



Automated stress identification and speaker diarisation for speech-language pathology

KA Wimmer

 [orcid.org/ 0000-0001-9113-5506](https://orcid.org/0000-0001-9113-5506)

Dissertation accepted in fulfilment of the requirements for the degree *Master of Engineering in Computer and Electronic Engineering* at the North-West University

Supervisor: Prof H Marais

Graduation: August 2024

Abstract

This study presents an engineering-based approach to assist speech-language pathologists (SLPs) in their analysis of parent-child speech interactions. Currently, SLPs spend numerous hours evaluating video recordings of contact sessions to identify each speaker's specific speech and associated behaviours. This study offers an autonomous solution that utilises ECG and accelerometer sensors to identify stress and movement, respectively, and speaker diarisation to identify who spoke when. The algorithms were trained and evaluated using open-source datasets such as CommonVoice [1], LibriSpeech [2], MIT-BIH [3], WESAD [4], and HAPT [5]. Obtaining respectable results with a diarisation error rate (DER) of 5.207% for speaker diarisation, a validation accuracy of 93.939% (32-true positive, 2-false positive, 8-false negative, 165-total size) for stress identification with a beat detection error rate of 0.877% for normal beats, and a validation accuracy of 99.038% (6355-true positive, 76-false positive, 45-false negative, 12575-total size) for movement identification. These algorithms are integrated into a single analyser application that provides audio and visual results to the SLP. Additionally, a second recorder application was created to capture a single-channel audio recording and multiple ECG and accelerometer signals from patients wearing Polar H10 devices. These two applications automate the previous manual analysis procedures for speech, stress, and movement assessments. The study's findings offer a novel, automated tool for speech interaction analyses that can potentially benefit SLPs in their clinical assessments.

Keywords: Automation; Speech-language pathology; Speaker diarisation; Behavioural analyses.

Contents

1	Introduction	1
1.1	Background and Rationale	1
1.2	Problem Statement	4
1.3	Aim and Objectives	4
1.3.1	Aim	4
1.3.2	Primary Objectives	4
1.3.3	Secondary Objectives	5
1.4	Scope and Limitations	5
1.5	Ethical Considerations	6
1.6	The layout of the study	6
2	Literature	8
2.1	Introduction	8
2.2	Existing tools	9
2.3	Filling the gaps: Speech analysis	11
2.3.1	Speaker diarisation systems	12
2.3.2	Features	16
2.3.3	Evaluations and Datasets	17
2.3.4	Obstacles	20
2.4	Filling the gaps: Behaviour analysis	21
2.4.1	Anxiety	21
2.4.2	Heart rate variability	23
2.4.3	HRV: Related Factors	26
2.4.4	Heart monitors and other physiological sensors	27
2.5	Filling the gaps: Stress Identification	29
2.5.1	Pre-processing	29
2.5.2	QRS-detection algorithms:	32
2.5.3	Feature extraction	33
2.5.4	Classification	33
2.5.5	Evaluations and datasets	34
2.5.6	Correlation tools	37
2.6	Filling in the gaps: Movement Identification	38
2.6.1	Pre-processing	38

2.6.2	Feature extraction	39
2.6.3	Classification	39
2.6.4	Evaluation and datasets	40
2.7	Summary	41
3	System Design and Implementation	43
3.1	System Architecture	43
3.2	Recorder: Data Collector	46
3.2.1	Audio	46
3.2.2	Behavioural	46
3.3	Analyser: Speaker Diarisation	48
3.3.1	Pre-processing	50
3.3.2	Feature Extraction	51
3.3.3	Classification	52
3.3.4	Measurements	57
3.4	Analyser: Stress Identification	57
3.4.1	Pre-processing	58
3.4.2	Feature Extraction	61
3.4.3	Classification	63
3.5	Analyser: Movement identification	66
3.5.1	Pre-processing	66
3.5.2	Feature Extraction	67
3.5.3	Classification	68
3.5.4	Dataset Conversions	69
3.6	User Interface	70
3.6.1	Recorder	70
3.6.2	Analyser	71
3.7	Software and Tools	74
3.7.1	Software	74
3.7.2	Hardware	75
3.8	Integration	75
3.8.1	UI Integration	75
3.8.2	Audio	76
3.8.3	Behavioural	76
3.9	Scalability and Efficiency	77
3.10	Challenges Faced	78
4	Evaluation	79
4.1	Evaluation Metrics	79
4.1.1	Speaker Diarisation	80
4.1.2	Behavioural	80
4.2	Dataset Description	81
4.2.1	Speaker Diarisation	81
4.2.2	Beat Detection	82

4.2.3	Stress Identification	83
4.2.4	Movement Identification	83
4.3	Experimental Setup	84
4.4	Results and Analyses	85
4.4.1	Verification: Data Capturing	85
4.4.2	Verification: Speaker Diarisation	86
4.4.3	Verification: Beat Detection	90
4.4.4	Verification: Stress Identification	92
4.4.5	Verification: Movement Identification	94
4.5	Conclusion	96
4.5.1	Validation: Speaker Diarisation	96
4.5.2	Validation: Beat Detection	96
4.5.3	Validation: Stress Identification	96
4.5.4	Validation: Movement Identification	97
4.5.5	Validation: Applications	97
4.5.6	Validation: Overall	97
5	Conclusion	98
5.1	Summary and Results Overview	98
5.2	Limitations	99
5.2.1	Speaker Diarisation System	99
5.2.2	Behavioural Algorithms	100
5.2.3	Applications	100
5.3	Future Improvements	100
5.3.1	Speaker Diarisation System	101
5.3.2	Behavioural Algorithms	101
5.3.3	Applications	102
5.4	Possible Benefits	102
A	Polar H10 PMD GATT	103
A.1	PMD services	103
A.2	PMD commands	105
B	Viterbi Algorithm (Log-interpretation)	108
C	Algorithm	110
D	WESAD PKL-To-CSV Python Script	113
E	MIT-BIH-to-MATLAB Code	115
F	HAPT-to-MATLAB Code	117
G	Master Code	120

List of Figures

1.1	Projected shortage of audiologists and speech-language pathologists in South Africa till 2030.	3
2.1	LENA device and software package.	10
2.2	Traditional speaker diarisation systems.	13
2.3	MFCC creation.	17
2.4	Diagram of the SNS and PNS systems, indicating the functions of each in the human body.	22
2.5	Detailed (single-lead) ECG signal.	23
2.6	Einthoven triangle.	28
2.7	ECG noise types.	30
2.8	Variability in HRV data.	30
2.9	Limited wavelet list. (MATLAB self-generated image)	31
2.10	SWT decomposition.	32
2.11	RR-interval reduction during periods of stress. (Kubios self-generated image)	34
2.12	Kubios Overview. (Kubios self-generated image)	38
2.13	Accelerometer body movement illustration.	40
3.1	System architecture	44
3.2	System design outline - Recorder	44
3.3	System design outline - Speech analysis (Analyser)	45
3.4	System design outline - Behavioural analysis (Analyser)	45
3.5	Behaviour data collection procedure.	47
3.6	PMD data frame: ECG	48
3.7	PMD data frame: Acceleration	48
3.8	Speaker diarisation system - Flow diagram.	49
3.9	Speaker diarisation output example	49
3.10	Speech activity detection flow diagram.	52
3.11	Model-based approach - string HMM illustration.	53
3.12	Schematic view of the initial segmentation process	56
3.13	SWT noise reduction - Time-domain.	59
3.14	SWT noise reduction - Frequency-domain.	59
3.15	Illustration of detrending the HRV data.	61
3.16	ANOVA stress feature ranking.	62

3.17	Beat detection flow diagram.	63
3.18	Cubic SVM: Confusion Matrix	65
3.19	Cubic SVM: ROC Curve	65
3.20	Accelerometer artefact corrected.	67
3.21	Illustration of the ‘Filtered Accelerometer Signal’, ‘Body Movement Signal’, and ‘Gravity Signal’ all of the x-axis of the triaxial accelerometer sensor. . .	68
3.22	Movement identification - Feature illustration	69
3.23	Recorder application with two connected Polar H10 devices.	70
3.24	Recorder preview.	71
3.25	Example of a saved session.	71
3.26	Analyser - Main page.	72
3.27	Analyser - Patient Results	73
3.28	Analyser - Speaker diarisation results.	73
3.29	Analyser - Stress identification results.	73
3.30	Analyser - Movement identification results.	74
3.31	Behavioural data format.	77
4.1	Evaluation - Missed speech annotations.	82
4.2	Sensor placement for data capturing evaluation.	84
4.3	Data capturing evaluation - Real-time changes.	86
4.4	Data capturing evaluation - Heart rate trend.	86
4.5	Speaker diarisation output verification.	87
4.6	Incorrect speaker clustering result - Parallel execution entry 3	88
4.7	Time analysis for speaker diarisation execution.	89
4.8	Beat detection erros. MIT-BIH - 105m	90
4.9	ECG beat detection evaluation - Polar H10	92
4.10	Stress identification - Test confusion matrix	92
4.11	Stress identification - Test ROC	93
4.12	Stress identification - Kubios	93
4.13	Stress identification - Algorithm	94
4.14	Movement identification - Confusion matrices	94
4.15	Movement identification - ROCs	95
4.16	Movement measurement - Evaluation	95
A.1	PMD prerequisites.	105
A.2	PMD read available features.	106
A.3	PMD request available setting types.	106
A.4	PMD start streaming.	107
A.5	PMD stop streaming.	107

List of Tables

2.1	Top 12 long-term features for speaker diarisation.	17
2.2	Commonly used dataset for speaker diarisation comparisons.	18
2.3	Summary of the main HRV variables in psychophysiological research and their origins.	24
2.4	Normal PR interval as it correlates to age and heart rate (and upper limit of normal).	27
2.5	Typical features utilised in stress identification studies.	33
2.6	Common machine-learning models for stress identification.	34
2.7	Common variables utilised as human activity recognition features.	40
3.1	Speaker diarisation long-term features	52
3.2	ECG SWT decomposition	58
3.3	Summary of the HRV features used and their correlation to stress.	62
3.4	Stress Identification models comparison.	64
3.5	Movement identification: Classifier Comparison.	69
3.6	MATLAB packages utilised.	74
3.7	Hardware architecture of the host computer utilised for evaluation.	75
3.8	Audio protocol	76
3.9	Behavioural protocol	76
4.1	Speaker diarisation evaluation metrics	80
4.2	Beat detection - Evaluated recordings	83
4.3	Speaker diarisation evaluation results - Single-threaded execution	87
4.4	Speaker diarisation evaluation results - Parallel (3-cores) execution	88
4.5	Speaker diarisation measurement results - Ground Truth	89
4.6	Speaker diarisation measurement results - Hypothesised	90
4.7	Speaker diarisation measurement accuracies.	90
4.8	Beat detection evaluation results.	91
4.9	Beat detection evaluation results - Polar H10 ECG measurement	92
4.10	Beat detection measurement results.	92
4.11	Stress identification - Evaluation metrics results.	93
4.12	Stress identification measurement results - Stress observations	93
4.13	Movement identification - Evaluation metrics results	94
4.14	Movement identification measurement results - Movement observations	94

4.15 Movement identification measurement results - Maximum and average ac-	
celeration	95
A.1 PMD GATT Service and its Characteristics	103
A.2 PMD measurement types	104
A.3 ECG frame type	104
A.4 Acceleration frame type	104
A.5 PMD setting types	104

List of Abbreviations

ACC	Accelerometer/Accelaration
AHC	Agglomerative Hierarchical Clustering
ANS	Autonomic Nervous System
AR	Autoregressive
ASD	Autism Spectrum Disorder
BIC	Bayesian Information Criteria
BLE	Bluetooth Low Energy
bpm	Beats per Minute
CWT	Continuous Wavelet Transform
DER	Diarisation/Detection Error Rate
DFT	Discrete Fourier Transform
DNN	Deep Neural Network
DWT	Discrete Wavelet Transform
ECG	Electrocardiogram
EEND	End-to-End Neural Diarisation
FFT	Fast Fourier Transform
GATT	Generic Attribute
GMM	Gaussian Mixture Model
HAPT	Human Activities and Postural Transitions
HAR	Human Activity Recognition
HF	High Frequencies
HMM	Hidden Markov Models
HNH	Harmonic-to-Noise Ratio
HPA	Hypothalamic-Pituitary-Adrenal
HR	Heart Rate
HRV	Heart Rate Variability
Hz	Hertz
IBI	Inter-beat Intervals
JER	Jaccard Error Rate
LF	Low Frequencies
LPC	Linear Predictive Coding
MFCC	Mel Frequency Cepstral Coefficient
MLP	Multilayer Perceptron
ms	Milliseconds
mV	Millivolts

NN-intervals	Normal-to-normal intervals
PDF	Probability Density Function
PLDA	Probabilistic Linear Discriminant Analysis
PMD	Polar Measurement Data
pNN50	Percentage of Normal Sinus NN-intervals more than 50 ms
PNS	Parasympathetic Nervous System
QRS-complex	QRS-region within an ECG signal
RMSSD	Root Mean Square of Successive Differences
ROC	Receiver Operating Characteristics
SD	Standard Deviation
SDK	Software Development Kit
SDNN	Standard Deviation of the NN interval
SLD	Specific Language Disorder
SLP	Speech-language pathologist
SNS	Sympathetic Nervous System
SVM	Support Vector Machine
SWT	Stationary Wavelet Transform
UBM	Universal Background Model
UI	User Interface
ULF	Ultra-low Frequencies
VAD/SAD	Voice/Speech Activity Detection
VLF	Very-low Frequencies
WESAD	Wearable Stress and Affect Detection

Chapter 1

Introduction

This study outlines a Master of Engineering (MEng) study that introduces the creation of an automated speech interaction analysis system. The system uses speech-language pathology (SLP) as a contextual background for which the system is developed.

This chapter introduces the study, presenting the problems and motivations that led to its creation; it outlines its objectives, scope, limitations, and layout. Working towards this, Section 1.1 introduces the necessity of the subject and identifies the challenges. Section 1.2 defines the problem statement, and Section 1.3 outlines the study's goals. Thereafter, Section 1.4 establishes the study's scope and limitations; Section 1.5 the ethical considerations thereof; and finally Section 1.6 provides an overview of the dissertation's remaining structure.

1.1 Background and Rationale

Children are the future

Language and communication-interaction development are both indicators of and key mechanisms in cognitive development, psycho-social health, and academic performance and, therefore, are vital components of early childhood development [6, 7]. Despite the critical role of healthy reciprocal language interactions in health and well-being, this aspect does not often receive adequate attention in child healthcare and early development [8]. In a study done over nearly an entire decade, Hart and Risley [8] found, with a few exceptions, that the more parents communicated with their children, the faster the child's vocabulary growth and the higher the child's intelligence quotient (IQ) test score would be at age three and later [8]. By age three, the differences in a child's language experience were significant: some children had heard nearly 33 million words, while others had as few as 9 million in their lifetime. Furthermore, it is stated that the intellectual gap between children at this age remained prominent for the remainder of their lives [8]. While various factors might impact a child's rate of vocabulary growth, vocabulary use, and IQ score by the age of three, the authors in [8] found that the number of words a parent communicates per hour had a more significant impact than the parent's education or socioeconomic sta-

tus. Leading to the question of what impacts a parent-toddler communication interaction besides (not) speaking.

Parent-toddler communication interactions

In a study done by Fields-Olivieri et al. [9] found that the emotional expressions and temperaments of both parents and children can affect parent-toddler communication interactions [9]. The research suggests that when toddlers display positive emotions, it encourages parents to initiate conversations with their children, leading to more communication attempts and more time spent interacting with the child. On the other hand, negative emotions such as sadness and fear can disrupt parent-toddler communication. However, anxiety is a more significant contributor to disrupting or prohibiting communication interactions between parents and children. When children feel anxious, it can be challenging to communicate with them effectively. This is because anxiety is the body's way of telling us that it finds the environment or change in the environment unpleasant and responds to it in one of two ways: fight or flight [10]. With ill-knowledge and technique, parents can inadvertently cause anxiety in their children when trying to communicate. To ensure proper communication and promote a child's development, parents may consider consulting an SLP.

Speech-language pathologist's settings

Knowing the importance of early development, parents seek the help of a professional, for instance, an SLP, commonly referred to as a speech therapist. A typical contact session with an SLP would start with the parents scheduling a contact session with the SLP. The session is set for a specified time and date, which the parents and the child will attend. The session takes place in private at the SLP's office. Games or tasks are laid out within the office to allow the child to choose which tasks they want to participate in and in what order. The SLP will explain the rules and guide all participants through the tasks. The SLP will set the example of proper communication and allow the parents to replicate it. The tasks are designed in such a way as to enable all participants to act and communicate freely, allowing the SLP to get a sense of the family's daily interactions with one another and establish personalities. At the end of the session, the SLP will give feedback to the parents, provide them with tools to improve their communication interactions, and, if needed, make another booking to see the family again. The process repeats as long as the SLP sees fit; this ranges from a month up to a year or sometimes longer.

Speech-language pathologist's analyses

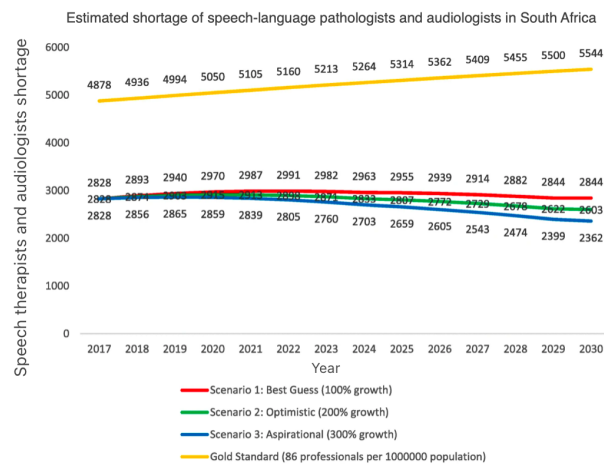
During speech therapy sessions, SLPs typically evaluate an individual's communication skills and personality traits, while also investigating underlying communication impairments such as developmental disabilities (e.g., autism spectrum disorder, attention-deficit disorder, specific language impairment), psychiatric disorders (e.g., schizophrenia), and respiratory patterns and compromises (e.g., bronchopulmonary dysplasia) [11]. They use both visual and audio recordings of these sessions to allow for further assessment afterwards.

When analysing these recordings, the SLP determines the time frames of each person’s speech (mimicking an algorithm called speaker diarisation to identify who spoke when), extracting metrics such as turn-taking, phrase length, and conversation length [12]. Additionally, the SLP relies on visual cues to interpret the child’s behaviour throughout the session.

While these analyses are essential for accurate diagnostics and effective treatment planning, they can be time-consuming, which limits the number of patients the SLP can assist at any given time.

SLP Statistics

The need for SLPs and audiologists is increasing in South Africa due to its growing population and burden of illness and disease. A recent statistical analysis conducted by Pillay et al. [13] established a gold standard of 86 audiologists and SLPs per million population based on the service delivery guidelines of the Health Professions Council of South Africa (HPCSA) [14]. According to this standard, there was a shortage of approximately 4800 professionals in 2017, which is expected to increase to 5500 by 2030 [14]. Figure 1.1 illustrates the projected shortage of audiology and speech-language pathology professionals in South Africa by 2030.



This figure shows the shortfall in the number of speech therapists and audiologists in four projection scenarios – best guess, optimistic, aspirational and gold standard.

Figure 1.1: Projected shortage of audiologists and speech-language pathologists in South Africa till 2030 [14].

1.2 Problem Statement

The demand for SLPs is increasing with a rapidly growing population and socially aware parents. More than ever, parents send their children to speech-language therapy to ensure they do not fall behind. However, due to a shortage of SLPs and the substantial time needed to conduct parent-toddler speech interaction analyses, there are limitations to the number of families that can be helped.

Since the interaction analyses appear to be algorithmic **the question remains whether parent-toddler speech interaction analyses can be automated.**

1.3 Aim and Objectives

1.3.1 Aim

Automate the analysis process that an SLP follows when conducting speech-interaction analyses.

According to the Oxford Dictionary of English [15]:

- **Automate** - convert (a process or facility) to be operated by largely automatic equipment.
- **Analysis** - a detailed examination of the elements or structure of something.
- **Process** - series of actions or steps taken to achieve a particular end.

This translates to ‘convert the detailed examination steps an SLP follows when conducting speech-interaction analyses (detailed examinations) to be operated largely by automatic equipment’.

Speech-interaction analyses are assessments of a person’s speech and behaviour in response to certain interactions or activities. The goal of this study is to create a system that can perform the preparation work required for conducting speech-interaction analyses, making the process less time-consuming for SLPs.

1.3.2 Primary Objectives

This study focuses on lightening the workload of SLPs by:

1. Automatically identifying person-specific speech regions; and
2. Automatically identifying person-specific behavioural regions - i.e. stress and movement.

To accomplish this the following three aspects must be addressed:

1. Speaker diarisation;
2. Stress (anxiety) identification;
3. Movement identification.

1.3.3 Secondary Objectives

This study also sets out an ease-to-use means of fulfilling its primary objectives by providing:

1. An interactive UI; and
2. A data capturing and synchronisation application.

1.4 Scope and Limitations

This study focuses on the automation of an SLP's parent-toddler speech interaction analysis process. It aims to provide SLPs with a simple means of collecting speech and behavioural data during a contact session. As well as analysing the collected data and presenting it in a diarised format.

Contact sessions are conducted within an SLP's office and possess a non-restrictive approach. This means that the patients are free to move and interact within the office. As a result, sudden postural changes and movements are highly possible, especially when working with children. The patients are also not restricted in their speech, which allows for mixed-language phrases, interruptions from various speakers (overlapping speech), and unknown speaker linguistic variants. It is also assumed that the patients will speak in either English or Afrikaans. Addressing overlap speech is one of the most difficult challenges within the speech processing domain, and is placed outside of the scope of this study.

The study assumes that there are no prior recordings of the patients, which means that each individual's speech signature would be unknown. Additionally, the number of speakers present and their orientations would be unknown.

The analyses of the contact sessions will be performed offline, which means that real-time processing is not required. However, the processing of the analyses would have to be fast enough to allow for the processing of at least five one-hour contact session analyses per day. The execution time is not as strict because, while the analysis is being performed, the SLP can conduct additional contact sessions or attend to other tasks.

To develop and evaluate the speech-interaction analysis tool, data collection will be necessary. The data can be collected either in the form of open-source datasets or direct data recording from human participants. Ethical approval must first be obtained when collect-

ing data from human participants. Open-source datasets would prevent this requirement and allow the development of an initial analysis tool which can be further fine-tuned later with the addition of data collected in practice.

It is important to note that the study is not intended to replace the SLPs or their diagnoses, but rather to support their analysis process and to extend the possible analyses to be conducted simultaneously. This includes speech analyses overlaid with behavioural analyses.

By no means should it be assumed that the entirety of the algorithms used within this study are self-developed. MATLAB toolboxes are used openly to accomplish low-level functions. Where this is not possible the algorithms are either altered or self-developed.

This study was conducted between 2021 and 2023.

1.5 Ethical Considerations

This study has no ethical risks. Ethics number: NWU-00463-24-S3.

1.6 The layout of the study

The rest of this study is divided into four main chapters on automating speech-interaction analyses within speech-language pathology. A brief description of what is to be expected in each chapter follows.

Chapter 2 introduces the appropriate literature regarding the proposed problem: how to automate speech-interaction analyses? It provides an **overview of speech-interaction analyses** within the domain of speech-language pathology and **existing tools and techniques** readily available to the SLP. It then addresses their shortcomings and means to fill the gap. It looks at means of automating **speech analyses** through **speaker diarisation systems**, providing common structures, features, datasets, evaluations, and obstacles typically faced by speaker diarisation systems. It also addresses **behavioural analyses** in the form of anxiety. To provide automation in this regard, **stress and movement identification algorithms** are assessed by establishing, among others, common pre-processing, feature extraction, classification, and evaluation techniques, as well as commonly used datasets.

Chapter 3 describes the design and implementation of the analysis and data recording tool. In doing so, it depicts the architecture of the automation system and the creation of each subsystem, which includes speaker diarisation, stress identification, movement identification, and the data recording tool. Additionally, it provides a detailed description of the user interface created for these systems, their integration, and the software and tools required to make it all happen. Lastly, it also covers the scalability and efficiency of the solution, and the challenges encountered.

