Smartphone: Platform Overview and Medication Response Detection 2016;1–12. Available from: http://arxiv.org/abs/1601.00960

240. Lakshminarayana R, Wang D, Burn D, Chaudhuri KR, Galtrey C, Guzman NV, et al. Using a smartphone-based self-management platform to support medication adherence and clinical consultation in Parkinson ’s disease. Npj Park dis. 2016;2017:1–8 Available from: https://doi.org/10.1038/s41531-016-0003-z.

241. Silva de Lima AL, Hahn T, de Vries NM, Cohen E, Bataille L, Little MA, et al. Large-Scale Wearable Sensor Deployment in Parkinson’s Patients: The Parkinson@Home Study Protocol. JMIR Res Protoc. 2016;5:e172 Available from: http://www.researchprotocols.org/2016/3/e172/.

242. Del Din S, Godfrey A, Mazzà C, Lord S, Rochester L. Free-living monitoring of Parkinson’s disease: lessons from the field. Mov Disord. 2016;31:1293–313.

243. Adams JL, Dinesh K, Xiong M, Tarolli CG, Sharma S, Sheth N, et al. Multiple wearable sensors in Parkinson and Huntington disease individuals: a pilot study in clinic and at home. Digit Biomarkers Karger Publishers. 2017;1:52–63.

244. Andrzejewski KL, Dowling AV, Stamler D, Felong TJ, Harris DA, Wong C, et al. Wearable sensors in Huntington disease: a pilot study. J Huntingtons Dis IOS Press. 2016;5:199–206.

245. Linares-Del Rey, M., L. Vela-Desojo, and R. Cano-de la Cuerda. "Mobile phone applications in Parkinson's disease: A systematic review." *Neurología (English Edition)* (2018).

246. Estévez-Martín, Sonia, et al. "Mobile Applications for People with Parkinson’s Disease: A Systematic Search in App Stores and Content Review." *Journal of Universal Computer Science* 25.7 (2019): 740-763.

247. Stanković, Radomir S.; Astola, Jaakko T. (2012). "Reminiscences of the Early Work in DCT: Interview with K.R. Rao" (PDF). Reprints from the Early Days of Information Sciences.

248. T. H. Falk, C. Zheng, W.-Y. Chan, "A non-intrusive quality and intelligibility measure of reverberant and dereverberated speech", *IEEE Trans. Audio Speech Language Process.*, vol. 18, no. 7, pp. 1766-1774, Sep. 2010, [online] Available: http://ieeexplore.ieee.org/document/5547575/.

249. F. Chen, O. Hazrati, P. C. Loizou, "Predicting the intelligibility of reverberant speech for cochlear implant listeners with a non-intrusive intelligibility measure", *Biomed. Signal Process. Control*, vol. 8, no. 3, pp. 311-314, May 2013,
Available: http://linkinghub.elsevier.com/retrieve/pii/S17468094120012.

250. V. Grancharov, D. Y. Zhao, J. Lindblom, W. B. Kleijn, "Low-complexity nonintrusive speech quality assessment", *IEEE Trans. Audio Speech Lang. Process.*, vol. 14, no. 6, pp. 1948-1956, Nov. 2006.

251. N. S Nehe and R. S Holambe, “DWT and LPC based feature extraction methods for isolated word recognition,” EURASIP Journal on Audio, Speech, and Music Processing, vol. 2012, no. 1, January, 2012.

252. D. A. Reynolds and R. C. Rose, "Robust text-independent speaker identification using Gaussian mixture speaker models," in *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 1, pp. 72-83, Jan. 1995. doi: 10.1109/89.365379.

253. Varotsos, P. A., et al. "Entropy in the natural time domain." Physical Review E 70.1 (2004): 011106.

254. Misra, Hemant, et al. "Spectral entropy based feature for robust ASR." 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing. Vol. 1. IEEE, 2004.

255. Little M.A., Mcsharry P., Hunter E., Spielman J., Ramig L.Suitability of dysphonia measurements for telemonitoring of Parkinsons disease IEEE Trans. Biomed. Eng., 56 (4) (2009), pp. 1015-1022

256. Peker M.A decision support system to improve medical diagnosis using a combination of k-medoids clustering based attribute weighting and SVM, J. Med. Syst., 40 (5) (2016).

257. Erdogdu Sakar B., Serbes G., Sakar C.Analyzing the effectiveness of vocal features in early telediagnosis of Parkinson's disease. PLoS One, 12 (8) (2017), Article e018242.

258. Peker M., Şen B., Delen D.Computer-aided diagnosis of Parkinson's disease using complex-valued neural networks and mRMR feature selection algorithm. J. Healthcare Eng., 6 (3) (2015), pp. 281-302.

259. D. Le, K. Licata, C. Persad, and E. M. Provost, "Automatic assessment of speech intelligibility for individuals with aphasia," IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 24, no. 11, pp. 2187–2199, 2016.

260. D. W. Scharre, et al. "Self-administered gerocognitive examination (SAGE)", 2010. How to interpret the results. [Online]. Available: http://sagetest.osu.edu.

7. WORK DISSEMINATION

1. Video interview for LNK television in Lithuania. Date: May 2019. [Online]: https://www.facebook.com/ktuif/posts/10157213111908194/
2. Participation in Lithuanian "Radio of Knowledge" show "Mokslo Mygtukas". Date: May 2019. [Online]: https://www.ziniuradijas.lt/laidos/mokslo-mygtukas/tai-pades-nustatyti-itin-reta-liga-jau-ankstyvoje-stadijoje?video=1
3. Dissemination in Lithuanian press. Channel 1. Date: May 2019. [Online]: http://www.elektronika.lt/produktai/programos/68153/atpazinti-reta-liga-nuo-siol-gali-padeti-ir-mobilioji-programele
4. Dissemination in Lithuanian press. Channel 2. Date: May 2019. [Online]: https://www.delfi.lt/mokslas/technologijos/atpazinti-sunkia-liga-nuo-siol-gali-padeti-ir-mobilioji-programele.d?id=81222371
5. Dissemination in Lithuanian press. Channel 3. Date: May 2019. [Online]: https://www.etaplius.lt/atpazinti-reta-liga-nuo-siol-gali-padeti-ir-mobilioji-programele
6. Dissemination in Lithuanian press. Channel 4. Date: May 2019. [Online]: https://www.lrt.lt/naujienos/mokslas-ir-it/11/1060638/rytdienos-pranesimas-padesiantis-gyventi-ilgiau-atpazinti-reta-liga-gali-padeti-ir-mobilioji-programele
7. Dissemination in Lithuanian press. Channel 5. Date: May 2019. [Online]: https://kauno.diena.lt/naujienos/ivairenybes/mokslas-ir-it/atpazinti-reta-liga-nuo-siol-gali-padeti-ir-mobilioji-programele-914677
8. Dissemination in Lithuanian press. Channel 6. Date: May 2019. [Online]: https://klaipeda.diena.lt/naujienos/ivairenybes/mokslas-ir-it/atpazinti-reta-liga-nuo-siol-gali-padeti-ir-mobilioji-programele-914677
9. Dissemination in Lithuanian press. Channel 7. Date: May 2019. [Online]: https://www.diena.lt/naujienos/ivairenybes/mokslas-ir-it/atpazinti-reta-liga-nuo-siol-gali-padeti-ir-mobilioji-programele-914677
10. Dissemination in the worldwide press. Channel 1. Date: June 2019. [Online]: https://www.laboratoryequipment.com/article/2019/06/smartphone-app-can-help-detect-early-symptoms-rare-disease
11. Dissemination in the worldwide press. Channel 2. Date: May 2019. [Online]: https://www.eurekalert.org/pub_releases/2019-05/kuot-sac052119.php
12. Dissemination in the worldwide press. Channel 3. Date: May 2019. [Online]: https://www.business-standard.com/article/pti-stories/new-smartphone-app-may-help-diagnose-rare-genetic-disease-119052200565_1.html
13. Dissemination in the worldwide press. Channel 4. Date: May 2019. [Online]: https://globalgenes.org/2019/05/21/smartphone-app-can-help-diagnose-huntingtons-disease/
14. Dissemination in the worldwide press. Channel 5. Date: May 2019. [Online]: https://medicalxpress.com/news/2019-05-smartphone-app-rare-disease.html