Chapter 4 describes the validation and verification of this study, providing details of the metrics, datasets, and experimental setup used to evaluate each subsystem. Thereafter, the results and analyses are provided, fulfilling the verification of each subsystem. Lastly, validation of the subsystems, algorithms, and overall design of this study is provided in the conclusion section of this chapter.

Chapter 5 concludes this study with a brief review of the initial objectives, results achieved, and future directions.

Chapter 2

Literature review

2.1 Introduction

This chapter addresses the existing tools and techniques used by SLPs to conduct speech interaction analyses, their shortcomings, and how they can be mended.

According to the American Speech-Language-Hearing Association (ASHA), ~~a speech-language pathologist~~ an SLP is defined as a professional who engages in professional practice in the domains of communication and swallowing across the life span [11]. Communication includes language, speech production and fluency, cognition, voice, resonance, and hearing. Swallowing includes all aspects of swallowing, including related feeding behaviours. Throughout this study, the focus will be placed on verbal communication, i.e. speech.

This study focuses on the speech interactions that take place between parents and children in an SLP's office. The SLP evaluates the communication abilities of each individual and often uses techniques like language sampling [12] (see below) to obtain metrics that help measure the communication capabilities of both the child and the parent.

- **Language Sampling** - These techniques are used to elicit spontaneous language in various communication settings such as free play, conversation/dialogue, narration, and expository speech. It is furthermore, used to derive measures such as mean length of utterance (MLU)¹ and type-token ratio (TTR)², to complement data obtained from standardised language assessments [12].

SLPs often conduct post-interdisciplinary assessments. This entails the assessment of an audiovisual recorded contact session where various specialists (e.g., neurologists, psychologists, and psychiatrists) assess the communication capabilities of the patient [11]. Each specialist goes through the entire recording and annotates specific characteristics that may indicate a speech difference or disorder. This process can be quite tedious as it involves

¹MLU - is the average length of sentences an individual typically uses. A higher MLU indicates a higher level of language proficiency.

²TTR - the number of unique words (types) divided by the total number of words in the sample (tokens).

watching and re-watching hours of footage, assessing it and discussing the results with the other specialists.

It is through the above-mentioned techniques that the world of speech-language pathology opens to automation. Knowing when the child speaks and having context to what was asked and answered, algorithms can be established to automate a vital part of speech-language pathology.

2.2 Existing tools

E-prime is a stimulus presentation software used for behavioural and neuroimaging research [16]. E-prime allows physicians and researchers, alike, to design, collect, and analyse their experiments. It uses an object-oriented programming language that allows researchers to easily design experiments. Stimuli: visual (e.g., words/text, images, and videos) and audio (e.g., hearing, and speaking). It also allows for response recordings for example in the form of button presses. It also allows for the capturing of electroencephalogram (EEG), functional near-infrared spectroscopy (fNIRS), and physiological measurements via third-party devices. It has a minimum initial cost of \$495 and a yearly renewal fee of \$125. It is created to aid in the design of any experiment in which a computer serves as an interface between the subject (child) and the experimenter (SLP) [16]. This approach, however, does not allow for analyses in social environments, such as parent-child speech interactions which is the aim of this study.

SALT (Systemic Analysis of Language Transcripts) is a software program used for eliciting, transcribing, and analysing language samples [17]. It includes a transcription editor, standard reports, and reference databases for comparisons. With the use of SALT, SLPs can track the progress of children with intellectual and developmental disabilities (IDD) typically via their vocabulary development. SALT allows a simple means for conducting language samples. The first step SLPs follow is to choose a sample type (elicitation material) and to become familiar with its review protocol. Then, they would typically record their language sample with their patients. The software allows for any English or Spanish sample, however, if one of SALT's referenced samples is chosen such as the wordless storybook: 'Frog, Where Are You?' (Mayer, 1969) SLPs have the option to compare their results with other typical speakers in SALT's referenced database. Thirdly, the SLP would then use the SALT editor to transcribe their sample following SALT's transcription conventions. The last step is to analyse the data. SALT provides standard measures and performance reports that provide analysis outcomes across all expressive language domains such as intelligibility; syntax and morphology; semantics; discourse; verbal facility; and omissions and errors.

SALT is less expensive than E-prime, with a price tag of \$209 for a single clinical license. SALT, however, allows SLPs to analyse patients' language skills in a natural environment such as storytelling, conversations, and interactions with multiple speakers. The disadvantage thereof, however, is that it still requires the SLP to listen to the entire recording to

find the words elicited by the patient. It also provides no information on the behavioural state of the patient.

LENA (Language Environment Analysis) (see Figure 2.1) is a hardware device and software package designed to improve early language development [18]. The technology is designed with the intention of measuring a child’s verbal communication exposure in all settings (e.g., the SLP’s office; the patient’s home away from the SLP). It offers speech-language professionals the ability to capture the natural language environment of the child, as well as to monitor the constancy of the parents following the treatment (e.g., conversation rates and television exposures) and provides progress reports of the child’s vocalisation. It furthermore provides SLPs graphical representation of adult word counts, conversation turns, child vocalisations, and amount of TV exposure.

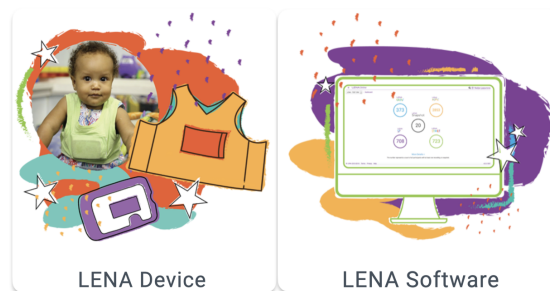


Figure 2.1: LENA device and software package [19].

One of the limitations of this technology is that it only distinguishes the speech between two entities, namely the wearer and the person speaking to them. It does not provide any behavioural information. Additionally, once the audio recording has been transmitted for analysis, it is permanently deleted as a precautionary measure. This inhibits the SLP from conducting other analyses, such as language sampling. As of now, a single recording device costs \$329.00.

There are also **computer-based games** that SLPs could use to encourage child vocalisation such as ‘TalkTime with Tucker’ [20]. The game is based on the education levels of pre-kindergarten to grade 4 and is for children who are reluctant to speak (autistic; intellectual and developmental disabilities; physical disabilities; and language-learning disorders). It was found that children respond better to animated characters, and this is where this game comes in. The game uses an animated character, Tucker, and voice-activated controls to encourage children to vocalise by leading him through his adventures. This aids SLPs when trying to analyse the speech of reluctant talkers. It, however, does not allow for language samples, nor an analysis of the child’s communication interactions in a natural environment.

While re-accessing this report, Nuance and Microsoft announced an AI-powered clinical documentation software, ‘**Dragon Ambient eXperience (DAX) Express**’ [21]. It is a workflow-integrated, fully automatic clinical documentation application that combines proven conversation and ambient AI with OpenAI’s natural language processing tool ChatGPT-4. Within the domain of speech-language pathology, this could provide SLPs

with a fully automated transcript of their contact sessions with their patients, allowing for faster language sample analyses (LSAs), speaker-specific time frames, and possibly faster speech interaction analyses. The software is, however, still unavailable commercially and does not provide behavioural information.

Currently, SLPs have a vast collection of tools and techniques at their disposal. Ranging from standardised tests to aided language sample programs and from computer-based games to conversation-assessing devices. However, most are tailored toward speech intelligibility and less to speech production. Meaning it neglects behavioural factors that might impact an individual's speech involvement.

E-Prime, on the other hand, is one of the most advanced software tools designed for behavioural research, however, it is meant for psychologists and neglects speech interaction analysis conducted by SLPs. Furthermore, the settings wherein this tool is meant to be used are within restrictive environments such as controlled laboratories and clinics. This makes this approach, however efficient in behavioural analysis, unequipped for children in a multi-speaker communication interaction.

None of these technologies, however, assist in speech interaction analyses. Such as assisting SLPs in obtaining person-specific segments in audio-recorded data or allowing them to have faster access to data vital for their analyses. Such as behavioural information (e.g., anxiety or stress) which also affects the communication interactions between individuals, especially in parent-toddler communication interactions [9]. This drives the need for an easy-to-use analysis tool for SLPs who have to analyse speech interactions between a child and their parents.

2.3 Filling the gaps: Speech analysis

Given the significant time expended and human resources required for speech interaction analyses, this study aims to provide SLPs with faster access to information needed to conduct these analyses efficiently. This means providing SLPs with an interface to access the precise moments a certain speaker speaks and information regarding their behaviour during the contact session. This leads to two research areas namely, **speech analysis** and **behaviour analysis**.

Speech as a research topic is well established, ranging from artificial speech to speech-to-text algorithms [22, 23]. From creating speech to understanding it. Here we will be referring to 'understanding speech' which leads to speech recognition.

Automatic speech recognition (ASR) is the ability of a program to transcribe natural speech automatically. It includes 1) isolated word recognition where words are separated by distinct pauses; 2) continuous speech recognition where sentences are produced naturally; and 3) speech understanding where the program can interpret what was said and provide an appropriate output [23].

In the recent past, special focus was given to a specific upstream process in ASR namely, speaker diarisation, which has become a research domain on its own. **Speaker diarisation** establishes ‘Who spoke when?’ and when incorporated with ASR can advance speech-to-text transcriptions to be speaker-intelligent [24]. This advanced form of ASR would be an ideal solution for automating speech interaction analysis within the domain of SLP and is widely researched in automating clinical documentation today [25]. It could provide SLPs with a person-specific transcription of the contact session, automating techniques such as LSAs and providing faster access to person-specific speech regions allowing for faster speech analyses. The effective use of ASR in clinical documentation is, however, still unexplored.

A systematic review [25] of the use of ASR in clinical documentation has found that the reported percentage of documents with errors ranged from 4.8% to 71% and that the reported word error rate ranged from 7.4% to 38.7%. The algorithms utilised vary, but generally, deep learning outperforms conventional methods when it comes to ASR [26,27].

For deep learning methods, large datasets are generally required to obtain any success. Large datasets for speech recognition particularly in English, such as those distributed by ‘Linguistic Data Consortium’ [28], do exist. Deep learning models are, however, heavily dependent on the training data, and any variation could cause poor results. In South Africa with a wide variety of spoken languages, an SLP session is never conducted exclusively in a single language. The majority of languages encountered in SLP sessions, within South Africa, are Afrikaans and English [29]. Speech recognition based on an English training set will be insufficient for Afrikaans speaking sessions. The same is true when working with toddlers or language-impaired individuals, the ASR models would prove to be unsuccessful when not trained with these linguistic variations.

ASR based on deep learning, although accurate, requires large datasets and significant computational resources to accomplish [26], and conventional methods might prove too inaccurate to be used in clinical settings [25]. The upstream process, speaker diarisation, might, however, still prove to be useful. Knowing who speaks when still provides SLPs with faster access to person-specific speech regions, allowing for less time spent on unwanted or unneeded intervals. It also provides the option for conventional audio signal processing methods as there is more leniency when clustering speech segments.

2.3.1 Speaker diarisation systems

Speaker diarisation is usually considered an upstream processing step for speaker and speech recognition [24,30]. Where speaker recognition involves identifying who spoke, and speech recognition the transcription of their speech. Speaker diarisation, on the other hand, answers the question ‘Who spoke when?’. In other words, it is the process of clustering each identified speech segment within an audio stream to a single individual.

Speaker diarisation was first introduced as an upstream processing step for speech recognition for multi-speaker audio streams to enable speaker adaptive processing [24]. However,

with increased applications to take advantage of this technology, speaker diarisation become a stand-alone research topic. In the late 1990s broadcast news became the main focus of research, creating diarisation algorithms to aid ASR in automatic transcriptions with speaker-specific metadata labels [30]. Since 2002 there has been increasing interest in the annotation of meetings. This resulted in the creation of various projects, among others, the European Union (EU) Augmented Multi-party Interaction with Distant Access (AMIDA) [31], and the EU Computers in the Human Interaction Loop (CHIL) [32]. These projects encouraged the research and development of multimodal technologies to enhance human-to-human communication interactions by automatically extracting meeting content and making it available to all participants [24, 31].

These technologies are linked to content indexing; summarising ongoing or archived meetings; and the inclusion of both verbal and nonverbal human communication (e.g., individual movement; emotions; and interactions with others). This is achieved by utilising synchronised data streams, such as audio; video; and textual information (discussion papers), that can incorporate different kinds of information, useful for structuring and analysing the content of meetings [24]. Speaker diarisation structures these meeting contents in terms of speaker turns, and allows for linguistic metadata to be added such as behavioural information, indication of a dominant speaker, and interaction levels of each speaker. These objectives correlate to that of SLPs' speech interaction analyses. Common applications for speaker diarisation, among others, are meeting transcriptions; conversational interaction analysis and behavioural modelling; audio indexing; and conversational AI. It has also been found useful in the diagnosis of autism spectrum disorder (ASD) in young children conducted by clinicians during adult-child communication interactions [33].

To date, there are two different speaker diarisation systems: Traditional speaker diarisation systems; and end-to-end neural diarisation (EEND) systems [30]. Traditional speaker diarisation systems usually consist of multiple independent sub-modules as indicated in Figure 2.2.

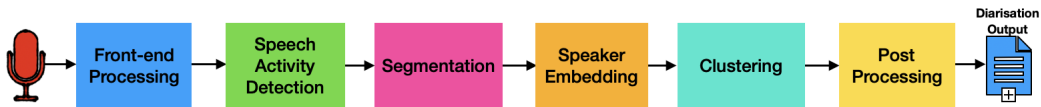


Figure 2.2: Traditional speaker diarisation systems [30].

The first module for traditional speaker diarisation systems is usually a pre-processing step (**front-end processing**), that mitigates possible artefacts corrupting the sound source (the speaker). In practice, audio recordings are not without fault. An audio source while in transit to the recording device's microphone can become corrupted by additive noise, or can cause reverberations as a result of the acoustic environment or both. Common techniques used within this module include; speech enhancement, dereverberation, and speech separation. Speech enhancement consists of primarily suppressing additive noise; dereverberation consists of mitigating the echo effect caused by room impulse responses; and lastly, speech separation consists of separating overlapping speech from multiple speakers.

SLPs' analyses usually take place within their own offices. When specialising in child verbalisation the environment is usually child-friendly; meaning it consists of carpet flooring, various soft toys, and a minimum of exposed hard surfaces. Resulting in softer acoustic conditions where reverberation is less of a factor. The same can be said for overlap speech. SLPs tend to create an environment of respected conversational turns. Allowing each individual clear time to express themselves. However, with natural conversations overlap does occur despite regulations. Additive noise, on the other hand, is always a possibility, from air conditioning, to open windows, to channel noise from the recording device. Additive noise follows a completely random signature, however, speech enhancement and denoising is a well-developed research area, from statistical methods to deep learning and under maintainable situations is completely containable [30,34].

After the audio stream has been cleaned to enhance speech, speech segments are identified by the **speech activity detection** (SAD/VAD) module. SAD distinguishes between speech and non-speech segments such as background noise [30]. SAD is an important pre-processing step in speaker diarisation systems. This is because the speech segments identified within this module are passed through for further calculations and missed segments are not. SAD algorithms usually contain two parts: feature extraction, and a classifier. Common acoustic features extracted are; pitch, zero crossing rate, signal energy, and Mel frequency cepstral coefficients (MFCCs). The classifier then uses these features to decide whether the segment belongs to speech or non-speech. Gaussian Mixture Models (GMM), and Hidden Markov Models (HMM) are traditionally used as classifiers [35].

The third module is **segmentation**. It entails segmenting the speech segments obtained from SAD into smaller segments for feature extraction. These segments are the segments that are clustered in the clustering module and therefore determine the speaker diarisation output unit. Generally, there exist two categories of segmentation: speaker-change point detection and uniform segmentation. Speaker-change point detection works on the premise of hypotheses: H_0 assumes that both the left and right speech windows belong to the same speaker, whereas H_1 assumes that they belong to two different speakers. Gaussian distributions are widely used to model the speech features of the windows [24,35,36] and slew criteria such as Bayesian Information Criteria (BIC) to establish which hypothesis fits best [30,37]. The result is, however, that the speech segments are not uniform, leading to the use of uniform segmentation, especially since the introduction of i-vectors and DNN-based³ embeddings [30].

Uniform segmentation uniformly segments the speech segments at a fixed window and overlap length, resulting in a speaker diarisation output with fixed intervals. This approach, however, does come with a trade-off error related to the segment length. The segments must be short enough to ensure the assumption that they do not contain multiple speakers, and not too short that acoustic information extracted is unreliable for speaker representation. The standard for this approach is 30ms windows with a 10ms overlap [36].

³DNN - Deep Neural Networks

The fourth module, **speaker embedding**, consists of speaker representation and similarity measures. It entails the representation of each speech segment and identifying the similarities between them. Similarity measurements are still mainly metric-based, and one of the most popular approaches is the BIC measurement, due to its effectiveness and ease of implementation. Speaker representation, over the last few decades, has been widely adapted to methods such as GMM-based Universal Background Models (GMM-UBMs); [38]; joint factor analysis methods such as the popular ~~I-vector~~ i-vector representation; [39]; and neural network methods such as the challenge-winning x-vector representation; [40, 41]. X-vectors are DNN-based embeddings that can profit from data augmentations, unlike i-vectors, allowing for an inexpensive means of multiplying training data [40], making them more suitable for smaller datasets.

The fifth module is **clustering**. It consists of grouping similar speech segments under the same cluster (potential speaker) based on speaker representations and similarity measurements established in the previous module. One of the earliest and most used clustering methods, still found in state-of-the-art systems; [41]; is the agglomerative hierarchical clustering (AHC) technique. AHC is a bottom-up clustering approach that reduces the number of clusters by iteratively merging them based on similarities until a specified criterion is met. Again, the BIC criterion is one of the most used in AHC clustering, but probabilistic linear discriminant analysis (PLDA) has been taking over as a stopping criterion [41–43]. Another, clustering method that has been indicated to outperform AHC, is spectral clustering. Many variants of spectral clustering exist, but the Ng-Jordan-Weiss (NJW) algorithm is often employed for speaker diarisation [44]. Other clustering algorithms include k-means, which is often used due to its simplicity and ease of implementation. It, however, underperforms in comparison to AHC and spectral clustering and is often used as a pre-clustering step for these algorithms.

The last module, **post-processing**, entails the further refinement of clusters created in the previous module. One such technique is resegmentation. It is the process of refining the speaker boundaries that are roughly estimated during clustering. An example thereof is the Viterbi-resegmentation based on the Baum-Welch algorithm [45]. It entails alternately obtaining the speaker GMM representation and applying the Viterbi segmentation based on the newest speaker GMMs.

After these modules have been completed the system provides a diarised output of the given audio stream. Allowing for the possibility of further analysis based on these results, such as turn-taking; dominant speaker identification; the number of speakers; and speaker-specific analysis such as conversational turns; utterance lengths; and even emotion analysis. A drawback, however, is that speaker diarisation is not speaker recognition. Meaning the system does not have information on who the speakers are. SLPs would, therefore, have to manually assign identifications to each label based on their standards and security measures.

EEND systems, on the other hand, encapsulate the sub-modules into a single neural

network. They have shown remarkable performance even against overlapping speech [46–48]. Despite their state-of-the-art performance, they are experimentally known to overfit the distribution of the speaker overlaps to that of the training data [49]. This is a result of domain mismatch and is universal in training-based methods, however, given a large enough dataset, this approach might be promising and does look to be the future for speaker diarisation systems.

As a result of EEND systems’ dependency on reliable domain-specific data, it is excluded from this study. This is because of the diverse accents, languages, language differences and disorders encountered within the SLP speech interaction analysis domain in South Africa.

2.3.2 Features

Feature extraction is domain-specific. For instance, when utilising a multi-channel audio-recording system, one would incorporate time-delay features to assess where the audio signal is coming from, allowing to separate sound sources within the pre-processing module. If one uses multimodal systems such as audiovisual then visual features are also of importance. However, for this report, the focus will be on single-channel audio systems, used throughout a traditional speaker diarisation system.

There are two main categories of features: short-term, and long-term. Short-term cepstral features are generally referred to as low-level features reflecting the voice parameters of a speaker. These features, such as Mel frequency cepstral coefficients (MFCCs), are dominantly used within speaker diarisation and speaker recognition algorithms. This is because of their ability to reflect information on the speaker’s physiology and is independent of phonetic content (text-independent). Long-term features, on the other hand, are defined as information extracted over longer regions than a frame and include phonetics among others [24, 50–52].

The idea of MFCCs is to represent an audio signal as the human cochlea perceives it. Human ears are more sensitive to the differences at lower frequencies, allowing us to distinguish what was said by whom, more so than at higher frequencies [53]. To extract MFCCs from an audio signal the following steps are required, in order: (optional) pre-emphasis; windowing; transformation to the frequency-domain (DFT); Mel-filterbank and its summation; rectifier (logarithm); and lastly the transformation to the time-domain (IDFT). A representation thereof is given in Figure 2.3 [54].

It has been found that the use of long-term features alongside MFCCs can provide a 30 % relative improvement in terms of diarisation error rate (DER) [50]. This is a result of their inherent robust nature against channel variations. Long-term features consist of phonetics; prosodics; and lexical information. Lexical information will, however, fall outside the scope of this study. Prosody relates to intonation, timing, and loudness. It captures the variations therein that can provide speaker-specific information as well as information about the affective state of the speaker. Phonetics correlates to the formation of words and

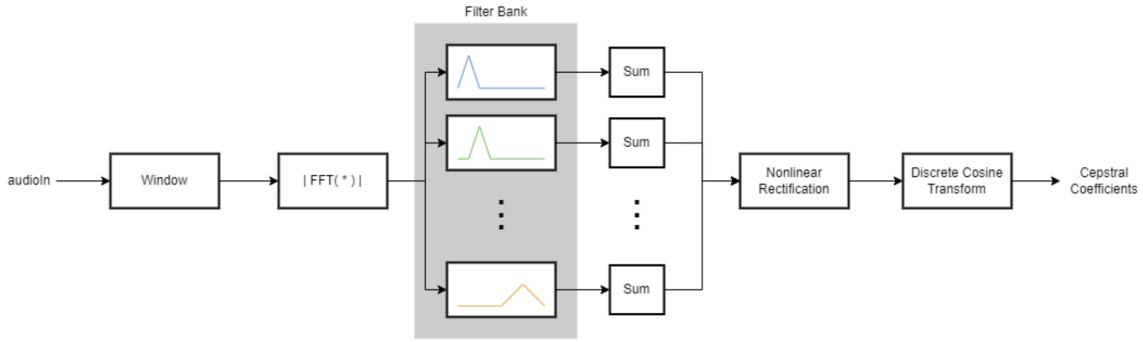


Figure 2.3: MFCC creation [54].

is very much language-dependent. According to [50], the top 12 most influential long-term features when it comes to speaker diarisation correlate to either pitch or energy. A list thereof is given in Table 2.1. A commonly used application to obtain these features is the PRAAT Library [55].

Table 2.1: Top 12 long-term features based on between-speaker and within-speaker variability ratio obtained on a per utterance basis [50].