APPENDICES

A. Appendice. Prevalence and genetic origin of Huntington Disease

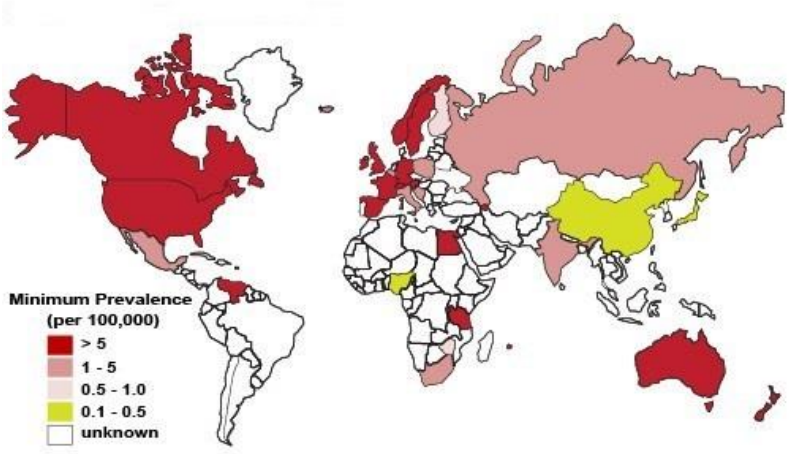


Figure A.1. Huntington Disease Prevalence Estimates Worldwide

HD is a rare progressive genetic disease, i.e., the mutation occurs in the 4-th chromosome of human cell in the DNA gene structure first exon. The 'DNA alphabet' is made up of four letters: C, A, G and T. These four nucleic acid units are combined in different orders to make DNA strands. In its non-disease version, the Huntingtin gene contains up to 35 repeats of the sequence CAG. More than 40 repeats result in a person developing Huntington's disease at some stage in their life (although the age at which they will develop it remains very difficult to predict). 36–39 repeats are grey area: individuals with these repeat numbers may develop Huntington's (the risk increases as the repeat number increases), but some may not develop it within their lifetime. Figure A.2 shows that HD patients have a Huntingtin gene with increased CAG repeats [15].

HD is an autosomal[3] dominant disorder. The child inherits one allele from each parent. The parent without HD has two non-HD alleles; thus, the allele [4] from this parent will be non-HD regardless of which one is inherited. The parent with HD has one non-HD allele and HD allele. There is an equal probability of passing either of these alleles to the child. Thus, the child has a 50% chance of getting the non-HD allele and a 50% chance of getting the HD allele. Since the chance of getting an HD allele is one in two, the child has a 50% chance overall of inheriting the disease (Figure A.3) [16].

[3] A person needs only one copy of the defective gene to develop the disorder.

[4] A version of Huntington gene that contains higher than normal number of repeats. Individuals with the HD allele will develop Huntington's disease. Such allele may be called the mutant Huntington allele in other sources.

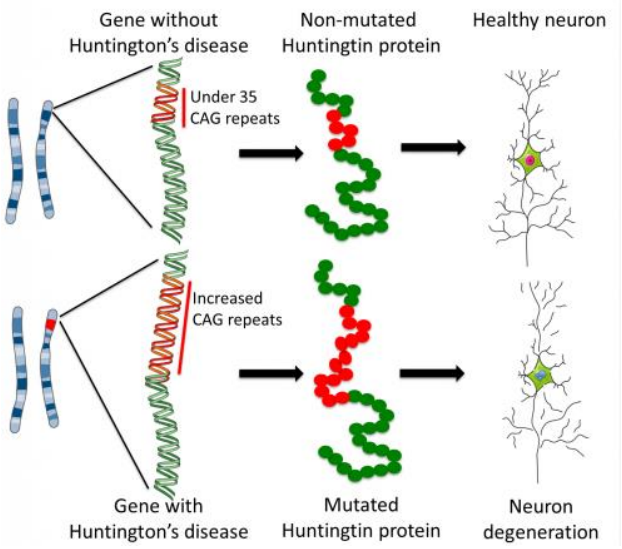


Figure A.2. Genetic origin of Huntington disease

For a grandchild of a person with HD, one copy (allele) will come from the parent who is not at risk. This copy will always be non-HD and does not affect your chances of getting the disease. The second copy comes from your at-risk parent. Since this parent is the child of an individual with HD, he or she has an equal chance of having either two non-HD alleles or one non-HD and one HD allele (Figure 4 in appendix 1). In the first case, this parent has two non-HD alleles and you will not inherit the disease, regardless of which of the two non-HD alleles you will get. This represents a 25% chance that you have inherited the disease [16].

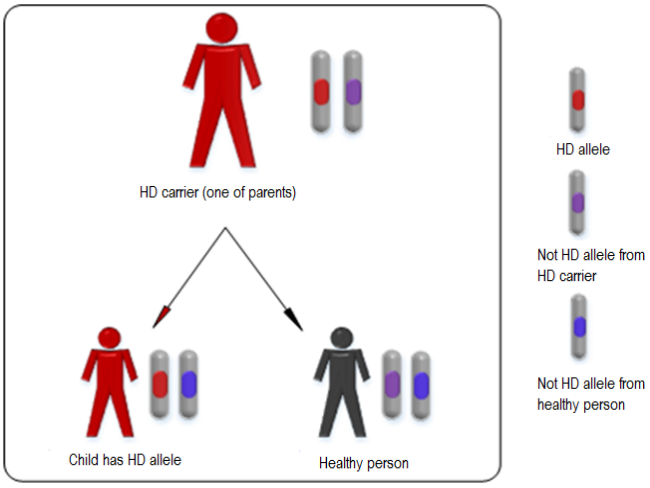


Figure A.3. Risk for having HD (2-nd generation)