Category	Feature	Between–Within Ratio
Pitch	F0 - Median	23.375
Pitch	F0 - Mean	9.413
Pitch	F0 - Min	3.073
Formants	F4 - Std	3.356
Formants	F4 - Median	2.191
Formants	F4 - Mean	2.616
Formants	F4 - Min	2.885
Formants	F5 - Std	2.251
Formants	F5 - Mean	2.438
Formants	F5 - Min	2.171
Harmonics-to-noise ratio	H2N - Mean	2.845
Long-term average spectrum	Ltas - std	2.322

2.3.3 Evaluations and Datasets

For speaker diarisation one of the most used evaluation tests is ~~diarisation error rate~~ DER. DER is the sum of three different error types: False Alarm (FA) of speech; missed detection of speech; and confusion between speaker labels, divided by the total duration of speech according to the referenced segments. DER is indicated in Equation 2.1. To establish a one-to-one mapping between the hypothesised output and the reference transcript, the Hungarian algorithm [56], is applied. A 250 ms ‘no-score’ collar is generally applied on both sides of the reference transcript’s annotations to account for inconsistent start and end point labelling.

$$DER = \frac{FA + MISSED + SpeakerConfussion}{Total\ Duration\ of\ Speech} \quad (2.1)$$

A newly introduced evaluation measure is the Jaccard Error Rate (JER), first introduced in the DIHARD II evaluation [57]. JER aims to evaluate each speaker with equal weight. Unlike DER which is calculated over all the speech utterances, JER first calculates a per-speaker error rate and then an average is taken. The equation for JER is provided in Equation 2.2. The speaker-confusion error within DER is perceived within JER's FA_i . Since JER uses the union of the reference and hypothesis speaker segments it cannot exceed 100%, whereas DER can become much larger than 100%. As opposed to DER, JER tends to indicate a higher than ordinary percentage when it comes to dominant speaker subsets in an audio recording.

$$JER = \frac{1}{N_{ref}} \sum_i^{N_{ref}} \frac{FA_i + MISS_i}{TOTAL_i}, \quad (2.2)$$

where $TOTAL_i$ is the union of the i_{th} speaker's speaking time in the reference transcript and the i_{th} speaker's speaking time in the hypothesis transcript. N_{ref} is the number of speakers in the reference script. ~~The speaker-confusion error within DER is perceived within the FA_i in JER. Since JER uses the union of the reference and hypothesis speaker segments it cannot exceed 100%, whereas DER can become much larger than 100%. As opposed to DER, JER tends to indicate a higher than ordinary percentage when it comes to dominant speaker subsets in an audio recording.~~

These evaluation metrics are usually applied to standardised datasets to ensure comparability between different studies. A summary of commonly used datasets is given in Table 2.2. A small description follows, for more information see [30].

Table 2.2: Commonly used dataset for speaker diarisation comparisons [30].

Datasets	Language	Size [hr]	Style	# Speakers
CALLHOME	Multilingual	20	Conversation	2-7
AMI	English	100	Meeting	3-5
ICSI Meeting	English	72	Meeting	3-10
CHiME-5/6	English	50	Conversation	4
VoxConverse	Multilingual (Mostly English)	74	Youtube Videos	1-21
LibriCSS	English	10	Read Speech	8
DIHARD II Tr.3,4	Multilingual	262 (dev), 31 (eval)	Miscellaneous	4
DIHARD III Tr.1,2	Multilingual	34 (dev), 33 (eval)	Miscellaneous	1-7

CALLHOME: NIST SRE 2000: CALLHOME [58] was first created for the NIST 2000 Speaker Recognition Evaluation (SRE) challenge and has since become widely used for speaker diarisation evaluations. The dataset contains approximately 500 sessions of multilingual (Arabic; Chinese; English; German; Japanese; and Spanish) telephonic speech. The value of the dataset comes from its diversity in terms of speakers, speaking styles, and conversational scenarios. It also introduces challenges to speaker diarisation systems, as it reflects real-world scenarios with overlapping speech, and speakers having various

speaking behaviours in a single session. Each session has two to seven speakers with two dominant speakers in each conversation.

AMI Corpus: The AMI Meeting Corpus [59] is a multimodal dataset consisting of 100 hours of meeting recordings from three recording sites (offices) over 171 meeting sessions. Each meeting has three to five speakers. The dataset provides audio data from body microphones that provide amplified speaker recordings. As well as audio data from omnidirectional microphones that are placed centrally during the meetings. It provides forced-aligned data that contains words and phoneme level timings along with its transcript and speaker labels, making it ideal for speech recognition algorithms.

ICSI Meeting Corpus: The ICSI Meeting Corpus [60] consists of 922 speech files (approximately 72 hours in total) recorded over 75 meetings. The audio files contain a sampling rate of 16 kHz⁴ (downsampled from 48 kHz) with a 16-bit resolution (big-endian). The audio sources are body microphones as well as omnidirectional microphones placed centrally in the room, all recording simultaneously. There are a total of 53 unique speakers were each meeting involved 3 to 10 speakers.

CHiME-5/6 Challenge: The CHiME-5 dataset [61] consists of multi-speaker (fixed to four), domestic conversational recordings. Over 20 houses, each with two or three different recording areas are used to create various acoustic challenges (room acoustics; background noise). The areas include the kitchen and dining/living rooms. The audio is collected from six four-channel microphone arrays, as well as body microphones, and contains over 50 hours of audio data. The dataset was created for the CHiME-5 challenge [62], and is also used in the CHiME-6 challenge [63], as well as the DIHARD II challenge [57].

VoxSRC: The VoxCeleb Speaker Recognition Challenge (VoxSRC) [64] is a speaker recognition challenge organised by the University of Oxford to probe how well current systems can recognise speakers from speech obtained ‘in the wild’. Recent VoxSRC challenges (2023) feature four tracks: Tracks 1 to 3 are speaker verification tracks and Track 4 is speaker diarisation. The datasets available for this challenge are the VoxCeleb (1 and 2); CNCeleb; VoxMovies; and VoxConverse. All of these are created from YouTube videos of celebrity interviews; news shows; talk shows; and debates, consisting of audio from both professionally edited videos as well as more casual conversational audio in which background noise; laughter; and other artefacts are observed in a wide range of recording environments [65]. VoxConverse which is usually used for speaker diarisation validation, contains over 60 hours of validation data, as well as over 33 hours of testing data. VoxCeleb is larger still with over 2000 hours of audio. The datasets are multilingual but contain mostly English content.

LibriCSS: The LibriCSS dataset [66], contains 10 hours of multichannel recordings collected from a 7-channel microphone array within a meeting room. The audio source is a playback of the audio found in the LibriSpeech corpus [2]. The LibriCSS dataset consists of 10 ‘meeting’ sessions, which are further divided into six 10-minute mini-sessions. Each

⁴kHz - kilohertz

mini-session contains 8 speakers and is designed to have different overlap ratios from 0% to 40%. A baseline system has been developed for speech separation and ASR; and for speech separation, ASR, and speaker diarisation evaluations.

DIHARD Challenge: The DIHARD challenges [67], focus on state-of-the-art speaker diarisation systems in difficult domains. The evaluation data includes challenging data such as clinical interviews, web videos, and speech in the wild (e.g., recordings in restaurants), as well as less opposing data such as audiobooks. With the newest iteration of the DIHARD challenge (DIHARD III), there are merely 2 evaluation tracks: Track 1 for diarisation from a provided SAD, and Track 2 for diarisation and speech segmentation (without SAD). A conversational telephone speech (CTS) and clinical dataset were also introduced for this newest iteration for evaluation purposes. The metrics used are primarily DER and secondarily JER, without a ‘no-score’ collar and with overlap regions.

Rich Transcription Evaluation Series: The RT Evaluation [68] was a pioneering evaluation series in ASR and speaker diarisation in the early 2000s. The audio data utilised in these evaluations usually included broadcast news, CTS, and meeting recordings with multiple speakers. The RT evaluation series promoted and gauged advances in speaker diarisation and ASR technologies, and ended with the RT-09 evaluation in 2009.

2.3.4 Obstacles

One of the biggest challenges for speaker diarisation systems to be wary of is overlapped speech. Overlap is inevitable when it comes to conversations, some reports have shown an average of 12% to 15% of overlap in meeting recordings [69, 70], and even more so in daily conversations [62, 71]. Despite the awareness of this problem some evaluation standards still disregard overlapped speech in their evaluation sets as it results in major degradation, especially in traditional clustering-based speaker diarisation systems [68]. One example is a traditional system utilising MFCC features, speaker change detection for segmentation, and AHC for clustering had an average DER degradation from 2.21% to 21.08% [72].

Other challenges include domain mismatch and online processing. Domain mismatch is especially a problem when incorporating training-based methods in the speaker diarisation system, such as EEND systems [49]. This is because models that are trained on data from a specific domain (acoustic environment) usually perform poorly in another. Online processing, on the other hand, investigates the issue of speaker diarisation latency as some applications require fast transcriptions and most systems assume that the entire recording is first conducted and then transcribed.

The evaluation of overlap speech is outside the scope of this study. The use of deep learning speaker diarisation models such as EEND appears to be unsuitable for this study due to the absence of a suitable English and Afrikaans speech corpus containing various South African accents. Similarly, the use of online processing is outside the scope of this study.

2.4 Filling the gaps: Behaviour analysis

Behaviour is defined as the way an object (person) acts or behaves in response to a certain situation or stimulus. When dealing with behaviour analysis in people emotional regulation (the ability to regulate emotional reactions to environmental cues) is the main focus. In the field of speech-language pathology and psychology, the importance of early development of emotion regulation becomes quite apparent. Its influence reaches to personality, social competence, and problematic behaviours [73–75].

SLPs evaluate patients’ behaviour during certain interactions or tasks. The behaviours they typically assess are related to speech. However, it can be challenging to assess many behaviours in children who cannot communicate verbally or effectively. SLPs aim to identify underlying conditions or disorders that may hinder proper communication, such as ~~autism spectrum disorder (ASD)~~ ASD or specific language disorders (SLDs). Anxiety is another condition that can significantly affect speech interactions, especially between parents and children [9], and it may be considered the underlying cause that disrupts or inhibits most speech interactions.

This subsection explores the domain of behavioural analyses from the perspective of an SLP. It defines anxiety and investigates its correlation with the human heart. It illustrates measurements widely used for the classification of autonomic nervous responses and identifies related factors.

2.4.1 Anxiety

In this study stress and anxiety will be used interchangeably. Various definitions of stress exist. Hans Selye defines stress as a response to a stimulus, that breaks the mental or physical balance and stability of the body, to maintain the previous state of stability or homology [76]. Kenneth Hambly defines stress, however, as a maladaptive state in which the sympathetic nervous system (SNS) is overactivated, causing acute or chronic physical, psychological, and behavioural impairment [77]. Repeated exposure to stress and the dysregulation thereof has been linked to adverse health outcomes among children, such as obesity, asthma, inflammation, and depression [78, 79]. Research regarding stress typically considers the autonomic nervous system (ANS) and the hypothalamic-pituitary-adrenal (HPA) axis [10, 78–81]. These two stress systems coordinate the response of many physiological systems to a stressor, including the immune and cardiovascular systems, bringing the body back to homeostasis [79].

When a stimulus or situation threatens the homeostasis of the body, i.e., a stressor, the ANS and HPA axis becomes activated, following an interconnected and temporal sequence to maintain homeostasis [78]. The ANS quickly promotes physiological changes through synaptic transmissions by its two branches: the sympathetic and parasympathetic nervous systems (SNS and PNS, respectively). The SNS is responsible for the “fight or flight” response, preparing the body to take action, the PNS on the other hand is responsible for maintaining and returning the body to homeostasis. A diagram thereof is given in Figure

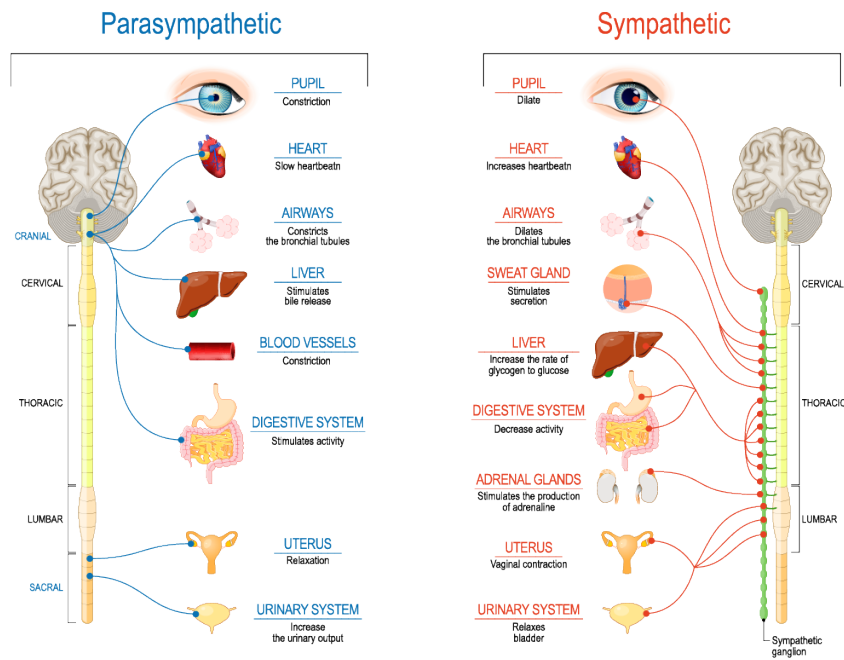


Figure 2.4: Diagram of the SNS and PNS systems, indicating the functions of each in the human body [82].

2.4, obtained from [82]. When the body prepares for a stressor the SNS becomes activated and the PNS withdraws its inhibitory effects to facilitate the process [80, 83]. One of the most prominent indicators of PNS activity is vagal tone. It is often used as a measure of self-regulation [80, 83], and stems from the tenth cranial nerve or vagus. The vagus nerve is believed to be responsible for homeostasis via bidirectional messages between the internal organs (e.g., heart and lungs) and the brain [84, 85]. Vagal tone, itself, cannot be measured directly, however, some of its effects can.

Although the heart's natural pace occurs automatically as a result of its various pacemaker tissues, the heart rate (HR) and rhythm are primarily controlled by the ANS [86]. It was Lacey and Lacey in the early 1960s [87], who first noted the change in heart rate due to the change in one's environment, where a decrease meant acceptance and an increase in rejection. Rajendra et al. [88], also found that an increase in SNS activity and a decrease in PNS activity led to cardio-acceleration (increased excitation frequency and conduction velocity), whereas decreased SNS and increased PNS activity led to cardio-deceleration. This means the SNS and PNS can modulate the HR intervals of the QRS complex at distinct frequencies. This variability of the heart rate, known as heart rate variability (HRV), indicates the heart's ability to respond to physiological and environmental factors such as mental stress, physical exercises, breathing, and sleep [88], indicating the importance of HRV measurements.

Conversely, the HPA axis is a hormonal system. With the activation of the HPA axis, the body releases cortisol from its adrenal cortex, impacting various physiological systems (e.g., immune system, metabolism) [78]. It also plays a supportive role in augmenting the

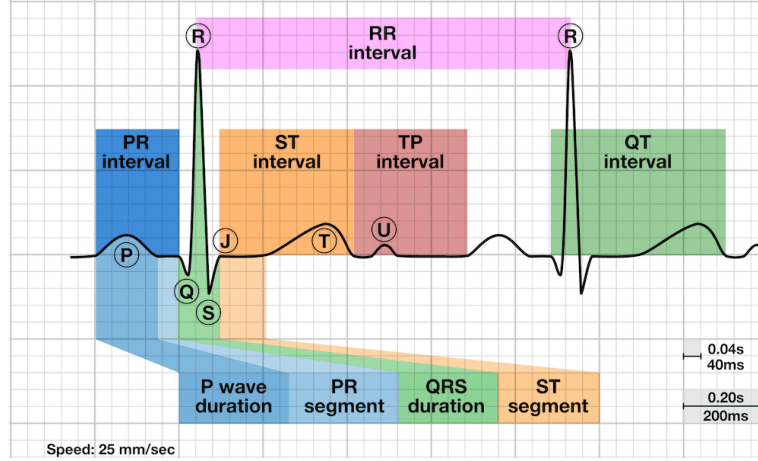


Figure 2.5: Detailed (single-lead) ECG signal [94].

activity of the ANS such as enhancing the sympathetically mediated cardiovascular response to stress [89]. Cortisol measurements, such as AUC_{AG} and AUC_1 , can be obtained from saliva samples minutes after the stressor has occurred. These samples are a proven means of stress indication [78, 79]. Another is through neural and immune biomarkers in sweat [79].

The use of cortisol in saliva, and biomarkers in sweat, although reliable stress indicators, has the problem of not indicating the moment a stress response occurs, unlike HRV.

2.4.2 Heart rate variability

Heart rate variability (HRV) is a physiological signal derived from the length fluctuations between consecutive beats. HRV increased in popularity [10, 86, 90–93] because of its non-invasive and continuous means of identifying physiological changes within the body. This has been made possible through the use of electrocardiograms (ECGs). Electrodes are placed strategically on the chest and abdomen of an individual. The ECG waveform consists of various segments and intervals that can be used by healthcare professionals to analyse and detect arrhythmias in patients. An example of an ECG waveform and all its components is depicted in Figure 2.5 obtained from [94]. Heart rate (HR), also derived from an ECG signal, is the number of R-peaks observed in a given minute, measured in beats per minute (bpm). Heart rate variability (HRV), on the other hand, is the variation between two consecutive RR intervals, typically measured in milliseconds (ms).

With the advances in heart monitoring technology and standards being set by the Task Force of the European Society of Cardiology and the North American Society of Pacing and Electrophysiology [86], and further concurring guidelines from the Society for Psychophysiological Research [90], HRV became a strong focus of psychophysiological research [95]. This is because, HRV shows a strong correlation to the parasympathetic nervous system (PNS) [86], which is further associated with psychophysiological aspects, such as self-regulation mechanisms linked to cognitive, affective, social, and health phenomena [80, 83, 95, 96]. Within the HRV data there exist various variables that show

Table 2.3: Summary of the main HRV variables in psychophysiological research and their origins [95].

Category	Variable [unit]	Description	Physiological Origin
Time-domain	SDNN [ms]	Standard deviation of the NN interval	Cyclic components responsible for HRV
	RMSSD [ms]	Root mean square of successive differences	Vagal tone
	pNN50 [ms]	Percentage of normal sinus NN-intervals more than 50ms	Vagal tone
	Peak-valley [ms]	Time-domain filter dynamically centred at the exact ongoing respiratory frequency	Vagal tone
Frequency-domain	ULF power [ms ²]	Ultra-low frequencies ULF ≤ 0.003 Hz	Circadian oscillations, core body temperature, metabolism and the renin-angiotensin system
	VLF power [ms ²]	Very-low frequencies $0.003 \text{ Hz} < \text{VLF} \leq 0.04 \text{ Hz}$	Long-term regulation mechanisms, thermoregulation and hormonal mechanisms
	LF [ms ²]	Low frequencies $0.04 \text{ Hz} < \text{LF} \leq 0.15 \text{ Hz}$	A mix of sympathetic and vagal activity, baroreflex activity
	HF [ms ²]	High frequencies $0.15 \text{ Hz} < \text{HF} \leq 0.4 \text{ Hz}$ (Adults) $0.24 \text{ Hz} < \text{HF} \leq 1.04 \text{ Hz}$ (Children)	Vagal tone
	LF/HF [%]	Low-high frequency ratio	Mix of sympathetic and vagal activity
Non-linear	SD1 [ms]	Standard deviation - Poincare plot Crosswise	Unclear, depicts quick and high frequent changes in heart rate variability
	SD2 [ms]	Standard deviation - Poincare plot Lengthwise	Unclear, depicts long-term changes in heart rate variability

a strong correlation to cardiac vagal tone (parasympathetic activity). A summary of the main HRV variables and their physiological origins are given in Table 2.3 obtained from [95]. Other simple measurements often used in stress classification applications include mean RR-interval, and mean HR [97, 98].

The so-called normal-to-normal (NN) intervals are inter-beat intervals (IBIs) from which artefacts have been removed. IBI is the time interval between consecutive heartbeats. Time-domain indices assess the amount of HRV recorded throughout monitoring periods ranging from approximately 2 minutes to 24 hours. Frequency-domain values compute the absolute or relative amount of signal energy contained within component bands. Nonlinear measurements quantify the unpredictability and complexity of a set of IBIs [92].

All these measurements (time-domain, frequency-domain, and non-linear) are dependent on their analysis period, therefore standardised long-term and short-term recordings are defined as a 24-hour recording period and a 5-minute recording period, respectively [86]. Short-term variables (e.g., RMSSD and pNN50) usually indicate high-frequency variations in heart rate, whereas long-term variables (e.g., SDNN) indicate very-low-frequency to ultra-low-frequency variations. Despite high correlations between short-term variables,

RMSSD is probably the most widely used short-term variable because of its statistical superiority. The variable, RMSSD, reflects the vagal tone and has a high correlation with the high-frequency (HF) HRV frequency-domain measurement [99]. The benefit of RMSSD, however, is that it seems to be relatively free of respiratory influences, unlike HF in the frequency domain [100].

For the analysis of momentary stress short-term derivatives play a significant role, more so than long-term. Two sources contribute to short-term HRV measurements. The first is the ANS via the SNS and PNS branches. The second is the body's regulatory mechanisms that control HR via respiratory sinus arrhythmia (RSA), the baroreceptor reflex (negative-feedback control of blood pressure), and rhythmic changes in vascular tone [92]. RSA refers to the respiration-driven speeding and slowing of the heart via the vagus nerve [101]. The use of RSA and HF power as vagal tone indices are, however, still up for debate [92, 102, 103]. This is because of the link between RSA and HF with respiration frequency and its influence on heart rate. Eckberg describes RSA as a state dependent on and affected by factors such as breathing pattern, sleep and wakefulness, psychological stress, and physical activity [104]. Moreso, HF power shows an increase as respiration frequency is reduced and tidal volume increases [105].

Frequency-domain measurements are divided into frequency bands with set allocations (see Table 2.3 above). These measurements are expressed as either absolute or relative power. Absolute power is calculated as milliseconds squared divided by cycles per second (ms^2/Hz). Relative power is calculated as the absolute power of a specific frequency band divided by the summed absolute power of the LF and HF bands and is measured in normal units (nu). The **ultra-low frequency (ULF)** band (≤ 0.003 Hz) indexes fluctuations in IBIs with a period from 5 minutes (inverse of 0.003 Hz) to 24 hours and can solely be calculated from 24-hour recordings. The **very-low frequency (VLF)** band (0.003 – 0.04 Hz) is comprised of rhythms with periods between 25 to 300 seconds and can be derived from short-term recordings. The **low frequency (LF)** band (0.04 – 0.15 Hz) has rhythms with periods between 7 and 25 seconds and is affected by breathing rates from approximately 3 to 9 bpm. The **high frequency (HF)** or **respiratory** band (0.15 – 0.4 Hz for adults) is influenced by breathing rates from approximately 9 to 24 bpm [86, 92]. The HF band, however, is adjusted to account for the population of interest, for example, the band is widened to 0.15 to 1.00 Hz for physical activities or from 0.24 Hz to 1.04 Hz for very young children at rest [106], depending on the respiratory rate. The **LF/HF ratio** may estimate the ratio between the SNS and PNS activity under controlled conditions [92].

Lastly, the calculation for the power density spectrum (PSD), the measure of the signal's power content (i.e. variance) versus frequency, is generally classified as non-parametric and parametric. The advantages of non-parametric methods are two-fold 1) ease of calculation (Fast Fourier Transform (FFT) - in most cases) and 2) increased processing speed. The advantages of parametric are three-fold, 1) smoother spectral components that can be distinguished independently of pre-selected bands, 2) easy post-processing of the spectrum

with an automatic calculation of low and high-frequency power components and easy identification of the central frequency of each component, and lastly, 3) an accurate estimation of the PSD on which the signal is supposed to maintain stationarity. The disadvantage of parametric methods is the need to verify the sustainability of the chosen model and its order [86]. A study done on the optimum order of autoregressive models (parametric) for HRV, found an order of no less than $p = 16$ should be used at a 4 Hz sampling rate [107]. For more information on HRV measurement standards see [86,90] or for recommendations on HRV research in psychophysiological applications see [95].

2.4.3 HRV: Related Factors

A healthy heart is not simply a metronome, but a complex system capable of oscillating under different physical and psychological stresses. Some of the variables that can cause heart oscillations are:

Respiration: Respiration frequency and tidal volume influence heart rate [105]. Greater tidal volumes and lower respiration rates enhance RSA and HF power [108] while increasing respiration rates may lower short-term time-domain measurements and LF power. This correlates to when the body is experiencing psychological stress.

Age: Age is a significant factor. The heart grows with the body and undergoes continuous physiological changes from neonate to adulthood [109]. At birth, the right ventricle is larger and thicker than the left ventricle, reflecting high pulmonary vascular resistance *in utero*, and persists up to three years of age. This produces the presence of a right axis and dominant R waves (R wave voltage larger than S wave voltage) in the right precordial leads of the ECG monitor and inverted T waves. This is a reflection of right ventricular hypertrophy in adults [94, 109]. Moreso, the conduction intervals (PR interval and QRS duration) are shorter than in adults due to the smaller cardiac size. Heart rates are much faster in neonates and infants decreasing as the child grows.

The variation of the normal PR interval as it correlates to age and heart rate is shown in Table 2.4, provided by [94]. It was also found that HRV time-domain measurements decline with age [110].

Gender: Women have generally higher mean HR (smaller RR intervals) and lower SDNN values, especially in long-term recordings than men. Furthermore, they have a lower total, VLF, and LF power, but greater HF power. Women show relative vagal dominance, despite higher mean HR, and men, despite their lower HF power, show relative SNS dominance [92, 111].

Heart rate: An accelerated heart rate reduces the time between successive beats and therefore the opportunity for the IBIs to vary, lowering the HRV [92]. The inverse has the reverse effect. This phenomenon is called *cycle length dependence* [91].

Health: An individual's physical health also impacts HRV values. Autonomic dysfunction is when part of the ANS or entire ANS is affected. If the ANS is altered the heart rate

Table 2.4: Normal PR interval as it correlates to age and heart rate (and upper limit of normal) [94].

Rate (bpm)	0-1 month	1-6 month	6-12 month	1-3 year	3-8 year	8-12 year	12-16 year	Adult
<60						160 (180)	160 (190)	170 (210)
60-79					150 (170)	150 (170)	150 (180)	160 (210)
80-99	100 (120)				140 (160)	150 (160)	150 (170)	150 (200)
100-119	100 (120)			(150)	130 (160)	140 (150)	150 (160)	150 (190)
120-139	100 (110)	110 (140)	110 (140)	120 (140)	130 (150)	140 (150)		150 (180)
140-159	90 (110)	100 (130)	100 (130)	110 (140)	120 (140)			(170)
160-179	100 (110)	100 (120)	100 (120)	100 (140)				
>180	90	90 (110)	100 (110)					

is affected, and some symptoms may include the inability to regulate heart rate during exercise, or dizziness and fainting. Medications such as beta-blockers, and anti-depressants also affect the heart rate and are usually excluded from HRV studies [95]. Furthermore, increased aerobic fitness increases time-domain measurements. In general, HRV time-domain metrics increase with increased health status. Other common considerations when conducting HRV studies are alcohol intake, smoking, caffeine, exercise, weight and height, and oral contraceptive intake for female participants [95].

These factors should be taken into consideration when collecting and analysing heart rate-related data.

2.4.4 Heart monitors and other physiological sensors

There are various ways of measuring heart rate activity: ECG and photoplethysmography (PPG). ECG signals are collected through various electrodes attached to specific anatomical positions. Clinical ECG recordings typically use a 12-lead placement for the determination of complex temporal dynamics of each cardiac cycle. However, for HRV analyses a three-lead placement will suffice. The three-lead placement is known as the Einthoven triangle (see Figure 2.6) and is placed equidistantly on the chest wall around the heart. Lately, there have also been accurate single-lead devices for the calculation of HRV values. Devices such as the *Polar H10* [112] and *Apple Watch* [113], are placed on the chest and wrist, respectively. The Polar H10 has been proven to be the gold standard for HRV analyses during physical extensive studies [114, 115] and even showed improvements against standardised Holter monitors.

PPG sensors, on the other hand, use light-emitting diodes to continuously monitor changes

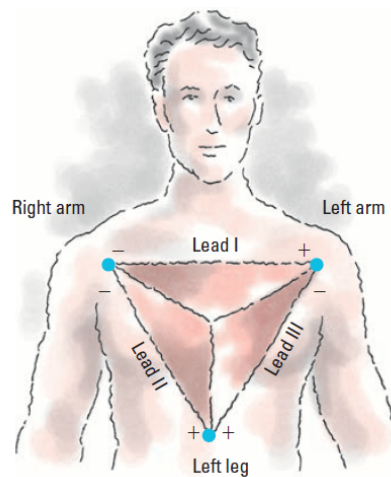


Figure 2.6: Einthoven triangle [116].

in blood volume in the capillaries (typically of the wrist, finger, and earlobes). PPG sensors are commonly used in smartwatches today and offer easy access to heart rate information. The use of PPG for accurate HRV analyses is still debatable [115,117].

Other than ECG and PPG sensors, a study using an EQ radio has found very promising results in not only determining heartbeats through RF signals but also classifying the emotional state of a person [118]. The disadvantage of this approach is that requires the individual to remain stationary during the collection process and therefore fails the requirements of this study.

With the current availability and cost of R 1999, the gold standard Polar H10 seems to be a viable option.

Besides HRV there are other means of evaluating physiological responses. In evaluating affecting states and stress identification the following physiological sensors are commonly studied: electroencephalogram (EEG) [119]; electromyography (EMG) [120]; galvanic skin response/electrodermal activity (GSR/EDA) [121]; and multimodal approaches thereof [122]. This study makes note thereof but will not explore it further.

2.5 Filling the gaps: Stress Identification

Stress identification from an engineering perspective consists of three major steps: pre-processing, feature extraction, and classification. An overview of these steps, from the utilisation of ECG signals and HRV analyses, is provided in this section.

2.5.1 Pre-processing

The HRV calculations stem from accurately detecting heartbeats and determining the time interval between them. When using ECG signals one is reliant on a clear depiction of the QRS-complex. However, the observed ECG signal can be corrupted by artefacts (noise), such as baseline wandering, electrode contact artefacts, muscle contractions, and power line interferences [123, 124]. Raising the need for denoising algorithms. A brief overview of noise and artefacts that can influence an ECG signal is described below.

- a) **Power line interference (PLI)** - This is the most common noise detected on an ECG signal. It originates from inductive interference between the power line and the electronics of the ECG monitor. Depending on the region PLI is present at either 50 Hz or 60 Hz, with a bandwidth below 1 Hz [123].
- b) **Electrode contact noise** - This is a randomly occurring noise with an amplitude equal to the maximum recorder output. It originates from the loss of contact between the skin and electrodes. A weak connection can result in a baseline transition, which decreases exponentially to the baseline value [123]. If there is no contact then the signal is lost.
- c) **Baseline wandering** - This is a low-frequency noise, originating from the respiration of an individual. It correlates to the respiratory band typically 0.15 – 0.30 Hz for adults but can change depending on the respiratory rate [124, 125].
- d) **Electromyographic (EMG)** - It originates from muscle contractions, causing noise in the region of 0 – 10 kHz. The average amplitude of EMG noise is about 10% of the peak-peak ECG signal amplitude. It overlaps with the ECG signal at 1 – 100 Hz [125], making it one of the more difficult noises to remove.

Examples of these noises and artefacts are given in Figure 2.7 obtained from [124].

Noise reduction: To handle these noises and artefacts various approaches were developed. The simplest of which are digital filters, e.g. a median filter [124]. A median filter is a non-linear filter often used as a pre-processing step for the removal of impulse noise, as well as baseline wandering. It works on the principle of setting each output sample to the median of its neighbouring samples under a given moving window, limiting outliers and the effect of low-frequency trends. Besides ECG signals, HRV, by nature is a non-stationary time series. This means that due to the variability of the heart's rhythm, there

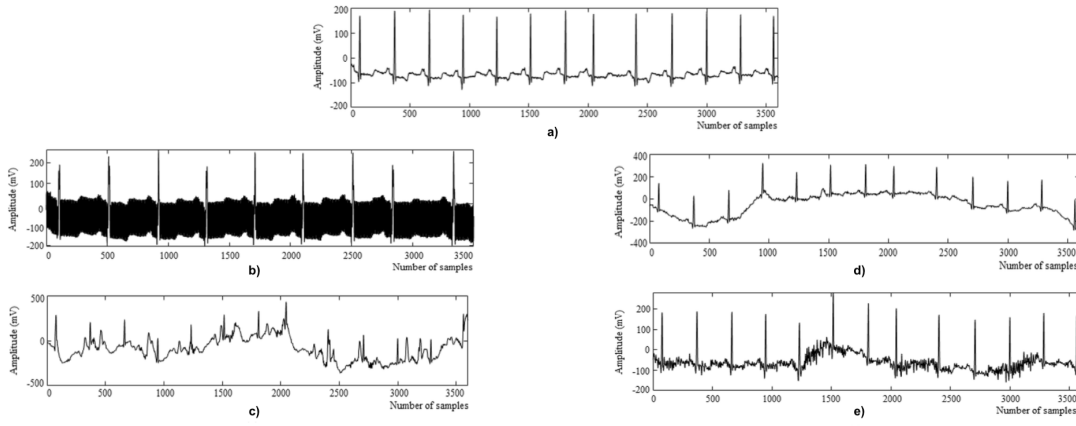


Figure 2.7: ECG noise types [124]. a) Reference signal - MIT-BIH ECG record 100; b) PLI noise; c) Electrode contact noise; d) Baseline wandering; e) EMG noise.

also exist variations in its variability causing trends in the HRV data. An example hereof is depicted in Figure 2.8.

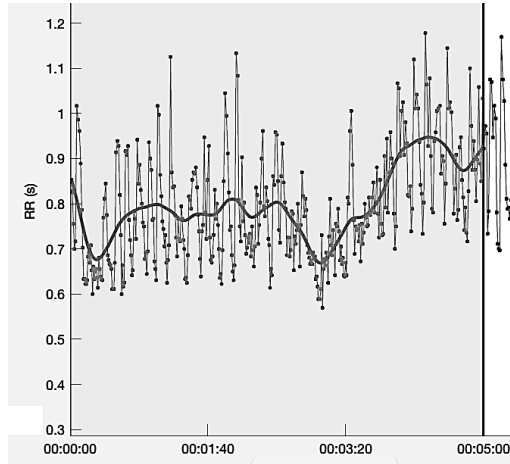


Figure 2.8: Variability in HRV data. HRV trendline is represented by the thick black line moving horizontally. RR-interval data points are represented by point markers.

A study conducted by Tarvainen et al. [126], utilised a time-varying FIR high-pass filter based on a smoothness-priors approach to detrending HRV data. The importance hereof is that spectral analyses (frequency domain measurements) expect a stationary input, as non-stationary data can distort time and frequency domain measurements [126], affecting further analyses. The importance of re-sampling HRV data to be equidistant before spectral analysis also becomes apparent.

ECG signals being non-stationary, possess complex time-frequency characteristics which cannot be analysed by time or frequency-domain techniques only. This led to the use of short-time Fourier transform (STFT) [127]. STFT provides content on how the frequency component of the signal varies with time. It does so by applying a Fourier transform to a moving fixed-width time window, extracting the frequency content of each segment. The size of the window is usually narrow such that the segment appears stationary. The

limitation hereof is that there is a trade-off between the time and frequency resolutions. A narrow time window provides good time resolution but poor frequency resolution, and vice versa.

A wavelet transform reduces this limitation [124]. Wavelet transforms improve the time-frequency resolution by varying the window size. The use of varying windows allows for the capturing of long, low-frequency, and small, high-frequency content simultaneously. Wavelet transforms use what are called wavelets to represent a signal. A wavelet is a waveform with a limited duration and a mean of zero. A limited list of wavelets is depicted in Figure 2.9. The wavelet that corresponds the most to the pattern of the analysed signal is typically used. For QRS-complex representation either Daubechies 6 (db6) or Symlets 4 (sym4) is used [128, 129]. Higher-order Daubechies wavelets are commonly preferred for preserving ECG features, while lower orders may be better for noise reduction. This stems from the impact the wavelet order has, lower orders have lower cut-off frequencies, preserving low-frequency information, while high-order wavelets have higher cut-off frequencies resulting in better high-frequency details while sacrificing some low-frequency information. The same is true for Symlet wavelets, however, they have the additional benefit of being symmetrical, balancing the preservation of ECG features with noise reduction.

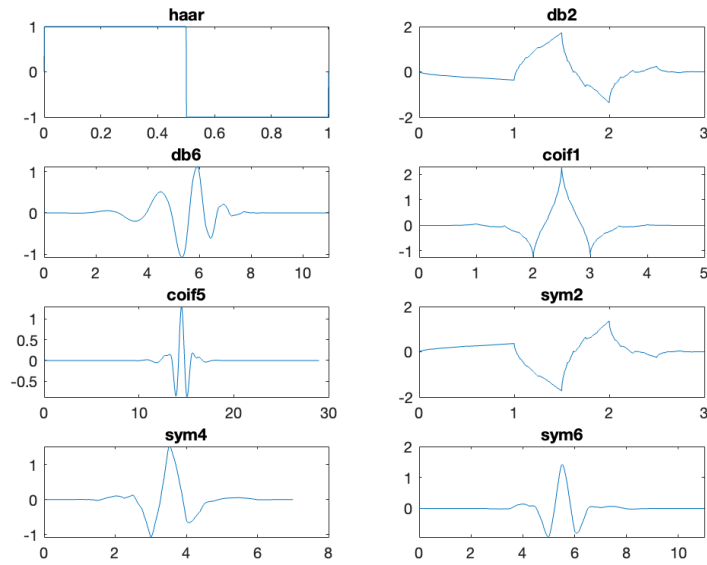


Figure 2.9: Limited wavelet list. (MATLAB self-generated image)

Employing these wavelets is done using either discrete wavelet transforms (DWTs) or continuous wavelet transforms (CWTs). CWT is used for continuous signals and has an exponential scale with a base smaller than 2, resulting in greater resolution but also computational requirements. DWT is a finite version of CWT using orthogonal wavelet bases that allow for dyadic scales and time steps. Due to the DWT's computational efficiency, it is often applied in beat detection and denoising algorithms [128]. The resolution of DWT has recently been improved by adding redundancy to the time steps (not downsam-

pling after decomposition), and is known as stationary wavelet transforms (SWTs). The decomposition of a waveform with the use of SWT is depicted in Figure 2.10 [130].

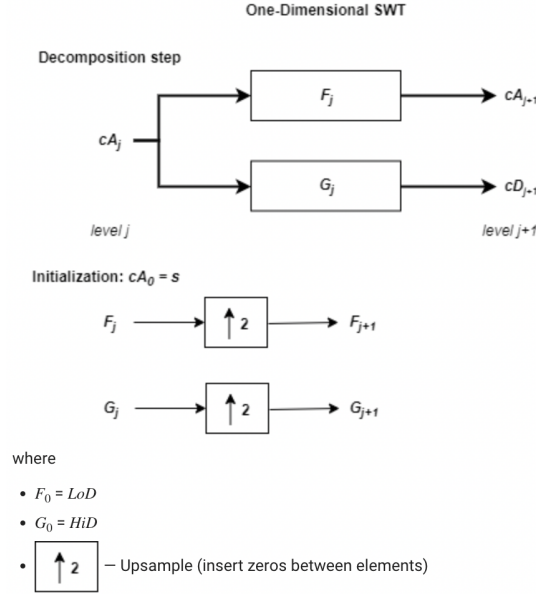


Figure 2.10: SWT decomposition [130].

Wavelet transforms have proven to be very useful in denoising ECG signals [124,128], and even found a place among QRS-complex detection algorithms [129].

2.5.2 QRS-detection algorithms:

The detection of cardiac beats within an ECG signal is crucial for the analysis of heart rate, rhythm, and derived cardiac analyses, such as ECG stress identification. One of the most widely used beat detection algorithms is the Pan-Tompkins algorithm. The algorithm consists of a pre-processing step followed by a decision. The pre-processing involves bandpass filtering (removing power line interference, baseline wander, and noise components), followed by differentiation (enhancing high-frequency components that make out the QRS complex), squaring (to highlight peaks), and moving average filtering (smooth close-by peaks). The decision rules are based on an adaptive amplitude threshold and a minimum time between successive R-peaks.

The Pan-Tompkins algorithm is effective but is solely based on R-peak detection and may require additional processing or algorithms for detecting other ECG features such as the Q-wave and S-wave, if necessary. Sabherwal et al. [128], utilised the independent detection of S-peaks to aid in detecting QRS complexes in ECG signals. It is based on the use of DWT to suppress noise and extract merely the QRS complex from the ECG signal. Digital filters are then applied to enhance the S-peak and reduce the influence of the Q and R-peaks. Thereafter, the location of the S-peaks is obtained by calculating the boundaries of the QRS complex using an adaptive threshold based on the zero-crossing rate of an added adaptive high-frequency signal. The maximum value within this boundary then correlates to the S-peak. The algorithm accompanied by the detection of R-peaks

obtained a 99.990% sensitivity and positive predictive value, and an error rate of 0.135%. The drawback of this approach is the identification of the QRS boundaries, which requires the determination of various adjusting parameters, making it quite sensitive.

Other methods would include the use of machine learning models and pattern recognition algorithms, such as Hidden Markov Models (HMMs), artificial neural networks, and support vector machines all of which require annotated datasets.

2.5.3 Feature extraction

After detecting the heartbeats and deriving HRV, the process of feature extraction can commence. HRV features are categorised into three groups: time-domain, frequency-domain, and non-linear. Table 2.5 presents some of the most commonly used features in psychophysiological research. The features that show the highest correlation to stress are identified in the table as having a vagal tone physiological origin. These features include RMSSD, pNN50, peak-valley, and HF power. Among these features are mean RR interval, and mean heart rate, which are often applied in stress identification, [97,98]. These features are further described in Table 2.5.

Table 2.5: Typical features utilised in stress identification studies.

Feature	Description [unit]	Formula
Mean RR interval	Mean of RR interval [ms]	$\overline{RR} = \frac{\sum_{i=1}^n RR(i)}{n}$, $n = \#$ of RR intervals
Mean HR	Mean heart rate [bpm]	$\overline{HR} = \frac{\sum_{i=1}^n HR(i)}{n}$, $n = \#$ of HR
RMSSD	Root mean square of successive differences. [ms]	$RMSSD = \sqrt{\frac{\sum_{i=1}^{n-1} [NN(i+1) - NN(i)]^2}{n-1}}$ $RMSSD = \sqrt{\frac{\sum_{i=1}^{n-1} diff(NN)^2}{n-1}}$ $, n = \#$ of NN intervals
pNN50	Percentage of normal sinus NN-intervals more than 50ms. [ms]	$pNN50 = \frac{count(abs(diff(NN)) > 50ms)}{n-1} \times 100$ $, n = \#$ of NN intervals
HF absolute power	High-frequency absolute power. [ms]	PSD(of HF-band), PSD = power spectral density, HF-band = 0.15 – 0.4 Hz for adults and 0.24 – 1.04 Hz for toddlers.

Before calculating the frequency domain features the HRV is made equidistant. This is accomplished by re-sampling the HRV signal at a constant rate (typically 2 – 8 Hz) with the use of cubic spline interpolation, thereby creating a signal that is equally spaced and can be used in frequency analyses. Thereafter, the power spectral density of the new equidistant signal is calculated. This can be done with the use of parametric methods such as autoregressive (AR) models with the added benefit of better spectral resolution than non-parametric methods such as the FFT.

2.5.4 Classification

The mean heart rate and mean RR interval features show a decrease during periods of stress, as seen in Figure 2.11.

This is consistent with reports by [93,131] indicating a reduction in HRV during stress.

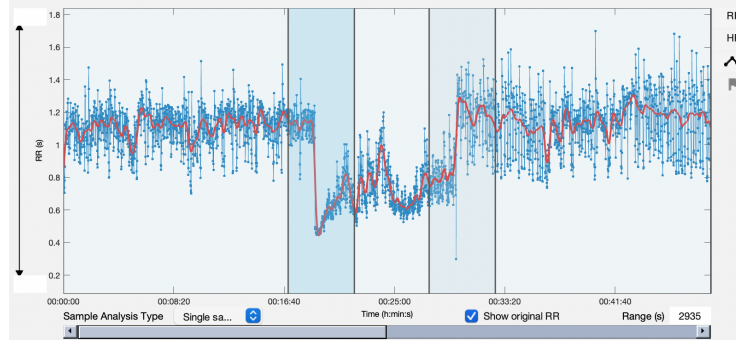


Figure 2.11: RR-interval reduction during periods of stress. (Kubios self-generated image)

This suggests that statistical methods like adaptive thresholding could be useful. However, it is the use of machine-learning models that have gained the attention of researchers [93]. A list of commonly used machine-learning models for stress identification (or relating to ANS activity) is given in Table 2.6.

Table 2.6: Common machine-learning models for stress identification.

MLP: Multilayer Perceptron; **SVM:** Support Vector Machine;

kNN: k-Nearest Neighbour; **LDA:** Linear Discriminant Analysis;

LR: Logistic Regression; **CNN:** Convolution Neural Network;

DT: Decision Tree; **NB:** Naive Bayes

Reference	Model(s)	Application (Comments)	Accuracy [%]
[132]	MLP; SVM;	Mental Stress	94; 88; 94; 94
[97]	kNN (IBk); LDA		
[133]	SVM; LR	Stress (ASD patients)	93; 87
	CNN	Stress (Raw ECG Classification)	90.19
			91.33±1.75;
[134]	LDA; SVM;	Arterial Hypertension (mean HR	86.73±2.24;
	DT; NB	was found most dominant for	87.34±3.08;
		classifiers over 90% score)	88.17±1.07
[135]	ANN (MLP)	Atrial Fibrillation (wearable devices)	98.3

Multilayer perceptron (MLP) models often outperform other models and have the additional benefit of inexpensive execution at $O(n)$. However, creating these models is complex and requires hyperparameter tuning large datasets and time to converge. They are also sensitive to feature scaling, which makes them less commonly used than algorithms like support vector machines (SVM). SVM is easier to employ with fewer tuning parameters, and the algorithm finds correlations between data automatically but with a higher execution complexity of $O(n^2)$.

2.5.5 Evaluations and datasets

With heartbeat detection being at the forefront of most heart-related analyses the use of standardised evaluations and datasets becomes very important. Some of these datasets and evaluation measurements are described here.

- **Massachusetts Institute of Technology - Beth Israel Hospital Arrhythmia (MIT-BIH) database:** The database [3] was initially distributed for the evaluation of heart arrhythmias in 1980. It contains 48 half-hour excerpts of two-channel ambulatory ECG recordings, obtained from 47 subjects (25 male, aged 32 to 89; 22 female, aged 23 to 89). It was sampled at 360Hz with an 11-bit resolution over a 10mV^5 range. It contains over 110,000 annotations of various beat and non-beat orientations [3, 136] and is publicly available on PhysioNet [137].
- **The American Heart Association (AHA) database:** The AHA alongside the National Heart, Lung, and Blood Institute (NHLBI) funded the database [138] for the evaluation of ventricular arrhythmia detectors, during the late 1970s and early 1980s. It consists of 80 two-channel excerpts of analogue ambulatory ECG recording, sampled at 250 Hz with a 12-bit resolution over a 10mV range. It contains 30-minute beat-to-beat annotated recordings with an extra two-and-a-half-hour un-annotated section of each in the long versions. It is freely available from ECRI [138].

The above-mentioned databases are standardised for the detection of arrhythmias and are often employed in QRS/R-peak detection studies [128, 129, 139]. There are, however, no standardised datasets for stress evaluation based on ECG-heartbeat measurements. The following publically available datasets provide ECG measurements for stress and emotional response but are not standardised:

- **Wearable Stress and Affect Detection (WESAD) dataset:** WESAD [4] is a multimodal dataset for the detection of stress and affect. It consists of 15 subjects equipped with a RespiBAN chest monitor for the detection of ECG, EDA, EMG, temperature, respiration, and acceleration; as well as an Empatica E4 wrist monitor for the detection of acceleration, BVP, EDA, and temperature of the subject. The data was collected over 4 scenarios: baseline (subject reading a neutral magazine); amusement (watching funny video clips); stress (based on the Trier Social Stress Test consisting of public speaking and a mental arithmetic task); and lastly, meditation. The annotations are compiled using four self-reports: PANAS; SAM in Valence-Arousal space; State-Trait Anxiety Inventory (STAI); and Short Stress State Questionnaire (SSSQ) [4].
- **Multimodal dataset for Affect Recognition and Implicit Tagging (MAHNOB-HCI):** This multimodal dataset [140] utilises face videos, audio signals, eye gaze, and physiological signals (EDA, ECG, respiration) to assess emotional response in individuals watching 20 emotional videos. It consists of 27 subjects and contains self-reports in valence, arousal, dominance, predictability, and emotional keywords for evaluation [140].
- **Stress Recognition in Automobile Drivers (DriveDB):** The dataset driveDB2005 is created by equipping 17 subjects with ECG, EMG, GSR, and respiratory sensors, taking measurements while they are driving. The aim hereof is to conduct a mul-

⁵mV - millivolt

timodal approach to stress recognition. The data is collected over 24 trips of at least 50 minutes in duration, over three scenarios: rest, highway, and city driving, to distinguish three levels of driver stress. The stress levels were evaluated using independent coders from video footage [141].