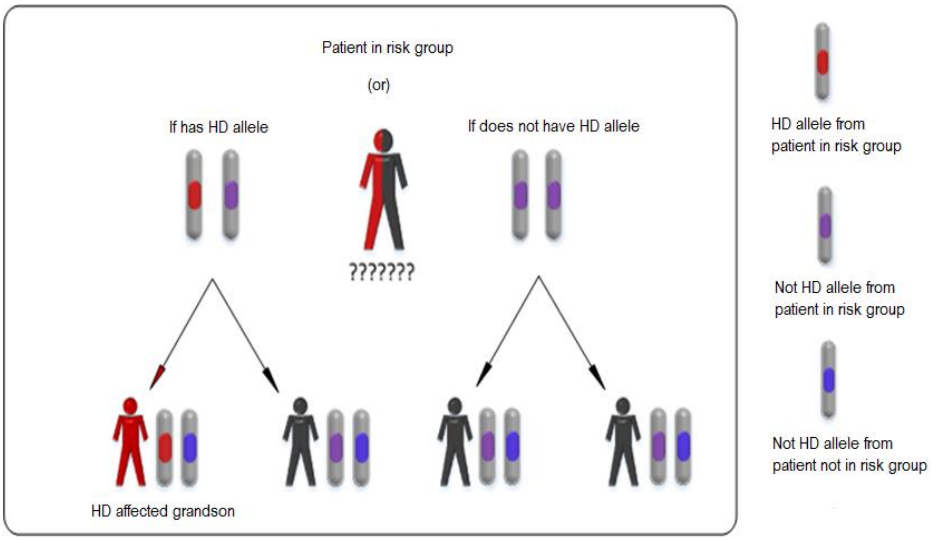
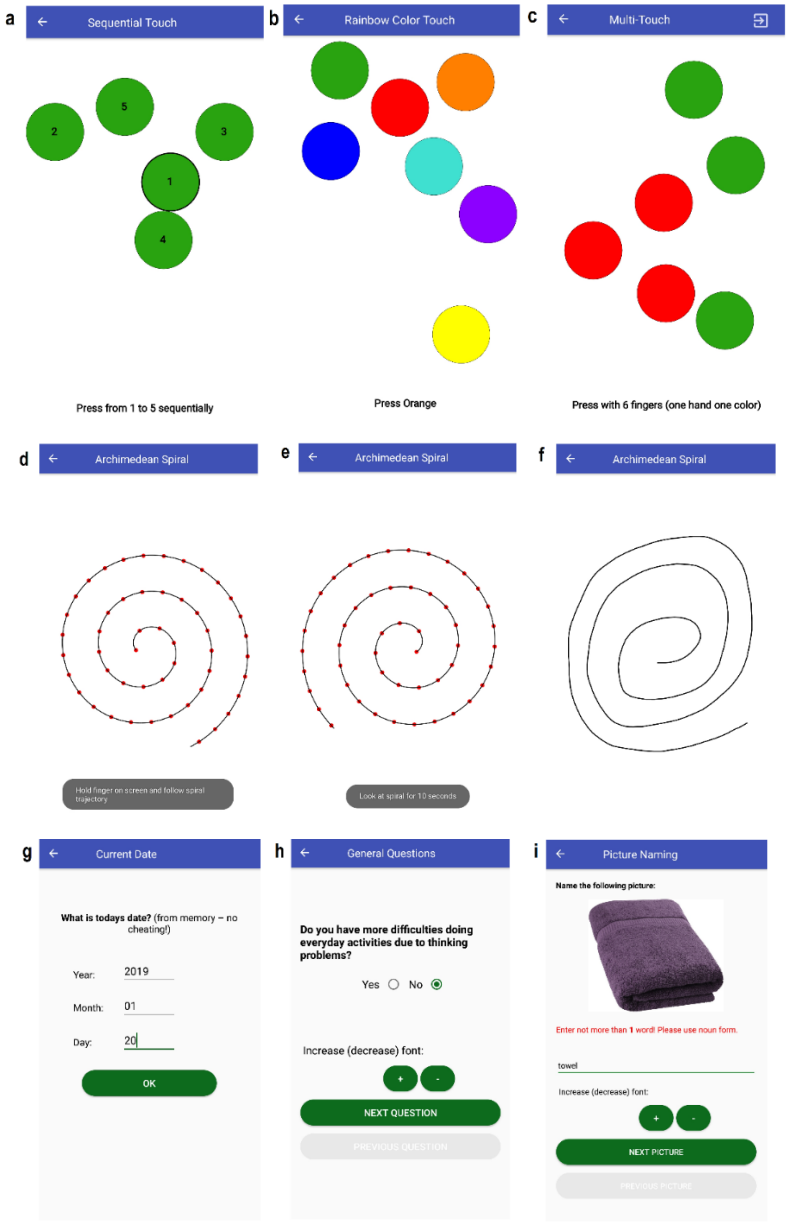


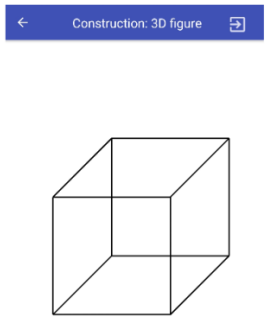
Figure A.4. Risk for having HD (3-nd generation)

B. Appendice. NITS mobile app screenshots and extra features

Figure B.1 in appendice illustrates the core functionality (tremor, cognitive, speech, energy expenditure tasks) for data collection platform implemented as a mobile app., which is available online free of charge as a self-assessment tool on Google Play under the name ‘Neural Impairment test suite’ [18].

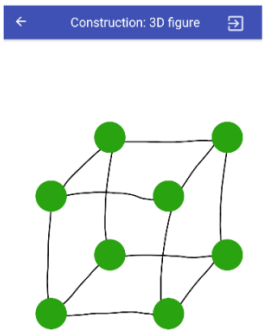


j



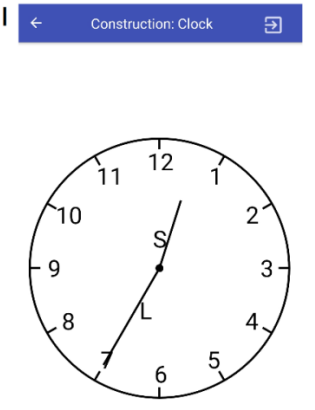
Look at the drawn 3-D figure
Remember connections and click on screen

k



Connection successful
Create another or exit

l

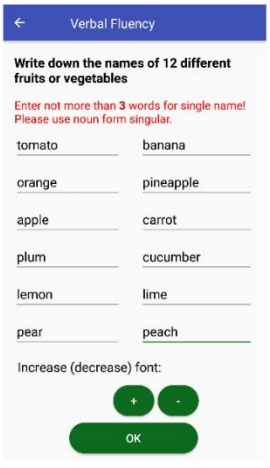


Example: clock is showing 12 h. and 35 min.
Hand labelling: S (hour), L (minute)
Remember clock and click on screen

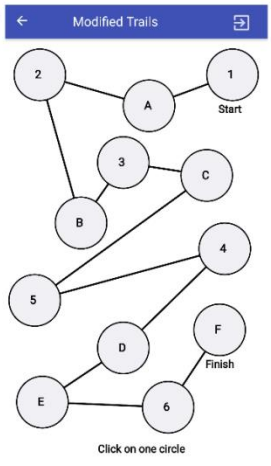
m

Clock is showing: 12 h. and 35 min.
Draw clock face, numbers in red area
Label hands (S, L)

n

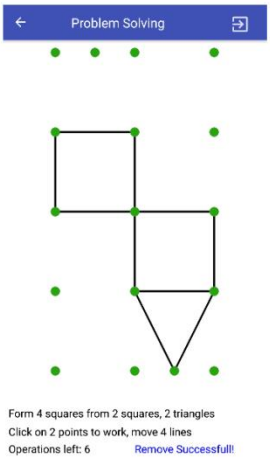


o



Click on one circle

p



Form 4 squares from 2 squares, 2 triangles
Click on 2 points to work, move 4 lines
Operations left: 6 Remove Successfull!