The evaluation metrics typically used for beat detection and classification algorithms are sensitivity, specificity, positive predictive value (PPV), negative predictive value (NPV), and accuracy [124, 128]. All of these metrics use the same criteria namely, true positive (TP), false positive (FP), true negative (TN), and false negative (FN). TP is the predicted values correctly identified as true values, FP is the predicted values incorrectly identified as true values, TN is the predicted values correctly identified as false values, and FN is the predicted values incorrectly identified as false values.

Sensitivity, also known as true positive rate, is the ratio of actual positives identified correctly.

$$\text{Sensitivity} = \frac{TP}{TP + FN} \quad (2.3)$$

Specificity, also known as true negative rate, is the ratio of actual negatives identified correctly and is inversely related to sensitivity.

$$\text{Specificity} = \frac{TN}{FP + TN} \quad (2.4)$$

PPV, also known as precision, is the ratio of correctly identified positive values to the total predicted positive values.

$$PPV = \frac{TP}{TP + FP} \quad (2.5)$$

Similarly, NPV is the ratio of correctly identified negative values to the total predicted negative values.

$$NPV = \frac{TN}{TN + FN} \quad (2.6)$$

Accuracy is the correctly identified observations divided by the total number of observations.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \quad (2.7)$$

The above measurements can be used to evaluate both beat detection and stress classification. There are, however, additional measurements often employed to describe the

performance of a classification algorithm, namely f-score, Area Under the Curve (AUC), and k-fold cross-validation [142]. The f-score is the weighted average of the sensitivity and PPV.

$$F - score = \frac{2 \times PPV \times sensitivity}{PPV + sensitivity} \quad (2.8)$$

The AUC measures how well an algorithm can distinguish between classes, e.g. Class 0 against Class 1. It is a probability measurement ranging from 0 to 1, where 1 represents completely distinguishable, 0 means the classes are reversed, and 0.5 means not distinguishable at all. The AUC is the area under the receiver operating characteristics (ROC) curve. The ROC curve is plotted with TPR against FPR. Where TPR (True Positive Rate) is sensitivity as described above, and FPR (False Positive Rate) is:

$$\begin{aligned} FPR &= 1 - specificity \\ FPR &= \frac{FP}{TN + FP} \end{aligned} \quad (2.9)$$

Lastly, k-fold cross-validation measures the accuracy of a classification algorithm, as seen above, by splitting the dataset into k-1 number of subsets that are used for training and 1 used for testing. Each subset is used for testing once, meaning the evaluation is carried out k-times. This evaluation technique is known to work well for limited datasets and offers a less biased evaluation.

2.5.6 Correlation tools

There exist various tools that can analyse HRV data, such as Kubios [98], Elite HRV [143], and Polar SDK [144].

Kubios is the gold standard when it comes to scientific HRV research software and has been cited in over 5900 scientific publications during the past 20 years. It allows for the analyses of not only HRV data but also of extracting HRV data from raw ECG, heart rate, and PPG signals. It includes pre-processing steps (e.g. detrending, noise reduction), beat detection, HRV analyses (time-domain, frequency-domain, and non-linear), and an interactive UI for researchers. Figure 2.12 depicts a typical view of the Kubios HRV software.

Elite HRV is a recently developed HRV analysis software package that has grown in popularity over the past few years [145]. Much like Kubios, the professional versions of Elite HRV allow data capturing, HRV analyses and inter-subject and intra-subject comparisons. Elite HRV, however, does not allow for ECG analyses as it only receives the RR intervals from sensing devices.

Lastly, the Polar SDK [144] is an open-source SDK (software development kit) that allows

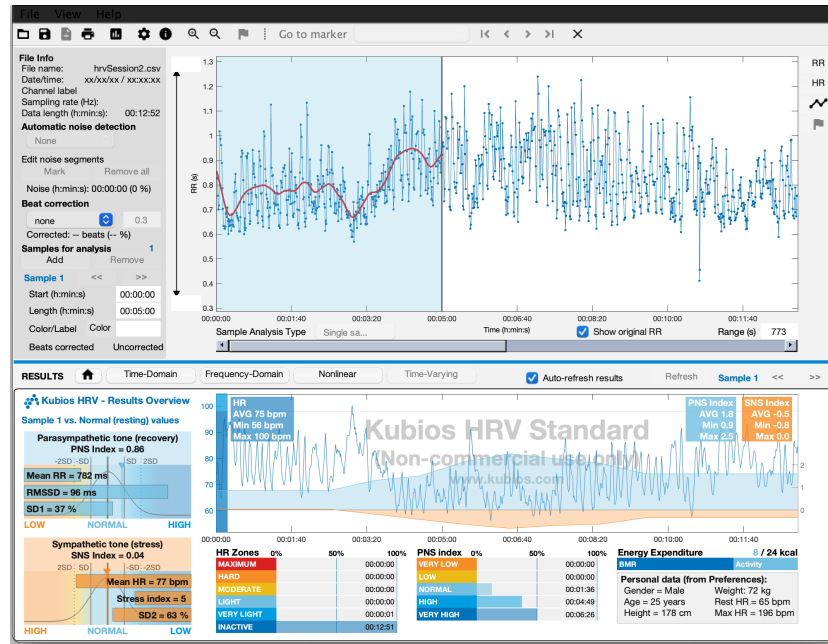


Figure 2.12: Kubios Overview. (Kubios self-generated image)

for the creation of applications that can communicate with Polar devices (e.g. Polar H10, Polar OH). This provides the opportunity to access raw signals made available by these devices. For instance, the Polar H10 exposes ECG and accelerometer data in addition to the HR and RR interval data. However, the Polar SDK does not provide any ECG or HRV analyses or processing algorithms.

2.6 Filling in the gaps: Movement Identification

Physical activity and posture are both factors that can affect HRV [73,92,95]. Accelerometers can be used to determine movement and body orientation [146,147]. Accelerometers detect acceleration by measuring the impact of gravitational forces on a piezoelectric. Tri-axial accelerometers can detect the effect of gravity on three axes: the x-axis, the y-axis, and the z-axis. Along with acceleration, the gravitational force on an axis can help determine the angle of a plane (body orientation). The process for identifying movement using accelerometers is provided in this subsection. Comprising of pre-processing, feature extraction, classification, and evaluation.

2.6.1 Pre-processing

Accelerometers are sensors that can measure acceleration or gravitational force. However, these sensors are not immune to noise, which can interfere with the accuracy of the readings. Power line interference and baseline wandering are among the sources of noise that can affect the signals from accelerometers.

DC offset caused by gravity can be either useful information or noise, depending on the application. For instance, knowing the orientation of the body can help calculate body posture, which can influence heart rate variability (HRV). When someone is seated, their

HR is more relaxed, and their HRV is increased while standing has the opposite effect. On the other hand, body movement caused by physical activity can also affect HRV, due to the amount of energy expended. This measurement, on the other hand, should be determined apart from the effects of gravity for accurate measurement [148].

The DC gravitational frequency range in accelerometer signals is still up for debate, however, it mainly ranges from 0.1 Hz to 0.5 Hz [149]. Authors of the HAPT dataset [5], found the use of a low-order Butterworth low-pass filter with a cut-off at 0.3 Hz satisfactory for the extraction of gravitational data from the accelerometer signal. Body movement on the other hand is expected to range from 0 to 15 Hz [150]. The cut-off frequency for body movement is dependent on the application. For instance, in high-professional sports, a higher sampling frequency and cut-off frequency (nearer to 15 Hz) are preferred, while slower-paced activities can make use of a lower sampling rate and cut-off frequency. For instance [151], utilised a 0 to 3 Hz window for the sole inspection of walking since there are fewer high-frequency variations in the accelerometer signal. The use of a simple band-pass filter can aid in isolating the body movement information from the accelerometer data.

2.6.2 Feature extraction

Within accelerometer data, features are typically extracted from the following derived signals: body acceleration, gravity, Euclidean magnitude, and time derivative [146]. All these signals are calculated from the absolute sum of the triaxial (X, Y, Z) acceleration signals.

Anguita et al. [146], utilises the above-mentioned signals with the variables illustrated in Table 2.7 to extract features for their classification of human activities. They obtained a total of 541 features and an accuracy of 96% using an SVM classifier for the classification of six human activities; namely walking, walking upstairs, walking downstairs, sitting, standing, and lying down.

These variables determine the intensity and distribution of the movement extracted from an accelerometer. With this vast set of variables, high accuracies can be obtained even when distinguishing between different movements. To solely distinguish between movement and non-movement, these variables can be simplified to determine the energy present in the accelerometer data. This simplification makes statistical classification a feasible option.

2.6.3 Classification

For multi-classification tasks, SVM is typically used, and it is no different for human activity recognition (HAR). As noted above [146], utilises SVM for their HAR and obtained an accuracy of 96%. The use of SVM models stems from their ease of implementation and success with small datasets. However, SVM does tend to suffer when the number of features for each data point exceeds the number of training data samples. It also has a computational complexity of $O(n^2)$ making it expensive when working with large datasets.

Table 2.7: Common variables utilised as human activity recognition features [146].

Function	Description
Mean	Mean value
Std	Standard deviation
Mad	Median absolute value
Max	Largest value in array
Min	Smallest value in array
SMA	Signal magnitude area
Energy	Average sum of the squares
IQR	Interquartile range
Entropy	Signal entropy
arCoeff	Autoregression coefficients
Correlation	Correlation coefficients
maxFreqInd	Largest frequency component
meanFreq	Frequency signal weighted average
Skewness	Frequency signal Skewness
Kurtosis	Frequency signal Kurtosis
energyBand	Energy of a frequency band
Angle	Angle between two vectors

The mere identification of movement allows for simple statistical methods, such as thresholding. An illustration of body movement based on accelerometer data was created and is shown in Figure 2.13. Here it is clear that the magnitude of the summed body acceleration can indicate movement.

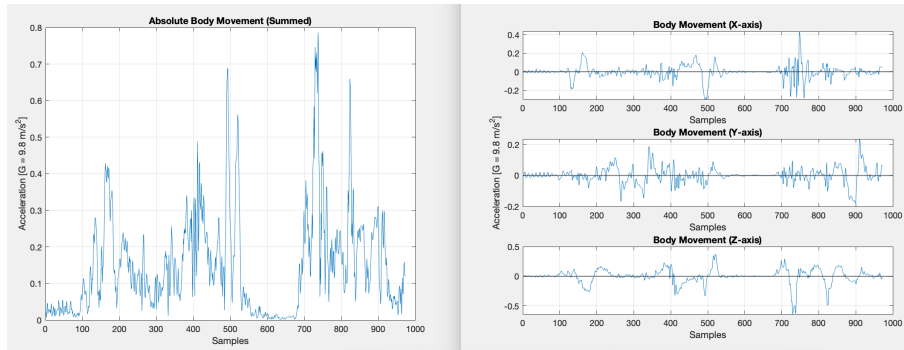


Figure 2.13: Accelerometer body movement illustration.

2.6.4 Evaluation and datasets

When it comes to human activity recognition (HAR) there are few standardised datasets. One of the best-known is the Human Activities and Postural Transitions (HAPT) dataset [5]. Which not only allows for the detection of movement but postural transitions as well. The dataset is made publicly available by the UCI Machine Learning Repository [5]. It comprises triaxial linear acceleration and triaxial angular velocity captured from a waist-worn smartphone at a constant rate of 50 Hz. It provides the raw signals as well as the train and test feature sets.

The evaluation metrics for movement classification are the same as those identified in

section 2.5.5, ‘Evaluations and datasets’, on page 34 above.

2.7 Summary

This chapter reviewed the literature required to address the shortage of SLPs in South Africa and amend their sometimes overbearing workloads. It establishes the importance of identifying individual behaviours and determining ‘who spoke when’ for speech interaction analyses. Furthermore, it indicates the shortfalls of current speech interaction analysis techniques and tools, leading to the investigation of an automated speech and behavioural analysis tool.

To identify ‘who spoke when’ the algorithm speaker diarisation is utilised. Traditional speaker diarisation systems consist of a front-end processing module (e.g., speech enhancement), SAD module (distinguishing speech from non-speech), segmentation module (uniform speech segmentation), speaker embedding module (speaker representation such as GMMs), clustering module (grouping similar speech segments for example via BIC), and lastly, a post-processing module (refinement of speaker cluster, for example, smaller segmentation windows). Typical features include MFCCs and long-term prosody variables to represent speaker-specific speech. Speaker diarisation algorithms are typically evaluated using the DER metric and exclude the presence of overlapping speech. There are also various standardised datasets developed to evaluate an algorithm within a specific environment (e.g., conference meetings, telecommunications, and ‘in the wild’). Obstacles speaker diarisation algorithms still face are the presence of overlapping speech, online processing, and the need for domain-specific datasets.

Behavioural analysis entails identifying acts or conducts an individual might perform that would negatively influence their speech interactions with others. Anxiety has been found to inhibit parent-toddler speech interactions [9]. Anxiety has also been linked to heart rate variations [88] thereby providing a means of identifying it through the use of ECGs. The heart rate of an individual increases in the presence of anxiety and decreases again once the individual is no longer anxious [87]. HRV variables are often employed to characterise the presence of anxiety. These variables include, among others, RMSSD, pNN50, LF, and HF. Besides anxiety, various factors can influence heart rate and thereby HRV, among which are the individual’s age, respiration, gender, general health, and physical activities.

The process of identifying stress (anxiety) through HRV using ECG signals involves several steps, including pre-processing, QRS-complex detection (beat detection), feature extraction, and classification. Pre-processing entails removing noise such as power line interferences and baseline wandering. It is often accomplished by utilising SWTs to decompose the signal and extract the desired frequency bands for signal reconstruction.

QRS-complexes are identified in ECG signals to detect heartbeats, and algorithms used to achieve this include the Pan-Tompkins algorithm [152], and variations thereof such as a technique developed by Sabherwal et al. [128]. These algorithms are based on adaptive zero-crossing and thresholding methods to detect QRS complexes. Various features are

then extracted from the heart rate and HRV data including mean heart rate, mean RR-interval, RMSSD, pNN50, and HF power.

To identify stress, classification methods such as MLP [132,135] and SVM models [97,132,134] are commonly used. There also exists various evaluations and datasets for both beat detection and stress identification. Some of these datasets include the standardised MIT-BIH arrhythmia dataset [3] for beat detection and the WESAD dataset [4] which contains ECG signals for stress identification. Evaluation metrics include sensitivity, specificity, and accuracy. There are also publically available tools such as Kubios [98] which accept HRV data to extract HRV variables (e.g., RMSSD) and determine stress. These tools can be used as correlation tools to verify accurate stress identifications.

Physical activity and posture are both factors that affect HRV [73,92,95]. One method to obtain body movement and orientation is through the use of accelerometers [146,147]. The process of identifying movement using accelerometers consists of a pre-processing, feature extraction, and classification step. The evaluation metrics are the same as that of the stress identification. The pre-processing step entails removing noise such as power line interference and baseline wandering. As well as extracting gravitational and body movement signals. Simple features such as energy can be used to identify movement, while a range of features such as mean, standard deviation, energy, entropy, and maximum and minimum values are extracted to classify movement [146]. Statistical thresholding methods might be sufficient for identifying movement, however, machine learning approaches are often applied when classifying movement such as SVM [146]. One of the best-known datasets for human activity recognition is the HAPT dataset [5].

Chapter 3

System Design and Implementation

This chapter provides technical insights into the fulfilment of this study.

3.1 System Architecture

This study consists of two primary systems: a recorder application and an analyser application designed to automate speech interaction analyses for SLPs. Each system possesses two subsystems, one for speech analysis and the other for behavioural analysis, providing a unique solution of analysing both speech and behavioural data in a single package. This is in contrast to other solutions (see Section 2.2) that provide one or the other. Figure 3.1 illustrates the architecture of this study. With Figure 3.2 providing the outline of the recorder's system design, and Figure 3.3 and Figure 3.4 that of the analyser's speech and behavioural analyses, respectively.

This study's architecture allows for not only independent development and execution of the two primary systems (the recorder and analyser applications) but also that of the subsystems (the speech subsystem separate from the behavioural subsystem). This provides SLPs with the opportunity to conduct new sessions with the recorder application while the analyser application is being executed on a previous session in the background. Furthermore, the independent nature of the subsystems allows for parallel executions of the data collection as well as the analyses thereof.

The remainder of this chapter consists of the development of the recorder application (Section 3.2), and the analyser application which consists of the speaker diarisation system (Section 3.3), the stress identification system (Section 3.4), and the movement identification system (Section 3.5). It also contains the development of the user interface for both the recorder application and the analyser application (Section 3.6), the software and

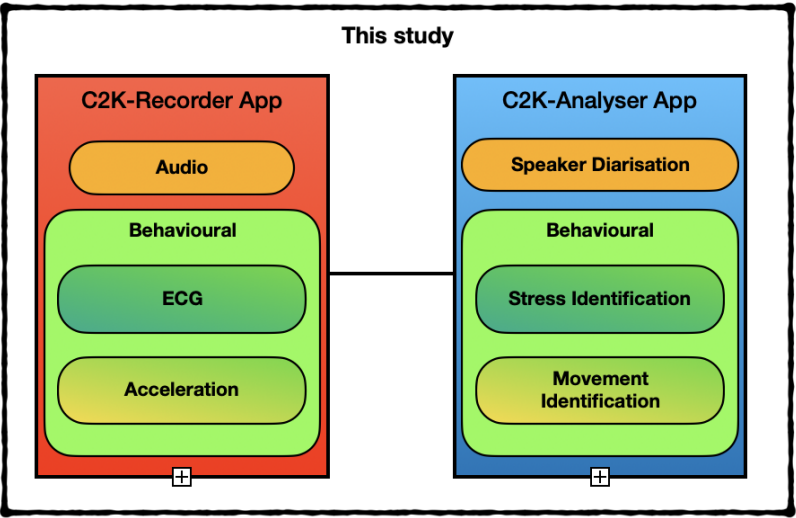


Figure 3.1: System architecture. 1) Speech systems illustrated in orange, 2) behavioural systems illustrated in green.

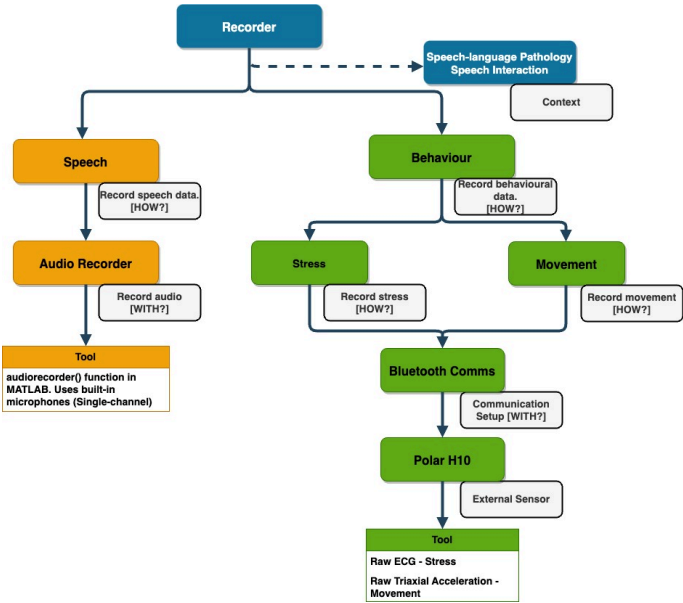


Figure 3.2: System design outline - Recorder

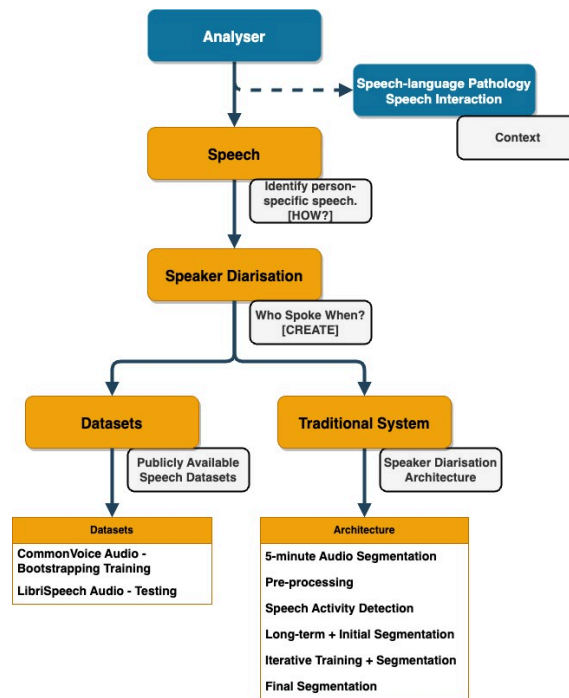


Figure 3.3: System design outline - Speech analysis (Analyser)

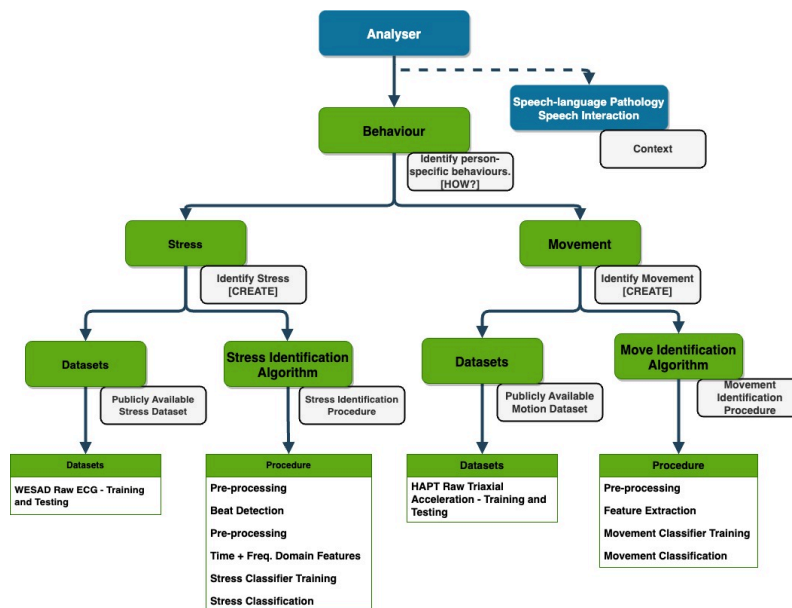


Figure 3.4: System design outline - Behavioural analysis (Analyser)

tools used for the creation of this study (Section 3.7), and the protocols developed for the integration between the recorder and the analyser application (Section 3.8). A description of the scalability and efficiency (Section 3.9) is provided, along with the challenges faced during the development of the recorder and analyser applications (Section 3.10).

3.2 Recorder: Data Collector

A distinct application has been developed for data collection. Its main purpose is to capture and store audio and behavioural data such as ECG and acceleration during speech-language sessions to be used later by the analyser application.

The application is equipped with a central microphone and a Bluetooth Low Energy (BLE) module that communicates with various Polar H10 devices simultaneously. This wireless approach ensures the collection of audio and behavioural data without placing any restrictions on an individual or limiting their actions during a speech-language session.

3.2.1 Audio

Audio data is captured by utilising the recorder application's host device's microphone. MATLAB possesses two means for capturing audio: the `audiorecorder()` object or `audioDeviceReader()` system object.

These methods have different architectures. The `audiorecorder()` object holds the entire audio signal in memory until it is saved, while the `audioDeviceReader()` is a system object that is specifically designed for processing real-time data and modelling dynamic systems. It is memory efficient as it offloads audio frames to storage and retrieves them as needed.

For an hour-long mono-channel audio recording at a sampling rate of 16 kHz and a resolution of 16 bits, the `audiorecorder()` object would use approximately 1 GB of memory. However, it offers built-in play, pause, resume, and stop functions that make it easy to create and test the recordings.