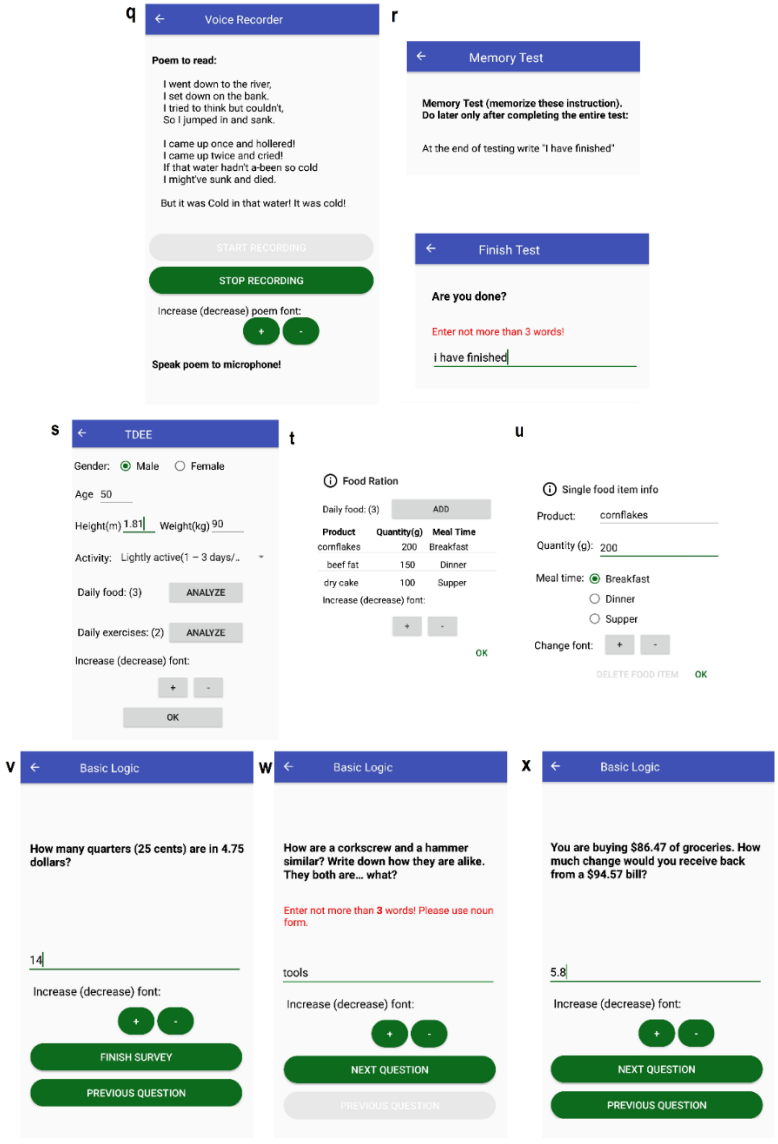
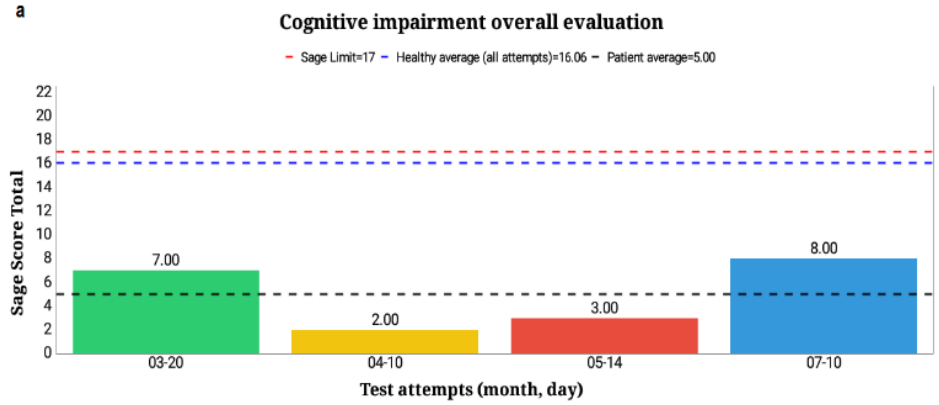


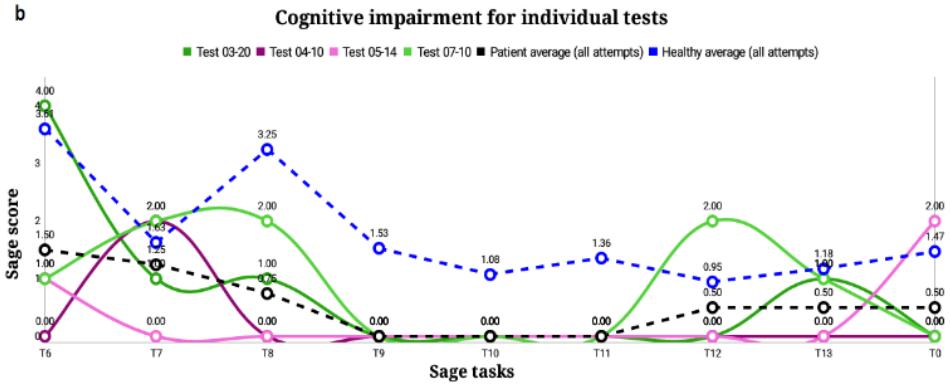
Figure B.1. Screenshots of the proposed mobile app (NITS) tasks: Touches (a – T1:Sequential, b – T2:Rainbow Colour, c – T3:Multi-Touch), T4:Archimedean Spiral (d – following contour clockwise, e – showing spiral counter clockwise, f – drawing contour counter clockwise), T6:Orientation (g), T5:Insights (h), T7:Picture Naming (i), T8:Similarities, Calculation (v - mathematical division, w – similarity search, x – mathematical subtraction), T9:Construction 3-D figure (j – showing, k – constructing cube), T10:Construction Clock (l – showing, m – constructing clock), T11:Verbal Fluency (n – entering 12 items), T12:Modified Trials (o – completing schema), T13:Problem Solving (p – after line remove operation), T14:Voice Recorder (q – reading a poem to a microphone), T16:Memory (r – getting instructions and entering a phrase), T15:TDEE (s – main screen, t – entered food list, u – adding a single food item)

Figure B.2 in appendice shows the health state monitoring using graphical form. The results can be observed in 8 different perspectives by viewing score (metric) for an individual patient. The test attempts are grouped by execution date by illustrating average evaluation of a patient (self-assessment purposes) and healthy subjects median. The example data is collected from Juvenile Huntington's disease (JHD) patient (4 iterations).

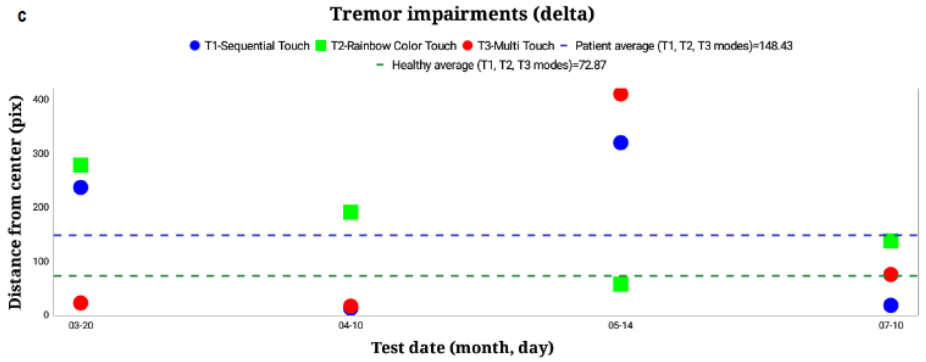
Pic. A (Cognitive Impairment overall evaluation: *SAGE_total* feature, Sage norm for healthy subjects = 17 (red dotted lined), healthy average (blue dotted line)). Pic. B (Cognitive Impairment for individual tests: features *tsage1*, *tsage2*, *tsage_3_4_5*, *tsage7*, *tsage8*, *tsage9*, *tsage10*, *tsage11*, *tsage6*, the solid line shows patient attempts by date, dotted line represents the average values). Pic. C and Pic. D (Tremor Impairments: features used are *t1_delta_avg*, *t2_delta_avg*, *t3_delta_avg*, *t1_reaction_time_avg*, *t2_reaction_time_avg*, *t3_reaction_time_avg*, blue circles represent T1 task, green squares – T2, red circles – T3, dotted lines indicate averages of healthy subjects and the patient). Pic. E (Archimedean Spiral Point matching: features used are *t4_touching_usertouch_percenteage*, *t4_drawing_usertouch_percenteage*, shows patient captured points in touch (blue bar) and draw (yellow bar) modes, compares in percentage with the best achieved results from the test subject). Pic. F (Tremor test duration: patient attempts are grouped by date (solid lines), features used are *t1_test_duration*, *t2_test_duration*, *t3_test_duration*, *t4_touching_test_duration*, *t4_drawing_test_duration*). Pic. G (Sage Drawing test speed: include T9, T10, T12 and T13 tasks, features used are finger motion features v_{avg}). Pic. H (Energy Expenditure: features used are *t15_tdee_calories*, *t15_patient_dailygained*, *t15_patient_dailylburned*, *t15_tdee_dailycaloriebalance*, patient TDEE norm (red line), patient calorie balance average (black line), daily gained calorie balance average (blue line), daily calories gained (for each arranged by the date test, i.e., purple bar), daily calories burned (for each test arranged by date, i.e., green bar).



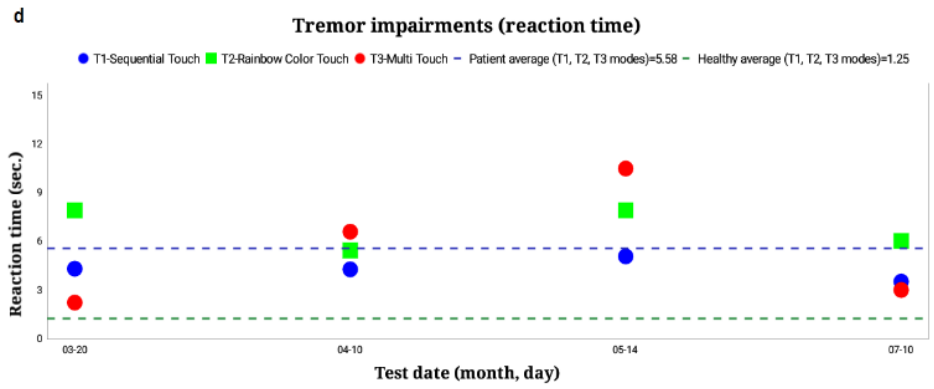
b


Cognitive impairment for individual tests
Test 03-20
Test 04-10
Test 05-14
Test 07-10
Patient average (all attempts)
Healthy average (all attempts)
Sage score
Sage tasks
T6
T7
T8
T9
T10
T11
T12
T13
T0

c


Tremor impairments (delta)
T1-Sequential Touch
T2-Rainbow Color Touch
T3-Multi Touch
Patient average (T1, T2, T3 modes)=148.43
Healthy average (T1, T2, T3 modes)=72.87
Distance from center (pix)
Test date (month, day)
03-20
04-10
05-14
07-10

d


Tremor impairments (reaction time)
T1-Sequential Touch
T2-Rainbow Color Touch
T3-Multi Touch
Patient average (T1, T2, T3 modes)=5.58
Healthy average (T1, T2, T3 modes)=1.25
Reaction time (sec.)
Test date (month, day)
03-20
04-10
05-14
07-10

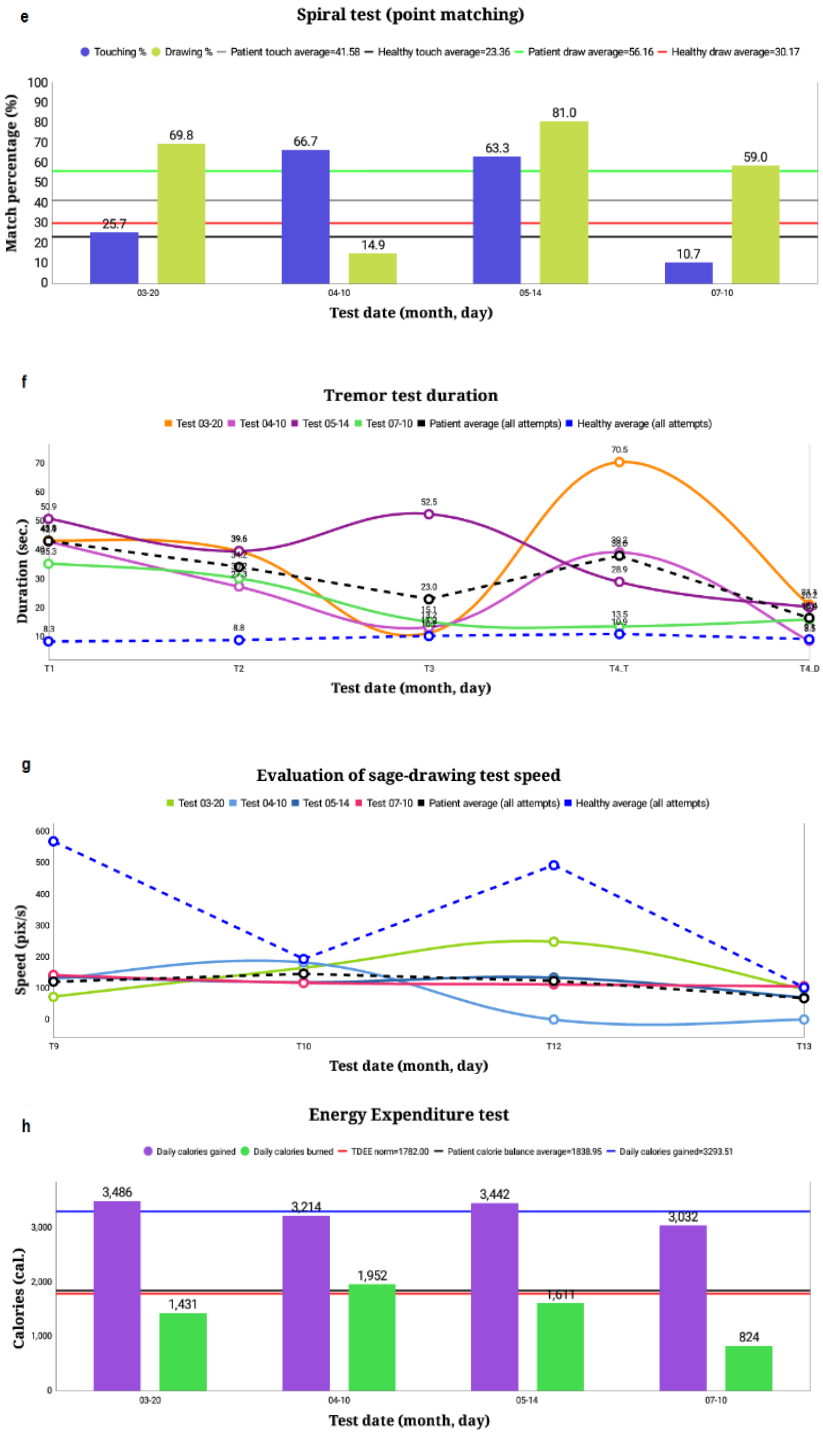


Figure B.2. Health state monitoring using graphical form for an individual patient serves as a practical application of the dissertation

C. Appendice. Complete list of the extracted features

Table C.1. Feature list with the names used for classifiers (each feature is separated by a comma)