As an initial system, the benefits of the `audiorecorder()` object exceed those of the `audioDeviceReader()` system object. The `audiorecorder()` object is easier to implement and memory constraints are not an implication for testing. However, for practical applications in the future, the `audioDeviceReader()` system object might be better suited.

3.2.2 Behavioural

The Polar H10 device is well-suited for recording behavioural data in the described environment (see Section 1.1 - Speech-language pathologist settings). It provides raw ECG and acceleration data and features a BLE module for communication with the host computer [144]. Renowned as the gold standard for HRV analyses in physical extensive studies [114, 115], the Polar H10 is a dependable choice for collecting behavioural data in this study. Additionally, the device is widely recognized, easily accessible, and has various

chest strap sizes for user comfort.

The Polar H10 is equipped with the BLE Generic Attribute Profile (GATT) Heart Rate Service [153] and holds information on the intended body location of the heart rate monitor, heart rate values, and RR interval values. However, this study will focus on the (non-registered) Polar Measurement Data (PMD) service, which possesses the raw ECG and accelerometer data. Appendix A provides the necessary PMD GATT services and characteristics, as well as the measurement types, frame types, and setting types of the Polar H10, required to obtain ECG and accelerometer data.

An overview of the behavioural data collection procedure via the Polar H10 device is illustrated in Figure 3.5. It consists of among others a prerequisite stage, a start stream stage, and a stop stream stage all of which are required to collect behavioural data from the Polar H10 device(s). Graphical illustrations of these stages and their commands are also provided in Appendix A. It also contains special functions to store real-time data in binary files while the Polar H10 device is streaming data.

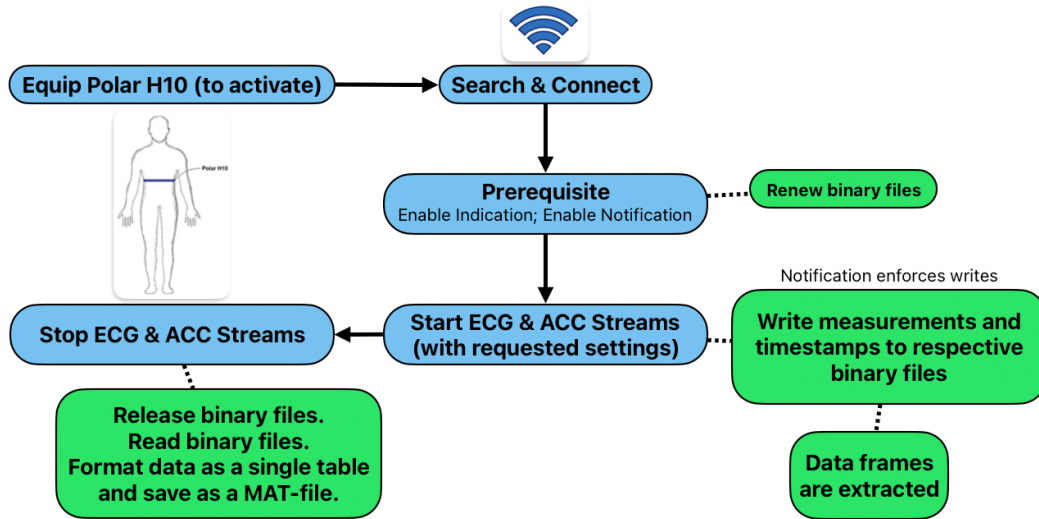


Figure 3.5: Behaviour data collection procedure.

The **prerequisite** stage establishes a BLE connection between the host device and the Polar H10 device(s). It also illustrates the activation of the indication and notification features of the PMD Control Point and PMD Data MTU, respectively. By enabling these features communication between the host device and the Polar H10 device(s) can commence.

To **start** streaming a command is given to the Polar H10 devices' PMD Control Point to start the transmitting process for a given measurement type: ECG or acceleration.

With the streaming process started the data is continuously being written to binary files situated on the host device. This allows for the data to be **stored** for later processing. Before the data can be stored it first undergoes an extraction and conversion process. The Polar H10 device transmits hexadecimal (decimal when using MATLAB) data frames

containing among others the measurement type data. Once the data frame is obtained the measurement type bytes, as well as those corresponding to the timestamp, are extracted and then converted to their represented values. An illustration of an ECG and acceleration data frame is provided in Figure 3.6 and Figure 3.7, respectively. The storing process occurs every time a new notification or indication is received.

ECG:

[00] [109 151 229 198 166 67 82 08] [00] [237 14 00] [225 14 00] [203 14 00]

- Measurement Type -> 0x00 = ECG
- Timestamp for last sample in the frame = 5,996160839742647e17 nanoseconds from '2000-01-01 epoch
- Frame Setting -> 0x00 = Type 0 = 3Bytes
- Samples: sample0 = 3821, sample1 = 3809, sample2 = 3787 uV

Figure 3.6: PMD data frame: ECG

Acceleration:

[02] [109 151 229 198 166 67 82 08] [01] [57 252 31 00 232 00] [63 252 34 00 236 00]

- Measurement Type -> 0x02 = ACC
- Timestamp for last sample in the frame = 5,996160839742647e17 nanoseconds from '2000-01-01 epoch
- Frame Setting -> 0x01 = Type 1 = 6Bytes (2Bytes per Axis)
- Samples -> sample0: [57 252] x-axis = -967, [31 00] y-axis = 31, [232 00] z-axis = 232 mG

Figure 3.7: PMD data frame: Acceleration

To **stop** the recording process a command is once again sent to the PMD Control Point, this time with the stop command. The stop command has to be sent for each measurement type that should stop streaming.

3.3 Analyser: Speaker Diarisation

The speaker diarisation system forms one of the three analyser submodules. This system identifies who spoke when providing the ability to assess an individual's speech visually and audibly separate from the rest of the audio file. It also allows the calculation of person-specific speech metrics and the evaluation of an individual's behaviour during speaking turns. A flow diagram representation of the speaker diarisation system is provided in Figure 3.8. A visual illustration of the system's output is provided in Figure 3.9.

This study utilises a traditional speaker diarisation system architecture, which was shown in Figure 2.2. This architecture allows for submodule optimisation and is not reliant on large domain-specific datasets such as end-to-end neural diarisation (EEND) systems. The speaker diarisation system developed for this study is based on the ICSI RT-09 system [36]. This system allows the execution of the speaker diarisation system without prior knowledge of the speakers or the number of speakers present. This provides a key advantage of not requiring the SLP to conduct prior person-specific speech recordings before their session can commence.

The different modules of the traditional system, such as 'Front-end Processing', 'Segmentation', 'Speaker Embedding', 'Speech Activity Detection (SAD)', 'Clustering', and

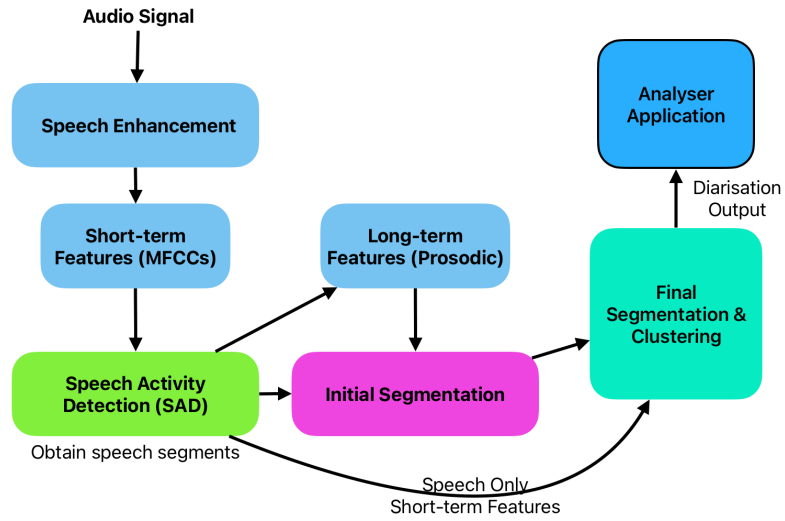


Figure 3.8: Speaker diarisation system - Flow diagram.

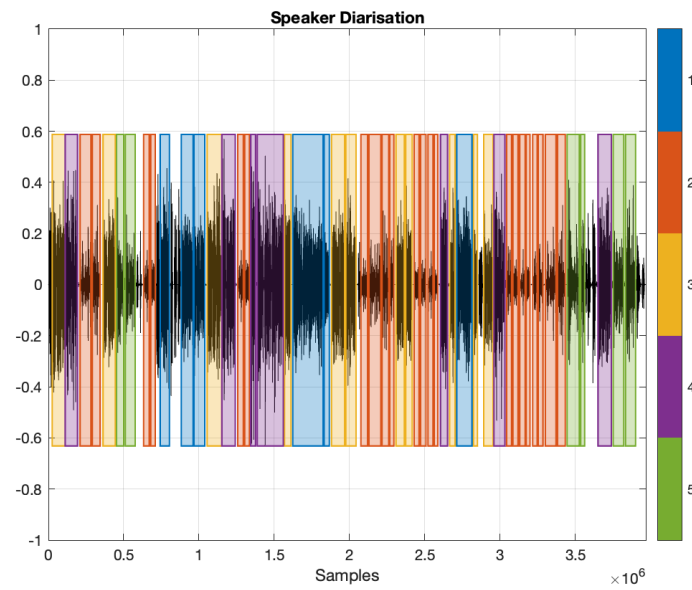


Figure 3.9: Speaker diarisation output example

‘Post-processing’, are addressed in ‘Pre-processing’, ‘Feature extraction’, and ‘Classification’. With the utilisation of the speaker diarisation system speech measurements can be calculated (see Section 3.3.4). A pseudocode representation of the system is also provided in Appendix C.

3.3.1 Pre-processing

As described in Section 2.3.1, speech signals are susceptible to various forms of noise, such as additive noise, reverberations, and overlapping speech. For this study, the sole focus is mitigating additive noise through speech enhancement techniques. Reverberations and overlapping speech are beyond the scope of this study.

Typical speech enhancement methods include spectral subtraction, subspace, and statistical-model-based methods. Among the statistical-model-based methods, the Wiener filter and minimum mean square error (MMSE) are among the most popular. The ICSI RT-09 speaker diarisation system [36] - which this study’s speaker diarisation system mainly resembles - uses a Wiener filter to mitigate the effects of noise. However, [34] found that the log-MMSE algorithm for speech enhancement resulted in less speech distortion and better speech intelligibility compared to the Wiener filter, as well as the spectral subtraction and subspace algorithms. For this reason, the log-MMSE algorithm is used for speech enhancement in this study.

The objective of MMSE methods is to obtain an estimate of the true signal, given that it is corrupted by noise or other imperfections. The speech enhancement algorithm, log-MMSE, estimates the true speech signal by minimising the mean square error of the log-magnitude spectra given a noisy non-speech observation [34]. The mathematical representation of the algorithm is illustrated by Equation 3.1,

$$\hat{X} = \exp(E(\log(X)|Y)), \quad (3.1)$$

where $\hat{X}$ represents the estimated signal, X the true signal, Y the observation, and E the MMSE algorithm [34].

The execution of the algorithm was made possible by the `v.ssubmmse()` function created by VOICEBOX [154], upheld and written by Mike Brookes, Department of Electrical and Electronic Engineering, Imperial College. The software package, VOICEBOX, is a speech processing toolbox meant for MATLAB and is publicly available under the GNU Public license.

Furthermore, the audio data is normalised to fall between $[-1; 1]$, then subjected to a non-linear function, and finally augmented with additive white noise. The normalisation equation and non-linear functions are illustrated in Equation 3.2 and Equation 3.3, respectively.

$$\text{norm} = \frac{\text{Sig}}{\max(|\text{Sig}|)} \quad (3.2)$$

$$\text{newSig} = \text{sign}(\text{Sig}) \times (|\text{Sig}|^{0.75}) \quad (3.3)$$

The normalisation equation scales the audio data between the values of -1 and 1 . The non-linear function helps reduce the impact of speakers who are far away. Lastly, the additive white noise maintains a signal-to-noise ratio (SNR) of 120 dB, which is just enough noise to ensure that silence segments have proper zero-crossing rates without distorting the speech segments.

These speech enhancement algorithms are applied to 5-minute audio segments. The complete audio signal is initially kept in storage (not in memory), and only 5-minute segments are gathered using tall arrays. This technique helps the system to optimise memory usage and enables parallel computation if required. This technique is implemented throughout the speaker diarisation system.

3.3.2 Feature Extraction

There are two categories of features considered for speaker diarisation in this study: short-term and long-term. These features are based on the recommendations of the ICSI RT-09 system [36]. The short-term features include zero-crossing rate and 12-MFCCs (Mel Frequency Cepstrum Coefficients), together with their first and second derivatives, contributing to a 39-dimension feature set. These features are well-established in speech analysis [30, 36, 50] and provide a feature set for both the SAD algorithm and the main clustering algorithm described in Section 3.3.3 below. These features are extracted using the `audioFeatureExtractor()` object in MATLAB. A fixed window size of 30 ms and an overlap of 20 ms (frame period 10 ms) is chosen for these features based on the recommendations of [36, 50].

The long-term features include statistical variables of formants, pitch, and harmonic-to-noise ratio (HNR) [50]. These features provide good speaker discrimination and are solely meant for the initial cluster segmentation stage. This initial segmentation provides an estimate for the number of clusters (speakers), k , as well as the number of Gaussians to be used in each Gaussian Mixture, g .

The long-term features used in this study are summarised by Table 3.1. The min, median, and max statistics are calculated as the 5th, 50th, and 95th percentiles, respectively, to avoid an otherwise large impact of outliers caused by artefacts [50].

The long-term features are extracted from speech segments obtained from the SAD algorithm. These speech segments are further divided into windows $\in [1000 \text{ ms}; 1500 \text{ ms})$ with 50% overlap. This procedure forms part of the initial cluster segmentation described in Sec-

Table 3.1: Speaker diarisation long-term features

Category	Features
Pitch	Median, Min, Mean
HRN	Mean
4th Formant	Std, Min, Mean
5th Formant	Std, Min, Mean
Formant Dispersion	Mean

tion 3.3.3. It differs from that of the ICSI RT-09 speaker diarisation system [36] as the windows in this study are overlapped allowing for augmented data in regions that could identify multiple speakers present in a single speech segment. Moreover, a Hamming window is applied to these overlapping windows. It is from these windows that long-term features are extracted. There are 98 pitch and HNR values and 78 formant values extracted per second. The pitch and HNR features are extracted using the `audioFeatureExtractor()` object within MATLAB, and the formants are calculated using linear predictive coding (LPC), as proposed by [155].

3.3.3 Classification

Here the speech activity detection (SAD) module, initial segmentation module, and core segmentation-and-clustering module of the speaker diarisation system are described.

SAD:

After pre-processing the audio data is sent to the speech activity detection (SAD) module to identify speech segments. The SAD algorithm used in this study is a fusion between the model-based approach used in the ICSI RT-09 speaker diarisation system [36], and an adaptive energy and spectral spread threshold method based on [156].

A flow diagram representation of the adapted SAD module is provided in Figure 3.10.

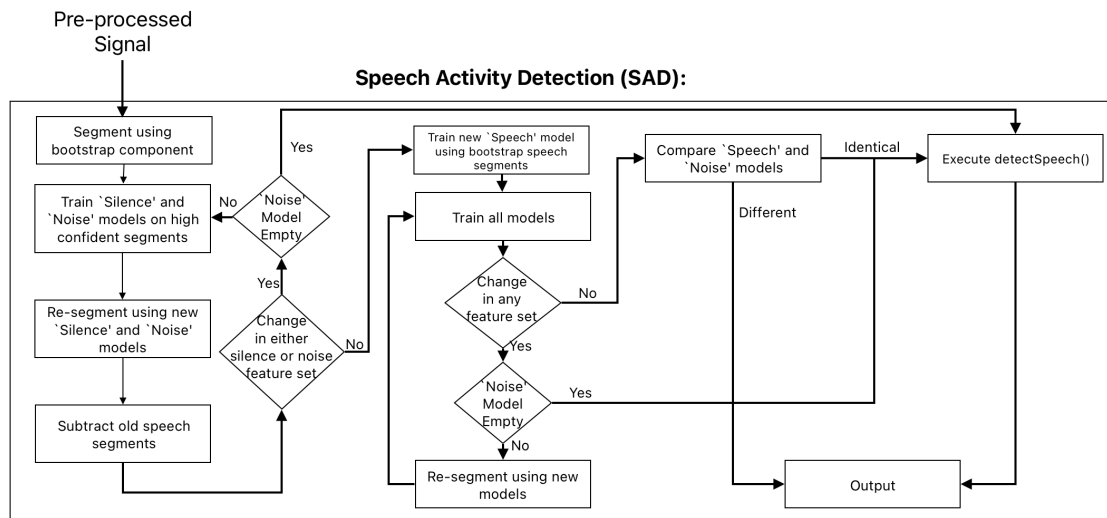


Figure 3.10: SAD flow diagram.

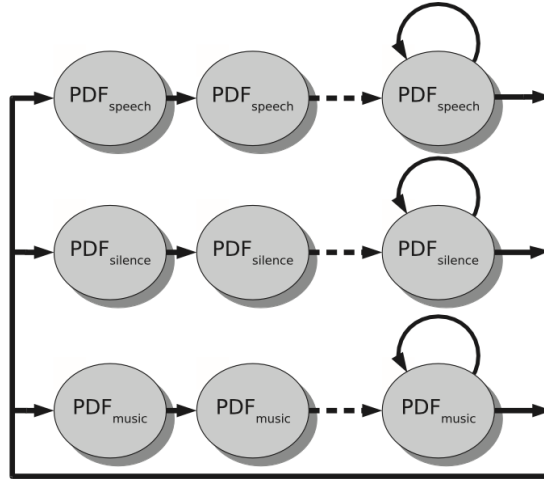


Figure 3.11: Model-based approach - string HMM illustration [35]. The structure contains parallel strings and shared PDFs, and only the last state in the string has a self-transition.

The model-based approach utilised with the Viterbi algorithm (Appendix B) provides a robust SAD solution capable of distinguishing between speech and noise (non-speech sounds) without prior knowledge of the speakers. This approach, however, can be computationally expensive. If there are no noise segments present, a less computationally expensive algorithm can be used. To address this, the system has been adapted to switch to the threshold method when the model-based approach detects the absence of noise in the audio data.

The model-based approach follows a string HMM architecture as illustrated in Figure 3.11. Each string of states (i.e. $\text{PDF}_{\text{speech}}$) represents one segmentation class (i.e., speech), and all states within a string share the same probability density function (PDF). The PDF is represented by Gaussian Mixture Models (GMMs) creating a string HMM-GMM architecture. This architecture together with the Viterbi algorithm is responsible for not only segmenting and clustering speech, silence, and noise segments within the SAD, but also the segmentation and clustering of speakers described in the main clustering module.

The HMM-GMM approach for the SAD module involves four steps: bootstrapping, training non-speech models, training all models (for speech and non-speech), and lastly comparing the noise and speech models (see Figure 3.10).

Step 1: Bootstrapping entails segmenting the audio data into initial speech, silence, and noise segments. This is accomplished by utilising an existing HMM-GMM model equipped with a Gaussian distribution for each class. The speech model was created using a subset of the CommonVoice dataset’s training data and holds 250 male and 250 female speakers of varying ages and accents [1]. The audio was resampled from 48 kHz to 16 kHz using the `dsp.FIRRateConverter()` object in MATLAB, which applies an anti-imaging and anti-aliasing filter before decimation. The threshold method mentioned earlier, based on [156], was also utilised to extract speech from the clips. The silence models were merely created from small white noise signals created using the `awgn()` function in

MATLAB, as well as recording small silent audio segments. The noise model was created by recording various toy sounds, fans, and door knocks. These recordings were conducted in MATLAB with a sampling rate set to 16 kHz, bitrate to 16 bits per sample, and recorded as a single audio channel. Short-term features were extracted from these audio samples and used to create the GMMs. By comparing the Akaike Information Criterion (AIC) values, the speech, silence, and noise models were created to possess 8, 6, and 8 Gaussians, respectively. The GMMs were created using the `fitgmdist()` function in MATLAB and configured to have a diagonal covariance matrix. This function applies an iterative Expectation-Maximisation (EM) algorithm to fit the GMMs to the data thereby optimising the GMM likelihood. What differs from the ICSI RT-09 [36] approach is that the k-means++ clustering algorithm [157] is employed to provide the initial starting values for the GMMs.

The hidden Markov model (HMM), on the other hand, was created to represent the state-flow diagram presented in Figure 3.11. The HMM is set with equal initial probabilities and only allows transitions between classes after a minimum duration. This duration is enforced through the use of the string architecture and has been employed to prevent short segments resulting from rapid class transitions. Speech segments have a minimum duration of 750 ms, while silence and noise segments have a minimum duration of 250 ms each. Lastly, the GMMs representing the classes make out the emission matrices of the HMM-GMM model. The audio data is then segmented based on this HMM-GMM structure with the use of the Viterbi algorithm (Appendix B).

Step 2: After the initial segmentation using the bootstrapping component the silence and noise models are iteratively retrained and the audio re-segmented. From the initial silence and noise segments energy and zero-crossing rate (ZCR) features are extracted. These features together with an adaptive threshold classifier identify the most probable silence and noise segments for re-iterative training of the silence and noise GMMs.

For the first iteration, silence and noise segments are merged under a single non-speech class. This prevents any misclassifications resulting from the initial bootstrapping component. The newly established non-speech audio is then segmented using a Hamming window with the same window size created by the HMM model for silence and noise (250 ms) with a 50% overlap. After extracting energy and ZCR features from these windows, silence, and noise are differentiated by implementing percentile thresholds. For the first iteration, the threshold is strict and only allows segments below the maximum of the first quartile for silence and above the maximum of the third quartile for noise. The audio data is then re-segmented via the Viterbi algorithm and a newly formed HMM-GMM model containing the previous bootstrap speech GMM and the newly retrained silence and noise GMMs as emission matrices.

After the first iteration, the newly created GMMs are believed to fit the audio accurately since they are trained on it. As a result of this and the use of less strict threshold classifiers, the newly identified silence and noise segments are not merged like they were in

the first iteration. Additionally, the newly formed noise GMM may perform better than the primary bootstrap speech GMM in fitting speech segments to its cluster. To address this possibility, the original speech segments identified by the bootstrapping component are removed from the newly segmented silence and noise segments. Energy and ZCR features are once again extracted and used for segment selection. The threshold this time around is less strict and will gradually fall per iteration until the 40th percentile is reached. This means if the energy and ZCR of a noise segment are above the 40% percentile, it is chosen for noise retraining. For silence segments, those with energy and ZCR below the 60% percentile are selected for silence retraining.

Step 3: After the final iteration of retraining and segmenting the silence and noise models, the speech model is retrained. All the identified speech segments longer than the minimum speech duration are used for speech retraining. After the speech GMM is retrained, segmentation occurs completing the first iteration of the third step. Thereafter, all the models are retrained and repeated two more times, ensuring stable segmentation.

Step 4: During the final step, the speech and noise models are compared to determine whether they share data characteristics making the noise model obsolete. This involves training a new model on both the speech and noise segments and using the combined amount of Gaussians found in the speech and noise models. If the delta-BIC in Equation 3.4 is found to be negative then the models were not trained on the same data and the segmentation and models remain as they were. If, however, positive then the noise model must be removed, resulting in there being no identifiable noise segments.

$$\Delta\text{BIC} = \text{BIC}_{\text{Combined}} - (\text{BIC}_{\text{Speech}} + \text{BIC}_{\text{Noise}}) \quad (3.4)$$

The entire process described above entails the model-based partition of the SAD algorithm. If there is, however, identified at any point that the audio does not contain noise then the model-based approach is replaced by the computationally inexpensive threshold function, `detectSpeech()`. This MATLAB function utilises a short-term energy and spectral spread adaptive threshold to distinguish between silence and speech.

Initial Segmentation:

After the SAD has been completed, the initial segmentation process can commence. This process utilises the output from the SAD to generate non-uniform speaker representations of the audio data, which will be used to train GMMs during the main segmentation and clustering phase. The segmentation process provides an initial estimation of the number of speakers and to whom a speech segment belongs.