Individual Classifier (Feature Count)	Feature List
T1: (9)	t1_delta_avg, t1_rt_avg, t1_duration, t1_num_outside, t1_num_inside, t1_std_delta, t1_std_rt, t1_avg_touch_area_ratio, t1_avg_scr_pressure.
T2: (10)	t2_delta_avg, t2_rt_avg, t2_duration, t2_num_outside, t2_num_inside, t2_std_delta, t2_std_rt, t2_avg_touch_area_ratio, t2_avg_scr_pressure t2_avg_visible_circles_touch.
T3: (28)	t3_avg_speed_all_pix_s, t3_avg_atan2_rad, t3_max_velocityX_pix_s, t3_max_velocityY_pix_s, t3_max_touch_area, t3_avg_touch_area_ratio, t3_avg_scr_pressure, t3_mgn_velocityX_div_by_max, t3_mgn_velocityY_div_by_max, t3_std_velocityX, t3_std_velocityY, t3_velocityX_change_rate, t3_velocityY_change_rate, t3_max_acc_X, t3_max_acc_Y, t3_num_direction_changes_X, t3_num_direction_changes_Y, t3_log_distance_all, t3_num_releases, t3_patient_path_size, t3_delta_avg, t3_rt_avg, t3_duration, t3_num_outside, t3_num_inside, t3_std_delta, t3_std_rt, t3_num_touch_stream.
T4 (spiral contour following): (22)	t4_touch_avg_speed_all_pix_s, t4_touch_avg_atan2_rad, t4_touch_max_velocityX_pix_s, t4_touch_max_velocityY_pix_s, t4_touch_max_touch_area, t4_touch_avg_touch_area_ratio, t4_touch_avg_scr_pressure, t4_touch_mgn_velocityX_div_by_max, t4_touch_mgn_velocityY_div_by_max, t4_touch_std_velocityX, t4_touch_std_velocityY, t4_touch_velocityX_change_rate, t4_touch_velocityY_change_rate, t4_touch_max_acc_X, t4_touch_max_acc_Y, t4_touch_num_direction_changes_X, t4_touch_num_direction_changes_Y, t4_touch_log_distance_all, t4_touch_frechet_distance, t4_touched_num_spiral_points, t4_touch_match_percentage, t4_touch_duration.
T4 (spiral contour drawing): (22)	t4_draw_avg_speed_all_pix_s, t4_draw_avg_atan2_rad, t4_draw_max_velocityX_pix_s, t4_draw_max_velocityY_pix_s, t4_draw_max_touch_area, t4_draw_avg_touch_area_ratio, t4_draw_avg_scr_pressure, t4_draw_mgn_velocityX_div_by_max, t4_draw_mgn_velocityY_div_by_max, t4_draw_std_velocityX, t4_draw_std_velocityY, t4_draw_velocityX_change_rate, t4_draw_velocityY_change_rate, t4_draw_max_acc_X, t4_draw_max_acc_Y, t4_draw_num_direction_changes_X, t4_draw_num_direction_changes_Y, t4_draw_log_distance_all, t4_draw_frechet_distance, t4_drawn_num_spiral_points, t4_drawn_match_percentage, t4_drawn_duration.
T9: (30)	t9_avg_speed_succ_pix_s, t9_avg_speed_all_pix_s, t9_avg_atan2_rad, t9_max_velocityX_pix_s, t9_max_velocityY_pix_s, t9_max_touch_area, t9_avg_touch_area_ratio, t9_avg_scr_pressure, t9_mgn_velocityX_div_by_max, t9_mgn_velocityY_div_by_max, t9_std_velocityX, t9_std_velocityY, t9_velocityX_change_rate, t9_velocityY_change_rate, t9_max_acc_X, t9_max_acc_Y, t9_num_direction_changes_X, t9_num_direction_changes_Y, t9_log_distance_all, t9_log_distance_succ, t9_num_releases, t9_patient_path_size, t9_graph_nm, t9_avg_angle, t9_err_single_node, t9_err_time_exceed, t9_err_notenough_nodes, t9_err_toomuch_nodes, t9_err_conn_exists, t9_num_reconstructions.
T10: (24)	t10_avg_speed_succ_pix_s, t10_avg_speed_all_pix_s, t10_avg_atan2_rad, t10_max_velocityX_pix_s, t10_max_velocityY_pix_s, t10_max_touch_area, t10_avg_touch_area_ratio, t10_avg_scr_pressure, t10_mgn_velocityX_div_by_max, t10_mgn_velocityY_div_by_max, t10_std_velocityX, t10_std_velocityY, t10_velocityX_change_rate, t10_velocityY_change_rate, t10_max_acc_X, t10_max_acc_Y, t10_num_direction_changes_X, t10_num_direction_changes_Y, t10_log_distance_all, t10_log_distance_succ, t10_num_releases, t10_patient_path_size, t10_err_clock_redraw, t10_err_contour_violation.

T11: (2)	t11_ jaro_closest_match_avg, t11_items_listed.
T12: (33)	t12_avg_speed_succ_pix_s, t12_avg_speed_all_pix_s, t12_avg_atan2_rad, t12_max_velocityX_pix_s, t12_max_velocityY_pix_s, t12_max_touch_area, t12_avg_touch_area_ratio, t12_avg_scr_pressure, t12_mgn_velocityX_div_by_max, t12_mgn_velocityY_div_by_max, t12_std_velocityX, t12_std_velocityY, t12_velocityX_change_rate, t12_velocityY_change_rate, t12_max_acc_X, t12_max_acc_Y, t12_num_direction_changes_X, t12_num_direction_changes_Y, t12_log_distance_all, t12_log_distance_succ, t12_num_releases, t12_patient_path_size, t12_graph_nm, t12_correct_conn_made, t12_err_toofew_nodes, t12_err_toomany_nodes, t12_err_binary, t12_err_conn_exists, t12_err_logical_conn, t12_task_reexecution, t12_succ_node_conn, t12_avg_Frechet_distance, t12_avg_angle_diff.
T13: (25)	t13_avg_speed_all_pix_s, t13_avg_atan2_rad, t13_max_velocityX_pix_s, t13_max_velocityY_pix_s, t13_max_touch_area, t13_avg_touch_area_ratio, t13_avg_scr_pressure, t13_mgn_velocityX_div_by_max, t13_mgn_velocityY_div_by_max, t13_std_velocityX, t13_std_velocityY, t13_velocityX_change_rate, t13_velocityY_change_rate, t13_max_acc_X, t13_max_acc_Y, t13_num_direction_changes_X, t13_num_direction_changes_Y, t13_log_distance_all, t13_num_releases, t13_patient_path_size, t13_graph_nm, t13_succ_operations, t13_err_same_node, t13_err_2nodes_needed, t13_err_op_violation.
T15: (4)	t15_tdee_calories, t15_patient_dailygained, t15_patient_dailylburned, t15_tdee_dailycaloriebalance.
Spelling (T7, T8, T11): (3)	t7_jaro_closest_match_avg, t8_jaro_closest_match_avg, t11_jaro_closest_match_avg.
SAGE: (10)	tsage1, tsage2, tsage3_4_5, tsage7, tsage8, tsage9, tsage10, tsage11, tsage6_12, Sage Total.
Duration: (16)	t1_duration, t2_duration, t3_duration, t4_touch_duration, t4_drawn_duration, t5_duration_s, t6_duration_s, t7_duration_s, t8_duration_s, t9_duration_s, t10_duration_s, t11_duration_s, t12_duration_s, t13_duration_s, t0_duration_s, t16_duration_s.

D. Appendice. Example of Weka *.arff* file design for data preparation

Figure D.1 shows how data is prepared for the classification model in *.arff* format. Such file contains three main parts: *relation* (name of the feature set), *attribute* (name of each feature, including data type) and *data* (actual values, separated by a delimiter). It can be directly loaded to Weka environment. The example from T1 task (9 features in total, 2 random data samples from healthy subjects).

```
@relation hd_t1
@attribute t1_delta_avg NUMERIC
@attribute t1_rt_avg NUMERICAspiration Disorders
@attribute t1_duration NUMERIC
@attribute t1_num_outside NUMERIC
@attribute t1_num_inside NUMERIC
@attribute t1_std_delta NUMERIC
@attribute t1_std_rt NUMERIC
@attribute t1_avg_touch_area_ratio NUMERIC
@attribute t1_avg_scr_pressure NUMERIC
@attribute IsSick {0,1}
@data
18.893,0.776,0,10,6.957,0.62,0.913,0.632,0
31.461,0.393,0,10,16.425,0.262,0.827,0.622,0
```

Figure D.1. Data pre-processing in Weka (T1 task)

E. Appendice. List of mobile apps related to tracking impairments for neurodegenerative disorders

Table E.1 provides a list of mobile apps for targeting neural impairments (mostly Parkinson Disease) as referenced in 2019 study [246]. + indicates that the listed mobile app is available on Google Play or Apple App Store.