The initial segmentation process utilises long-term features extracted from the SAD-derived speech segments to create an initial GMM representation of all the speakers present in the derived segments. The model is created by iteratively increasing the number of components used to create the GMM and comparing its Bayesian Information Criterion (BIC)

value to the previous model. If the BIC value does not improve (decrease) then the previous GMM's number of components is taken as an estimate of the number of speakers present, and its Gaussian distributions are used to represent the various speakers. An illustration of the initial segmentation procedure is provided in Figure 3.12. The segmentation of the speech segments in terms of speaker representation is accomplished by establishing which Gaussian distribution obtains the smallest log-likelihood given the segment.

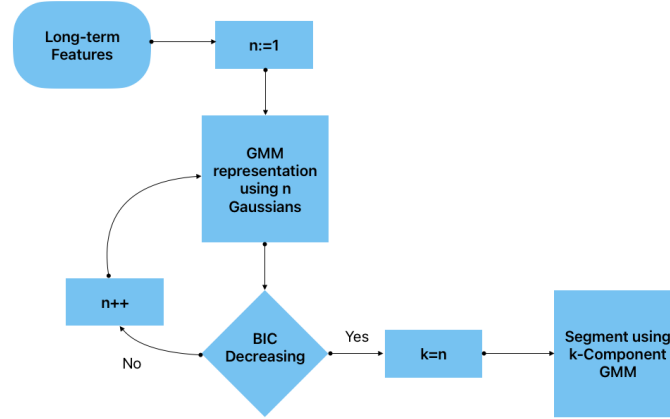


Figure 3.12: Schematic view of the initial segmentation process

Core Algorithm:

The core algorithm involves an iterative segmentation and clustering process which is summarised in the following steps:

- Step 0 - Initial segmentation (as described above).
- Step 1 - Model retraining and segmentation.
- Step 2 - Model merging based on BIC.
- Step 3 - Stopping condition (if no clusters were merged then stop else repeat from step 1).
- Step 4 - Final segmentation and output.

After the initial segmentation, the HMM-GMM model is retrained. The GMM emission probabilities are retrained on short-term features extracted from each speaker segment and the transition matrix is recreated to represent a string of states with a minimum duration of 2.5 seconds (250 samples). This means in order for a speech segment to be used it should possess a minimum duration of 2.5 seconds. This also causes a limitation to this approach because if an individual during the entire recording does not have a speech segment longer than or equal to 2.5 seconds then this individual will be missed entirely. This duration was chosen as shorter segments yielded poor results based on the work of [36] and individual testing. This might pose a problem for toddlers who still speak in two-word phrases. However, individual testing revealed that 2.5-second phrases are not uncommon for a three-year-old toddler when conversing with an adult.

After the speaker models are trained, the speech segments are re-segmented once again using the Viterbi algorithm.

With the newly segmented speech data, the algorithm employs an AHC approach to merge redundant clusters based on the delta-BIC equation (Equation 3.4). The algorithm selects the single most likely cluster pair, merges them, and then repeats the process until the stopping criterion (no delta-BIC values larger than zero) is met.

The entire classification section is repeated for every 5 minutes of audio data. After the 5-minute segments have finished, the segments are merged to make out the entire audio recording. The core algorithm is once again called to compare and merge clusters between the segments. Once the stopping criterion is met again, the speech data is segmented a final time with a smaller minimum duration of 1.5 seconds. As the speakers should be well-defined by now, the smaller minimum duration should allow for fewer missed classification errors resulting from too short speech segments. This final segmentation provides the speaker diarisation output.

3.3.4 Measurements

The speaker diarisation system provides a means of calculating person-specific speech measurements. The measurements considered in this study are:

- **Turn-taking** - the number of spoken turns;
- **Spoken duration** - total speech duration;
- **Longest phrase** - maximum phrase length;
- **Longest turn** - summation of speech within a single turn (without interruptions or long pauses).

These measurements are calculated for each speaker identified by the speaker diarisation system. Where *turn-taking* is taken as the number of times an individual speaks before or after another speaker. *Spoken duration* is the summation of a speaker's speech segments. *Longest phrase* is taken as the maximum speech segment of a speaker. Lastly, *longest turn* is the maximum speech duration within a single turn where the speech segments are not further than 1.8 seconds apart.

3.4 Analyser: Stress Identification

The stress identification system forms the second submodule of the analyser system and is the first of the two behavioural submodules.

The stress identification system is created through the use of ECG-derived HRV measurements. Throughout the development of this system, a sampling rate of 130 Hz is utilised. This is to represent the set sampling rate of the Polar H10 device.

3.4.1 Pre-processing

The pre-processing stage involves processing ECG signals for accurate HRV extraction. The HRV data is then in return processed for feature extraction. This subsection entails detrending and resampling ECG signals, HRV-beat correction, HRV resampling, and lastly detrending the HRV data.

Detrending the ECG signal entails removing low-frequency noise (0.15 - 0.3 Hz) which is usually caused by an individual's respiration. Due to the respiration differences between adults and children, children's being as high as 1 Hz, this study employs a moving average filter with a resulting cut-off frequency of 1.266 Hz. Ensuring the removal of most low-frequency noise without affecting the desired 5 – 25 Hz QRS-complex regions [128]), required for beat detection.

After the ECG signal has been detrended it is resampled to obtain a constant sampling rate of 360 Hz. This ensures proper frequency decomposition of the signal when performing stationary wavelet transforms (SWT).

By employing SWT the ECG signal can be reconstructed to contain merely the desired QRS-complex frequencies, thereby removing any remaining noise such as power line interference (see Figure 3.14). The SWT divides the signal into detail and approximate coefficients (see Table 3.2), for this study merely the fourth and fifth detail coefficients are used for ECG reconstruction.

Table 3.2: ECG SWT decomposition

Detail Coefficient	Frequency range [Hz]
$d_1[n]$	90 – 180
$d_2[n]$	45 – 90
$d_3[n]$	22.5 – 45
$d_4[n]$	11.25 – 22.5
$d_5[n]$	5.625 – 11.25
$d_6[n]$	2.8125 – 5.625
$d_7[n]$	1.40625 – 2.8125
$d_8[n]$	0.703125 – 1.40625
$d_9[n]$	0.3515625 – 0.703125
$d_{10}[n]$	0.17578125 – 0.3515625

The Symlets 4 (Sym4) wavelet was used for representation, preserving the ECG signal while providing adequate noise reduction, as illustrated in Figure 3.13. The overall noise reduction capabilities of the SWT decomposition are illustrated by Figure 3.14. The Daubenchies 6 wavelet (db6) was used in [128] however, the difference in results was negligible.

After the ECG signal has been reconstructed using the inverse-SWT, the signal is enhanced using a non-linear function, illustrated in Equation 3.5. This function enhances the peaks of the signal while mitigating the smaller areas.

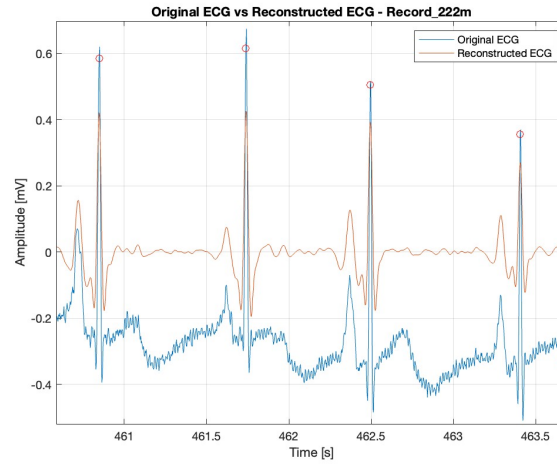


Figure 3.13: SWT noise reduction on MIT-BIH record 222 [3]. (Time-domain)

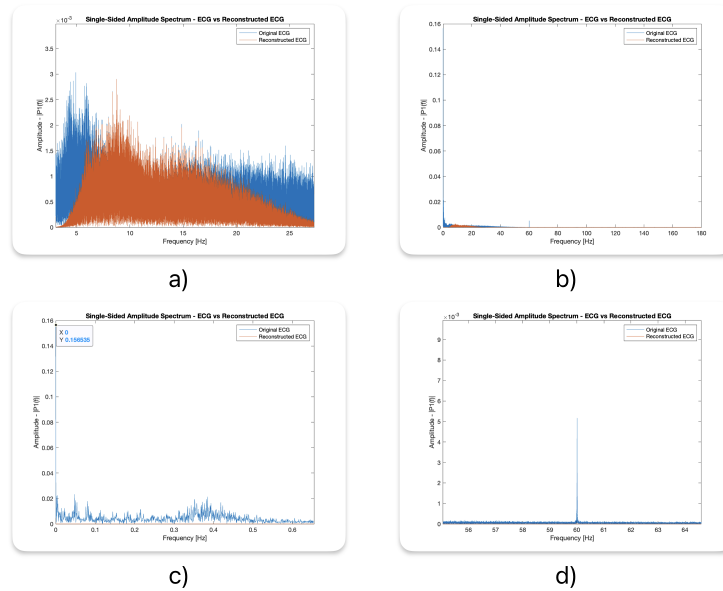


Figure 3.14: SWT noise reduction on MIT-BIH record 222 [3]. (Frequency-domain) a) Reconstructed signal frequency range, b) complete representation, c) Baseline wandering removal, d) power line removal.

$$\text{Enhanced Signal} = \text{sign}(\text{Sig}) \times \text{Sig}^2 \quad (3.5)$$

With the ECG signal properly processed, the **HRV** data can be extracted. This is done by detecting heartbeats through the detection of the QRS-complex and more specifically the R-peaks. This process is further described in Section 3.4.3.

After the HRV has been extracted, it then undergoes a pre-processing phase of its own. The HRV data is tested for outliers by incorporating a modified version of the interquartile range (IQR) outlier formula, as indicated by Equation 3.6 below.

$$\begin{aligned} \text{IQR} &= Q3 - Q1; \\ \text{LB} &= Q1 - (3 \times \text{IQR}); \\ \text{UB} &= Q3 + (3 \times \text{IQR}), \end{aligned} \quad (3.6)$$

where $Q1$ and $Q3$ are the first and third quartiles of the data, respectively, and LB and UB represent the lower and upper bounds, respectively.

This equation is then applied to the HRV data using a moving window approach with a window size of 16 samples. Outliers are replaced with the average of the adjacent samples.

After identifying any outliers, the HRV data undergoes two separate processes - one for time-domain calculations and the other for frequency-domain calculations. For frequency-domain calculations, the HRV data is first resampled to a constant sampling rate of 4 Hz. This ensures that the data is equally spaced and allows for accurate frequency-domain processes. A sampling rate of 4 Hz is ideal for conducting autoregression feature extraction [107], surpasses the Nyquist theorem (the highest frequency expected is approximately 1.04Hz for a child's breathing rate), and appears to be standard when conducting frequency-domain calculations [98, 107, 126].

The resampled data is then sent through a time-varying FIR high-pass filter, which is based on the work of [126]. This filter detrends the data and produces a stationary output. Additionally, the filter indicates the trend line that follows the HRV data, which is later used in the time-domain processing. The filter's single tuneable parameter is set to $\lambda = 500$, ensuring a cut-off frequency of $cf = 0.035$ Hz @ $F_s = 4$ Hz. An illustration of resampled-and-detrended HRV data is provided in Figure 3.15.

In comparison, time-domain computations are generally applied directly to the HRV values as noted in [86, 98]. Moreover, in this study, the HRV data is altered by removing its baseline drift. This is accomplished by taking the mean of the trend line described above and subtracting it from the HRV data. This is done to ensure comparable results, as each individual's base heart rate varies, creating offsets. It is the variation from the baseline

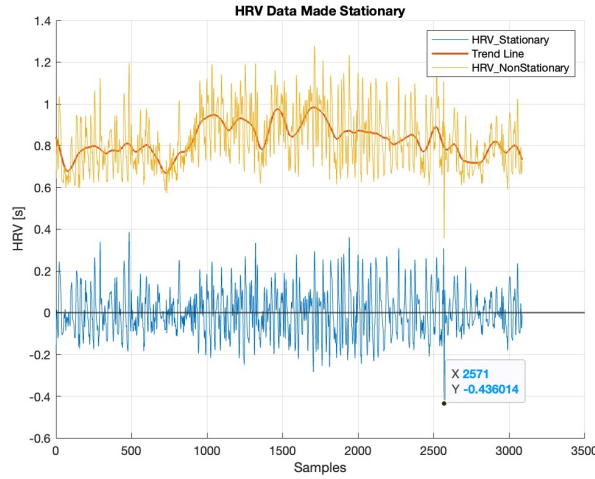


Figure 3.15: Illustration of detrending the HRV data.

that defines stress and non-stress regions. Therefore the features extracted are done so accordingly.

3.4.2 Feature Extraction

The features used for stress identification are divided into two categories: time-domain and frequency-domain HRV features. Moreover, special focus is placed on short-term (5-minute) features with a 75% overlap to identify momentary stress.

The time-domain features include the mean RR intervals, RR-interval skewness, RMSSD, and pNN50. Descriptions of these features are listed below.

- mean_RR - The average of the RR intervals. [ms]
- skewness_RR - The skewness of the RR intervals. [ms]
- RMSSD - The square root of the mean squared differences between successive RR intervals. [ms]
- pNN50 - Number of successive RR interval pairs that differ more than 50 ms (NN50) divided by the total number of RR intervals. [%]

The frequency-domain features comprise the absolute power of the low and high-frequency bands, as well as the ratio of the low to high-frequency bands. The absolute power features are measured in milliseconds-squared. For this study, the frequency domain is obtained using a parametric approach, specifically autoregression using the Burg method, with a model order of 16 (based on the recommendations of [107]). Descriptions of these features are listed below.

- (Absolute power) LF band - Area under the curve between the frequency ranges of (0.04; 0.15] Hz.

- (Absolute power) HF band - Area under the curve between the frequency ranges of (0.15; 0.4] Hz for adults and (0.24; 1.04] Hz for children. The HF band in this study was, however, fixed at (0.15; 0.4] Hz.
- LF/HF ratio - The ratio between the absolute LF and HF power.

A summary of the time-domain and frequency-domain features used in this study and their correlation to stress is provided in Table 3.3. The features are also ranked with the use of the ANOVA algorithm and are depicted in Figure 3.16.

Table 3.3: Summary of the HRV features used and their correlation to stress.

Category	Feature	Stress	Non-stress
Time-domain	mean_RR [ms]	<	>
	RMSSD [ms]	<	>
	pNN50	<	>
	skewness_RR	+	-
Frequency-domain	LF [ms ²]	<	>
	HF[ms ²]	<<	>
	LF/HF	Dependent	Dependent

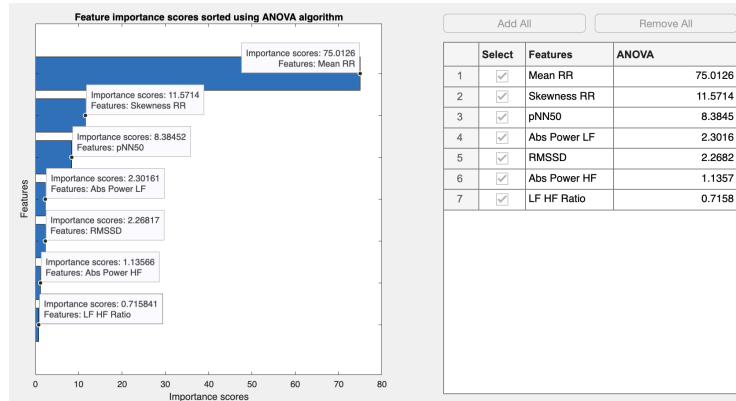


Figure 3.16: ANOVA stress feature ranking.

Based on the observations made in Table 3.3 and Figure 3.16, the *LF/HF ratio* is not a significant stress indicator and could be excluded from the feature set. It is also evident that the feature *mean RR intervals* is primarily responsible for the classification of stress and non-stress conditions. However, this feature's close correlation to age and respiration might affect its reliability, and hence, it cannot be the sole indicator. Therefore, further testing on larger and more age-diverse datasets is required. If stricter feature selection is, however, necessary, then the feature set would be reduced to *mean RR*, *skewness RR*, *pNN50*, *absolute power LF*, and *RMSSD*. The inclusion of *RMSSD* is based on literature where it has shown resistance against respiratory influences [100] however, further research in this regard is still required. The feature set for this study is reduced to the top 5 features illustrated in Figure 3.16, namely, *Mean.RR*, *Skewness.RR*, *pNN50*, *Absolute Power LF*, and *RMSSD*. This reduction allows for a less complex model without sacrificing significant prediction accuracy. The training accuracy dropped from 95.2% to 93.0% with the feature reduction.

3.4.3 Classification

The classification section starts with identifying R-peaks within the ECG signal, allowing for the extraction of HRV data. Thereafter different stress classification algorithms are tested based on the HRV features described in Section 3.4.2 above, providing the desired identification of momentary stress regions.

The beat detector utilised in this study draws inspiration from the Sabherwal et al. (2018) algorithm [128], which achieved impressive results with a 99.990% sensitivity and positive predictive value, coupled with a 0.135% error rate against the MIT-BIH arrhythmia dataset. However, as this study does not focus on arrhythmia detection, and is not intended for individuals diagnosed with the condition, the algorithm has been simplified to detect only the largest peak (either the R-peak or the S-peak) using the zero-crossing threshold method outlined in [128]. Furthermore, to ensure shift-invariance and improved frequency localisation, this study employs SWT instead of DWT. An overview of the beat detection process is provided in Figure 3.17

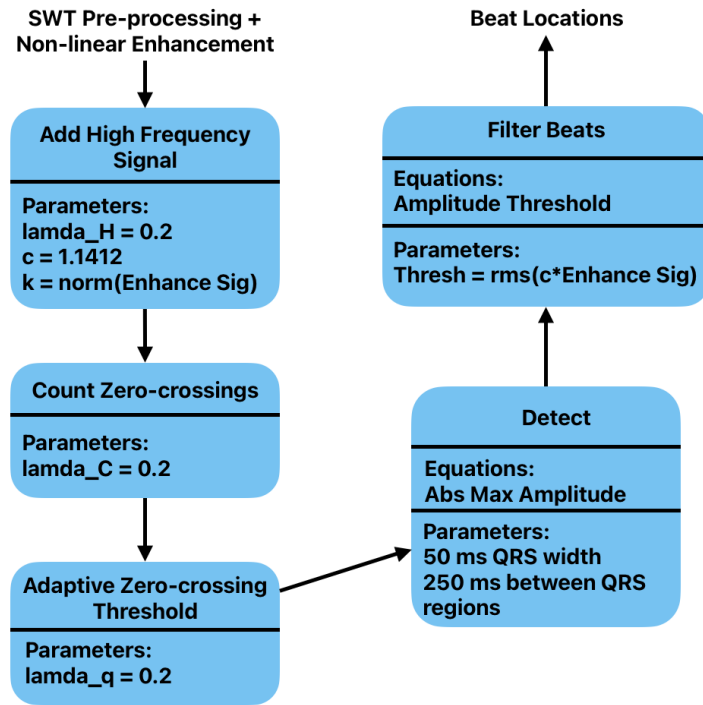


Figure 3.17: Beat detection flow diagram.

The selected parameters for the high-frequency signal are designed to produce high zero-crossing rates during non-QRS regions, and smaller rates during QRS regions. These parameters, along with the zero-crossing counting and adaptive threshold parameters, are carefully chosen as the presence of small non-QRS waves or near-zero regions can significantly alter the zero-crossing values. This is a drawback of this method. Furthermore, an amplitude threshold based on the root mean square value of the enhanced ECG signal is implemented to remove misclassified beats resulting from in-between waves.

The beat detection algorithm provides the location of heartbeats present in an ECG sig-

nal, which helps in calculating HRV (Heart Rate Variability) values. To calculate HRV, consecutive R-peaks are subtracted from one another, and the results are provided in seconds using the formula presented in Equation 3.7. Here, R represents the location of each R-peak, and F_s represents the sampling rate of the ECG signal.

$$\text{HRV} = \frac{R[n] - R[n - 1]}{F_s} \quad (3.7)$$

For stress identification, various classification models were tested, mainly variations of the Support Vector Machine (SVM) classifiers. For the creation of these models, the ‘Classification Learner’ application in MATLAB was employed. A 5-fold cross-validation method was used to protect against overfitting and 10% of the data was set aside for testing. The dataset used was the WESAD dataset [4] containing raw ECG data and stress labels among others. The dataset had to be converted from PKL files to CSV files. This was accomplished via the Python script indicated in Appendix D. After the feature extraction process described in Section 3.4.2 the dataset provided 184 observations (45 stress observations) from 15 subjects with a mean age of 27.5 ± 2.4 years.

The models trained with their results are illustrated in Table 3.4.

Table 3.4: Stress Identification models comparison.

Model	Train Accuracy [%]	Test Accuracy [%]	Prediction Speed [obs/sec]
Tree	93.3	94.4	~9400
SVM: Linear	92.7	94.4	~7100
SVM: Quadratic	93.3	100.0	~2800
SVM: Cubic	93.9	100.0	~3600
SVM: Fine Gaussian	87.3	94.4	~3000
SVM: Medium Gaussian	90.9	100.0	~3100
SVM: Coarse Gaussian	88.5	94.4	~3800

The speed at which predictions are made is not of great importance as the algorithms are designed to run in the background, offline. Additionally, the current feature extraction parameters allow all models to be fast enough, considering an hourly session holds only 45 observations. However, it is worth noting that the testing accuracies may not be entirely reliable as the test subset is limited in both size and the number of stress observations. Therefore, the conclusion will be solely based on the training accuracies.

Based on the training accuracies and the literature, the ‘Cubic SVM’ model is sufficient for this study’s stress identification needs. The model’s classification properties are evident from its ‘Confusion Matrix’ and ‘ROC Curve’ diagrams, shown in Figure 3.18 and Figure 3.19, respectively.

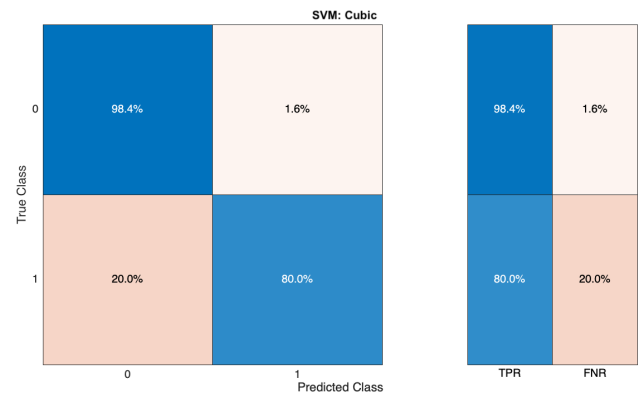


Figure 3.18: Cubic SVM: Confusion Matrix

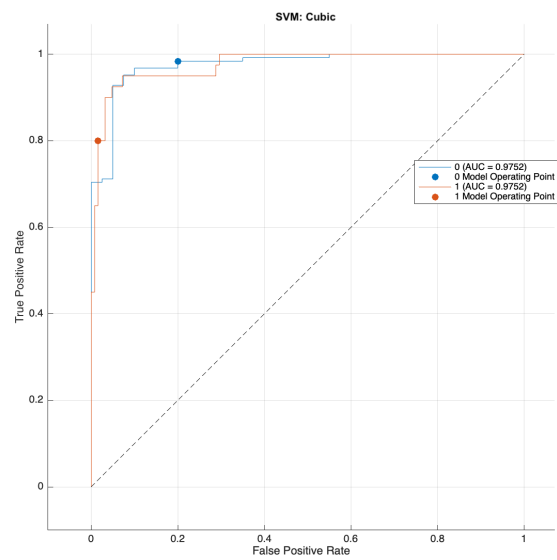


Figure 3.19: Cubic SVM: ROC Curve

3.5 Analyser: Movement identification

As with stress identification, movement identification has a significant impact on behavioural analyses. It holds information on a patient's activity providing data to the SLP on whether the stress identification can be trusted. Moreover, it indicates regions of high activity providing in itself data to the SLP to assess anxiety or short attention spans.

The movement identification system is the second and final submodule of the behavioural analysis system. Its purpose is to identify movement and calculate person-specific movement measurements. To do this, it uses acceleration data captured from a Polar H10 device, which samples at a rate of 50 Hz.