Table E.1. List of existing mobile apps for neural impairment detection

App Name	Available in Android (Google Play)	Available in iOS (Apple App Store)
9zest Parkinson's Therapy	+	+
AppTUG Clinic	-	+
AppTUG Home	-	+
ARAT Action Research Arm Test	+	-
Aspiration Disorders	-	+
Be Bionic	+	-
Beats Medical	+	+
Berg Balance Scale	+	-
CloudUPDRS CUSSP	+	+

CNS - Finger Tapping Test	+	-
Conversation Paceboard	-	+
Curamobi	+	-
DAF Pro	+	+
Dysphagia	-	+
Dysphagia Therapy	+	+
Dysphemia SMART DAF	+	-
Dystonia Disease	+	-
Earlystimulus	+	-
EncephaLog Clinic	-	+
Encephalog Home	-	+
Essential Tremor	+	+
Fox Wearable Companion App	+	-
Gait 101	+	-
iSwallow	-	+
Lift Pulse	+	-
ListenMee APP-Human Bionics	+	-
LSVT LOUD	+	+
MedeCrush	+	-
mememtum	+	-
Mobile SLP Dysphagia	+	-
MoveME	+	-
Movement Disorders	+	-
My Parkinson's Disease Manager	-	+
MyDystonia	+	+
myHealthPal ®	-	+
myParkinson's	+	-
Neuro Health Storylines	+	+
OneRing - Artificial Intelligence for Parkinson's Disease	-	+
OPDM Mobility	-	+

Pacing Board Pocket Edition	-	+
Parkinson Home Exercises	+	+
Parkinson mPower study app	-	+
Parkinson Tremor Monitoring, Hand Shake Measuring	+	-
Parkinson's Central	+	-
Parkinson's Diary	+	+
Parkinson's Disease	-	+
Parkinson's Disease Facts	+	-
Parkinson's Disease Monitor & Commentary	-	+
Parkinson's LifeKit	-	+
Parkinson's Nil By Mouth	+	-
Parkinson's Toolkit	+	-
Parkinson’s Disease @Point of Care™	+	-
Parkinson’s Speech Aid	-	+
Parkinson´s Disease	+	-
Parkinsonism & Related Disorders	-	+
PD Me	-	+
PD Me Tools	-	+
PD on the GO	-	+
PD Warrior - Archealth	+	-
pd-FIT	-	+
PD101.me	+	-
PDmove	-	+
Progress Recorder	+	+
Shaky Hands	+	-
Simply DAF	-	+
Speak Up: An SPL Meter	+	-
Speech Pacesetter	-	+
SpeechAssess	+	+

SpeechCompanion Speech Therapy	+	-
SpeechVive	-	+
Steady Hand Test	+	-
StudyMyTremor	-	+
Swallow ID	-	+
Swallow Prompt	+	+
Swallow Rehapp	+	-
TapPD Physician	-	+
Tone Pacer For Droid	+	-
TR_Meter	+	-
tremor measurer	-	+
TREMOR12	-	+
TremorSense min	+	-
TremTracker	+	-
TremWatch(TM) Hand Tremor Test	+	-
uMotif	+	+
UPDRS	-	+
V-VST	-	+
Vital Tones Parkinson Pro	+	-
Vital Tones Tremor Pro	+	-
Vocal Pathology: Neurological	-	+
Voice Analyst	+	+
Voice Trainer	+	+

F. Appendice. SAGE methodology. Basic principles, scoring instructions

SAGE (Self-Administered Gerocognitive Examination), [78], [176], is a brief self-administered cognitive testing methodology to identify signs of mild cognitive impairment (MCI) and early dementia in, e.g., Alzheimer, Huntington patients. There are 13 questions (12 of them are evaluated by scores) and 4 different testing forms (to avoid cheating) in SAGE methodology. A patient can score maximum 22 points. A score of 17 and above is considered normal. SAGE was invented in Ohio State University Wexner Medical Center (USA). Lead SAGE researcher is a physician in neurology Douglas Scharre.

The example of predefined SAGE questions (Form 1) is provided in Figure F.1. Firstly, a non-scored item is presented to find out about general insights of a patient: family history, motor, stroke, depression symptoms, personality changes and functional abilities. The first question (1) instructs to name current date (month, date and year), the second (2) to name two listed pictures. The third question (3) asks to find similarities of two items, the fourth (4) to perform mathematical division, the fifth (5) to perform mathematical subtraction, the sixth (6) to memorize a given phrase. The seventh (7) SAGE question instructs to draw a cube on the right side of the paper, the eight (8) to draw analogue clock, the ninth (9) to write 12 items of given category, the tenth (10) to follow a path from start to end, the eleventh (11) to construct a geometrical shape (e.g., a square), the twelfth (12) to repeat the memorized phrase.

Name_____ Date of Birth _____/_____/_____

How far did you get in school? _____ I am a Man_____ Woman_____

I am Asian_____ Black_____ Hispanic_____ White_____ Other_____

Have you had any problems with memory or thinking? Yes_____ Only Occasionally_____ No_____

Have you had any blood relatives that have had problems with memory or thinking? Yes_____ No_____

Do you have balance problems? Yes_____ No_____

If yes, do you know the cause? Yes (specify reason)_____ No_____

Have you ever had a major stroke? Yes_____ No_____A minor or mini-stroke? Yes_____ No_____

Do you currently feel sad or depressed? Yes_____ Only Occasionally_____ No_____

Have you had any change in your personality? Yes (specify changes)_____ No_____

Do you have more difficulties doing everyday activities due to thinking problems? Yes_____No_____

1. What is today's date? (from memory – no cheating!) Month_____ Date_____ Year_____

2. Name the following pictures (don't worry about spelling):

Answer these questions:

3. How are a watch and a ruler similar? Write down how they are alike. They both are… what?

_____

4. How many nickels are in 60 cents? _____

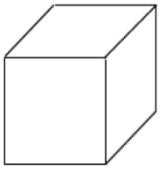
5. You are buying $13.45 of groceries. How much change would you receive back from a $20 bill?

_____

6. Memory Test (memorize these instructions). Do later only after completing this entire test:

At the bottom of the very last page: Write "I am done" on the blank line provided

7. Copy this picture:



8. Drawing test

- Draw a large face of a clock and place in the numbers
- Position the hands for 5 minutes after 11 o'clock
- On your clock, label "L" for the long hand and "S" for the short hand

9. Write down the names of 12 different animals (don't worry about spelling):

_____ _____ _____ _____

_____ _____ _____ _____

_____ _____ _____ _____

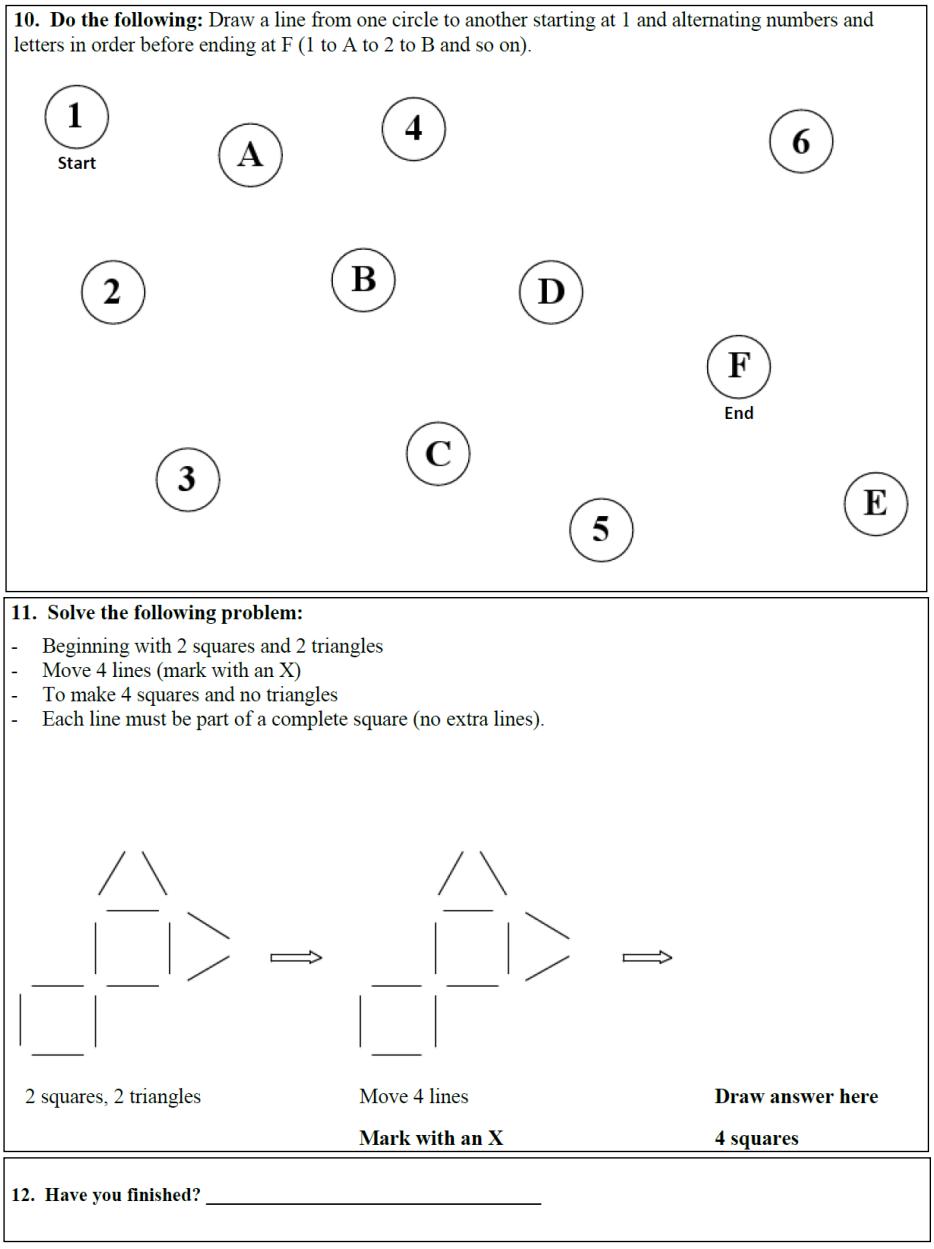


Figure F.1. Example of SAGE questionnaire (Form 1)

SAGE test is self-administered. It should be filled out without the assistance of others. The calendars and clocks should not be available during the testing. SAGE scoring instructions [260] are provided in Table F.1. A score is assigned to a patient (by supervised doctor) after the test completion. Total points = 0 (minimum) to 22 (maximum).

Table F.1. SAGE scoring instructions

Question no. (total possible points)	Evaluation criteria
1. Orientation: (4)	Month: 1 = Correct 0 = Incorrect Date: 2 = Exact 1 = ± 3 days 0 = All else Year: 1 = Correct 0 = Incorrect
2. Naming: (2)	Correct spelling not required. Each Picture: 1 = Correct 0 = Incorrect
3. Similarities: (2)	Correct spelling/grammar not required. 2 = Abstract 1 = Concrete 0 = All else
4. Calculation: (1)	Total possible point is 1. 1 = Correct 0 = Incorrect
5. Calculation: (1)	Total possible point is 1. 1 = Correct 0 = Incorrect
6, 12. Memory: (2)	Forms 1 and 2: 2 = Exact wording only, nothing extra: "I am done" 1 = Must contain the word "done": "Yes, I am done", "done", others 0 = All else Forms 3 and 4: 2 = Exact wording only, nothing extra: "I have finished" 1 = Must contain the word "finished": "Yes, I have finished", "I am finished", "finished", others 0 = All else
7. Construction: 3-D Figure: (2)	2 = 3-D, parallel lines within 10° and correct shape 1 = 3-D but lines not parallel within 10° or otherwise incorrect shape 0 = All else
8. Construction: Clock: (2)	4 components: clock face, clock numbers (all 12 numbers in correct order clockwise and approximately correct quadrant position), hand positions (hands to correct time and must be joined near the clock centre), and hand size (actual or if labelled correctly). 2 = 4 of 4 components correct 1 = 3 of 4 components correct; one of the three correct components must be hand positions 0 = All else

9. Verbal Fluency: (2)	Correct spelling not required. 2 = 12 different items listed 1 = 10 or 11 different items listed 0 = 9 or less different items listed
10. Executive: Modified Trials (2)	An error is if two items that should be connected are not or if two items that should not be connected are. 2 = Perfect or self-corrected errors only 1 = 1 or 2 errors 0 = More than 2 errors
11. Executive: Problem Solving (2)	Forms 1 and 2: 2 = Correct lines moved or marked and final diagram correct 1 = Correct lines moved or marked and no final diagram drawn Or Correct lines moved or marked but final diagram incorrect Or No lines moved or marked and final diagram correct 0 = All else including lines moved or marked incorrectly but final diagram correct Forms 3 and 4: 2 = Correct lines crossed out and final diagram correct 1 = Correct lines crossed out and no final diagram drawn Or Correct lines crossed out but final diagram Incorrect Or No lines crossed out and final diagram correct 0 = All else including lines crossed out incorrectly but final diagram correct

G. Appendice. Comparison of author's work with other researchers' scientific findings

In this appendice, a comparative pilot study of the author's dissertation and works of other researchers is provided. Table G. illustrates a comparison of disease identification (e.g., Parkinson, Huntington, Alzheimer) research based on speech impairments. The results show that the author proposed a method WST (Wavelet Scattering Transform) with SVM classifier (100%) for speech impairment tracking to overcome other researchers. However, for a fully adequate comparison, a unanimous dataset should be considered, because the listed authors in Table G. target different feature extraction methods, voice recording format and sample collection environments. In addition, the majority of related work of researchers for speech impairment detection adapts their own custom dataset.

Table G.1. Comparison of speech impairment detection scientific findings (author and other researchers)

Author's name (ref. no)	Classifier	Achieved accuracy (%)
Tsanas A. [94]	SVM	90.00
Caesarendra W. [96]	SVM	79.17
Hauptman Y. [216]	SVM	77.20
Wu, Kebin. [217]	Random Forest	96.37
Oung, Qi Wei. [220]	Extreme learning machine	91.11
Ali, Humair [221]	Random Forest	97.60
Jeancolas, Laetitia [224]	Bootstrap aggregation	83.00
Author of the dissertation	RNN + BiLSTM WST + SVM	86.63 100.00

SL344. 2020-02-05, 20 leidyb. apsk. l. Tiražas 14 egz.
Išleido Kauno technologijos universitetas, K. Donelaičio g. 73, 44249 Kaunas
Spausdino leidyklos „Technologija" spaustuvė, Studentų g. 54, 51424 Kaunas