3.5.1 Pre-processing

The pre-processing stage entails reducing noise and formatting raw accelerometer data such that it can be used later for feature extraction and movement identification. It entails data resampling, outlier detection, and various filtering stages i.e. median filter, lowpass filter, and a moving average filter. Each of these processes is described below.

The **resampling** process entails down-sampling the raw accelerometer data to a sampling rate of $F_s = 50$ Hz. This is accomplished through the use of a cubic spline data interpolation function in MATLAB, namely `spline()`. This allows workable variable sizes while maintaining enough information for accurate movement identification (efficient memory usage). The problem with this approach is that it does not compensate for sampling rates lower than 50 Hz, meaning it does not allow upsampling as accelerometer data does not follow a specific trend as ECG data does. Upsampling the accelerometer data would cause inconsistent interpolations thereby causing data fabrications and possible misclassification as a result.

The **outlier detection** algorithm follows the same algorithm described in Section 3.4.1 for ECG data. The only difference is the correction value, if an outlier is detected then the value is replaced by:

$$\begin{aligned} \text{acc}[n] = & \text{mean}(\text{acc}[\text{wind}_{\text{begin}} : n - 1]) \\ & + (-1)^n (\text{std}(\text{acc}[\text{wind}_{\text{begin}} : n - 1])) \end{aligned} \quad (3.8)$$

This allows for a reasonably random correction value based on previous 'corrected' samples. The importance of this algorithm stems from the unpredictability of the data. An example of an artefact causing multiple consecutive outlier samples is illustrated in Figure 3.20. The signal in blue illustrates the original signal containing the artefact, and the signal in orange is the corrected signal.

The **filtering** process entails removing baseline wandering and other noise sources as

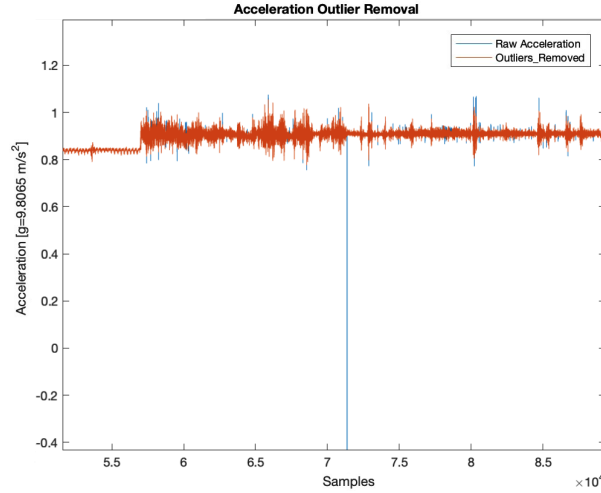


Figure 3.20: Accelerometer artefact corrected with Equation 3.8

well as isolating body movement and gravitational force from the total accelerometer signal. The first filter to be applied is a 3rd-order median filter which suppresses noise (smoothing technique). Thereafter, a 3rd-order lowpass Butterworth filter is applied. The filter possesses a 20 Hz cut-off frequency as most body movements occur below 15 Hz [150]. Thereafter, a moving average filter (MAF) with a 0.34 Hz cut-off frequency is applied to obtain both gravitational and body movement signals. Where the body movement signal is obtained by subtracting the gravitational signal (≤ 0.34 Hz) as illustrated in Equation 3.9.

$$\begin{aligned} \text{ACC_Gravity} &= \text{MAF}(\text{ACC}), \\ \text{ACC_Body} &= \text{ACC} - \text{ACC_Gravity} \end{aligned} \tag{3.9}$$

The effects of these filters are illustrated by Figure 3.21. It indicates the effect gravity has on accelerometer data and depicts body movement when it is extracted. The importance of the gravity signal can also be seen in its transitions. The step-like pattern thereof (see Figure 3.21) represents different gravitational effects on the x-axis of the triaxial accelerometer sensor, indicating postural changes in this direction.

3.5.2 Feature Extraction

The feature extraction process entails two basic steps, the segmentation and feature extraction of the body movement signal. The signal is segmented using a second-second, 50% overlap Hamming window.

From these segments, three basic statistical features are extracted, namely mean energy, absolute mean, and standard deviation. The energy of a segment is obtained by calculating its Euclidean distance (L2 norm):

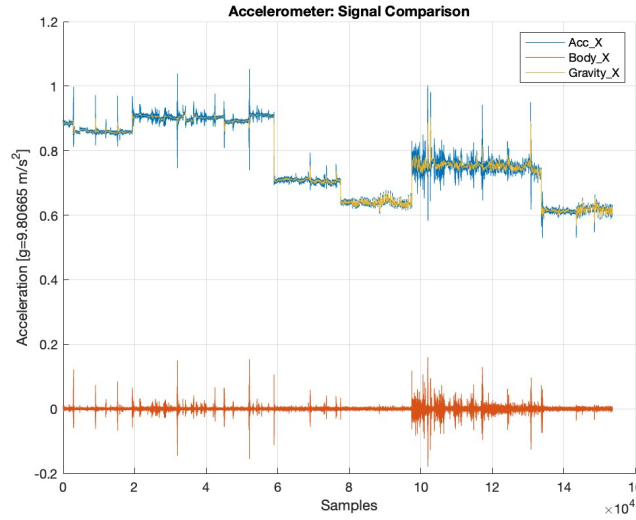


Figure 3.21: Illustration of the ‘Filtered Accelerometer Signal’, ‘Body Movement Signal’, and ‘Gravity Signal’ all of the x-axis of the triaxial accelerometer sensor.

$$E = ||x||_2 = \sqrt{\sum_{i=1}^n |x_i|^2}. \quad (3.10)$$

The energy of a segment is then averaged. Since a triaxial sensor is used the individual axes’ mean energy is then summed to obtain a single feature sample.

For the absolute mean feature, the values of a segment are made absolute and then summed. For the standard deviation feature, the standard deviation of a segment’s values is calculated. Once again the three axes’ values were summed together and used as a single feature sample.

The three basic features described above are simplistic ways to identify movement. By analysing variations within the signal, it is possible to determine whether movement has occurred or not. For example, if we examine Figure 3.21 and compare the samples around 8×10^4 to those around 10×10^4 , we can see that the signal activity (which indicates movement) at the later sample is much larger. This suggests that there was an increase in acceleration along the x-axis, indicating movement.

3.5.3 Classification

The classification section consists of comparing a simple threshold classifier to a machine-learning model (well-suited for binary classification and small datasets, and is often used when classifying acceleration data [147, 158]). For this purpose, the dataset, described later in Section 3.5.4, is divided into a training and testing set. The testing set is a randomly chosen partition of the dataset containing 20% of the data. The remaining 80% of the data was used to train a linear SVM model for the comparison. The parameters for the threshold method, on the other hand, were empirically obtained resulting in a

minimum threshold of 0.008, 0.100, and 0.120 for the mean energy, absolute mean, and standard deviation features, respectively. These parameters were selected to maximise the true positive results while minimising the false positive values without significantly compromising the false negative values. This was done to prevent an overload of false positives that might indicate a lack of attention or unsteady movements, resulting in incorrect information being provided to the SLP.

The results of the classifiers are indicated in Table 3.5.

Table 3.5: Movement identification: Classifier Comparison.

TP = True Positive. TN = True Negative. FP = False Positive. FN = False Negative [measured in Samples]

Classifier	Test Size	TP	TN	FP	FN	Correctly Identified	Execution Time [s]
Linear SVM	3143	1592	1520	13	18	3112	0.252
Threshold	3143	1529	1531	9	74	3060	0.579

Although the threshold is simplistic it is outperformed by the linear SVM classifier in both execution time and accuracy. An illustration of the SVM-movement identification can be seen in Figure 3.22.

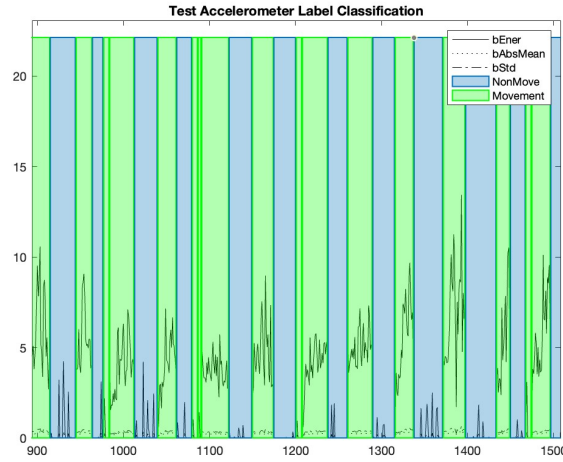


Figure 3.22: Movement identification - Feature illustration

3.5.4 Dataset Conversions

For the creation and evaluation of the movement identification classifier, a movement dataset containing accelerometer data is required. The HAPT dataset [5] is a common dataset used for this purpose. The acceleration signals from the dataset are extracted and a feature set is created as described in Section 3.5.2. The dataset contains various activities and labels such as walking, walking upstairs, walking downstairs, sitting, standing, laying, laying to seated, standing to laying and laying to standing. For this study, these activities were grouped into two categories, movement and non-movement. The features from all the experiments along with their corresponding labels were combined into a single MAT

file. The MATLAB script used to perform this task is included in Appendix F. In total, a 15718×4 feature set was created.

3.6 User Interface

User interfaces were created to aid in the usability of this study for SLPs. These interfaces provide an easy means of collecting and analysing speech interactions conducted during speech-language sessions.

3.6.1 Recorder

The recording of the SLP's contact sessions is entirely done through the use of the recorder application. It consists of 1) an audio recording protocol and 2) a Bluetooth comms that obtains raw ECG and accelerometer data from a Polar H10 device. An illustration of the recorder application is depicted in Figure 3.23 below.

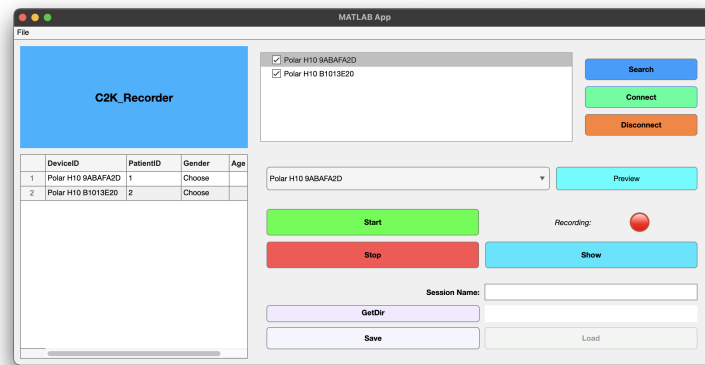


Figure 3.23: Recorder application with two connected Polar H10 devices.

The application allows for the discovery (search function) of Polar H10 devices. These devices only become active when attached to an individual, making it easy to discover only the devices that are being used - the application searches for these devices for 8 seconds and then times out. The user is then allowed to select which of the devices to connect to that are shown in the discovery list. Based on the connected devices, subject information can be entered in the patient table shown at the bottom left corner. Here the SLP can alter the *PatientID*, *Gender*, and *Age* of the patient that correlates to the *DeviceID*. To ensure the devices are attached properly to the patients and are transmitting ECG and accelerometer data, a preview button and device selector are provided. After the reliability has been checked the SLP can commence the session and start recording. The start button then starts recording audio and behavioural data (ECG and accelerometer). The show button allows for the visualisation of all the captured data. An example thereof is depicted by Figure 3.24.

It furthermore provides an option to pause and resume the recording session. The stop button stops the recording and allows the user to either save the recorded data or start

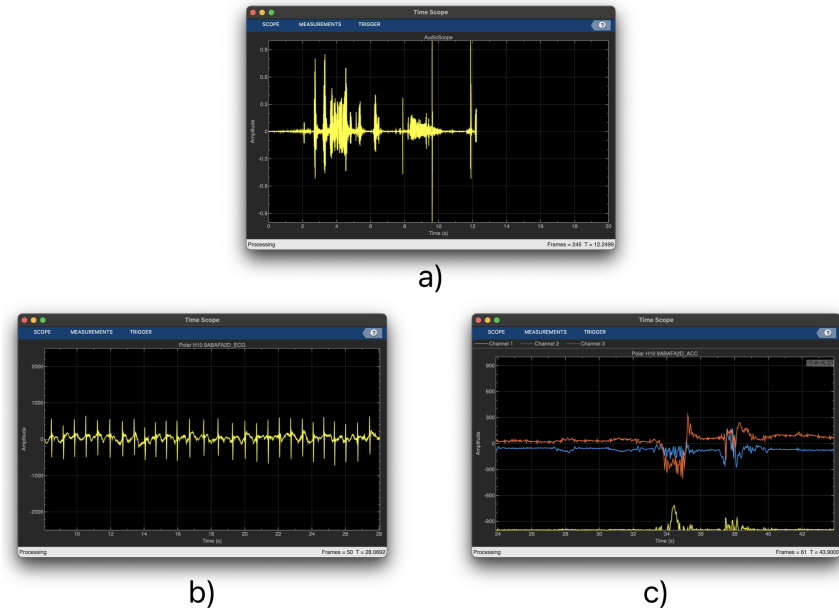


Figure 3.24: Recorder preview - a) audio signal, b) ECG signal, c) accelerometer signal

anew. Lastly, the application allows the user to select a desired file directory and provide the session with a name. The recorded data is then saved within a folder with the provided name in the form of a single WAV file for the audio data, two MAT files one for holding the recorded behavioural data with its timestamps and the other for the starting time of the recording, and lastly, a CSV file which holds the patient information entered into the patient table. An illustration of the saved directory is given in Figure 3.25.

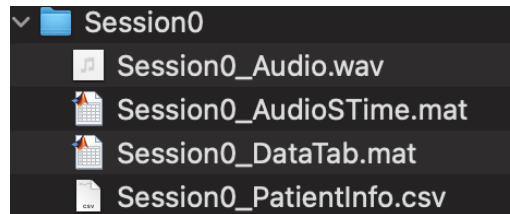


Figure 3.25: Example of a saved session.

3.6.2 Analyser

The analyser application's sole responsibility is to analyse and provide visual representations of the recording conducted by the recorder application. To accomplish this the application is provided a 'Load Folder' button that imports all files previously recorded during a contact session. This allows the SLP the ability to analyse any session at a later stage. An illustration of the analyser's main page is provided in Figure 3.26.

Once a session has been imported, the SLP has the following options: to add or remove new patients within the patient table (bottom-left corner), run both the behavioural and speaker diarisation analyses separately (purple and orange button, respectively), or to run them together (green 'analysis' button). The SLP can also listen to a 5-minute preview of the session, to ensure the correct session is being analysed.

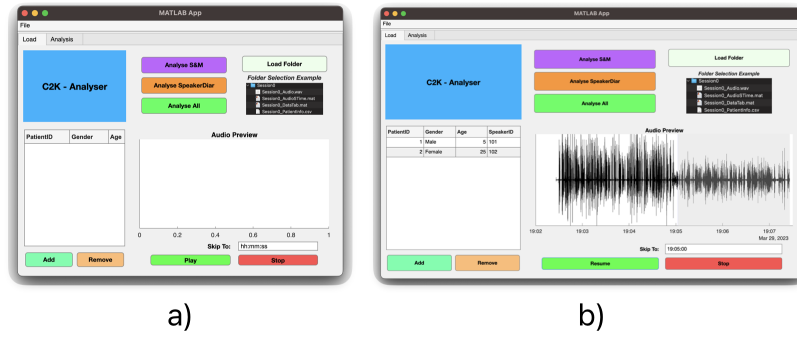


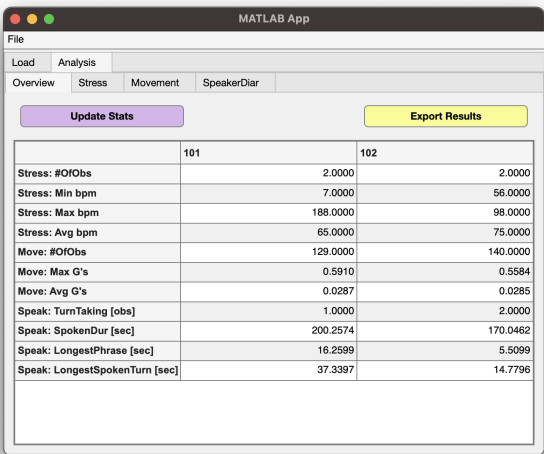
Figure 3.26: Analyser - Main page a) Unloaded, b) Loaded

The analyses occur automatically thanks to the setup of the speaker diarisation and behavioural (stress identification and movement identification) algorithms. Once these analyses have been conducted the SLP is provided with the following information about each patient: number of momentary stress observations, minimum heart rate, maximum heart rate, and average heart rate throughout the session, number of movement observations, maximum and average body movement acceleration throughout the session (measured in G 's = 9.807 m/s^2), the number of spoken turns (turn-taking), the total duration of speech, the longest phrase, and the longest speaking turn (without interruptions or long pauses). An illustration of these metrics is provided in Figure 3.27.

Within the 'Overview' tab where the results table is provided there also exists an 'Update Stats' button and an 'Export Results' button. The 'Update Stats' button updates the results table for example if the SLP decides to change the patient's identifier in the patient table. The 'Export Results' button, on the other hand, exports a PDF file containing the results table as well as all other graphical representations of the analysis results.

The analyser application provides graphical representations of speaker diarisation, stress, and movement results, see Figure 3.28 to Figure 3.30. The 'SpeakerDiar' tab illustrates the speaker diarisation results, as well as provides the ability to isolate speaker-specific speech both visually and audibly. The 'Stress' tab illustrates all individuals' raw ECG signals and HRV measures who were wearing a Polar H10 device during the recording of the contact session. Similarly, the 'Movement' tab illustrates each individual's raw acceleration signals and total movement measures.

Note that the measurements indicated in Figure 3.27 to Figure 3.30 were obtained within a single recording session from a single individual (the researcher). A clear limitation can be seen when comparing the "Stress: Min bpm" and "Stress: Max bpm" of patient 101 to those of patient 102. The results should have been nearly identical; however, due to a loose connection of the Polar H10 device (representing patient 101), the device failed to provide accurate ECG data, resulting in inaccurate HRV data and stress identifications.



The MATLAB App window displays a table of patient results. The table has columns for patient ID (101, 102) and various metrics. The metrics include Stress: #OfObs, Stress: Min bpm, Stress: Max bpm, Stress: Avg bpm, Move: #OfObs, Move: Max G's, Move: Avg G's, Speak: TurnTaking [obs], Speak: SpokenDur [sec], Speak: LongestPhrase [sec], and Speak: LongestSpokenTurn [sec].

	101	102
Stress: #OfObs	2.0000	2.0000
Stress: Min bpm	7.0000	56.0000
Stress: Max bpm	188.0000	98.0000
Stress: Avg bpm	65.0000	75.0000
Move: #OfObs	129.0000	140.0000
Move: Max G's	0.5910	0.5584
Move: Avg G's	0.0287	0.0285
Speak: TurnTaking [obs]	1.0000	2.0000
Speak: SpokenDur [sec]	200.2574	170.0462
Speak: LongestPhrase [sec]	16.2599	5.5099
Speak: LongestSpokenTurn [sec]	37.3397	14.7796

Figure 3.27: Analyser - Patient Results

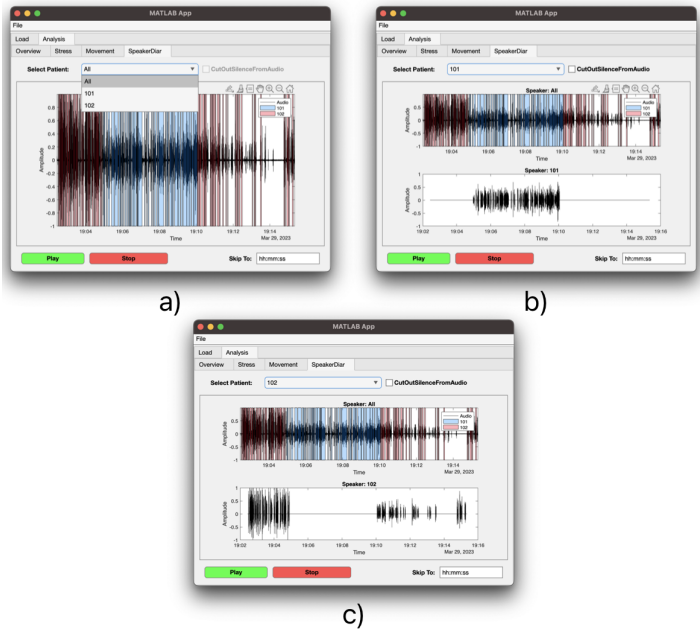


Figure 3.28: Analyser - Speaker diarisation results. a) Entire audio recording, b) Speaker: 101 extracted, c) Speaker: 102 extracted.

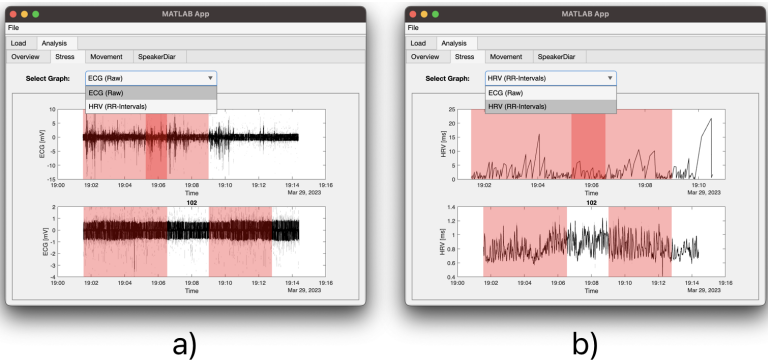


Figure 3.29: Analyser - Stress identification results a) Raw ECG representation, b) HRV representation.

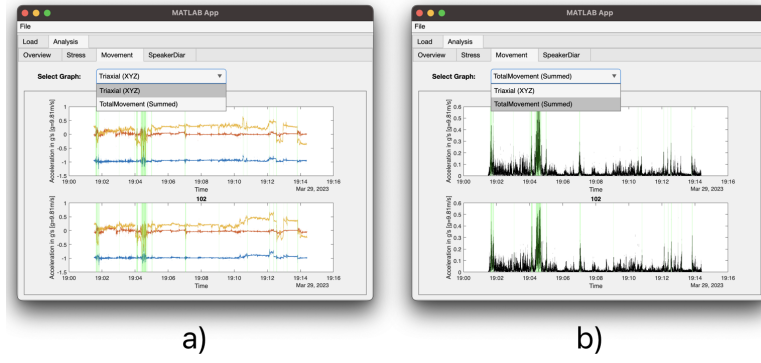


Figure 3.30: Analyser - Movement identification results. a) Raw triaxial acceleration representation, b) body movement acceleration representation.

3.7 Software and Tools

To have made this study possible software packages and measurement tools were required. The programming languages, libraries, and tools used for system development are described in this section.

3.7.1 Software

This study was built and tested on MATLAB[®] R2022b Update 5 (9.13.0.2193358). A list of MATLAB packages used for this study and their versions is provided in Table 3.6 below.

Table 3.6: MATLAB packages utilised.

Package	Version
Audio Toolbox	v3.3
Bluetooth Toolbox	v1.1
Communications Toolbox	v7.8
Deep Learning Toolbox	v14.5
DSP System Toolbox	v9.15
MATLAB Compiler	v8.5
Parallel Computing Toolbox	v7.7
Signal Processing Toolbox	v9.1
Statistics and Machine Learning Toolbox	v12.4
Wavelet Toolbox	v6.2

The App Designer application in MATLAB was utilised for the creation of the analyser and recorder applications which encapsulate all the algorithms used in this study. Moreover, the ‘Classification Learner’ application in MATLAB was utilised for the creation of some of the machine learning models and the evaluation thereof, together with the evaluation of the features utilised in each model.

Moreover, Python was used to extract some of the datasets used within this study. The script developed for this purpose is included in Appendix D.

Lastly, KubiosHRVStandard (version 3.5.0) was used as a reference tool for verifying HRV

features and stress identification (see Figure 2.12 in Section 2.5.6).

3.7.2 Hardware

The hardware utilised for this study includes Polar H10 devices and an Apple MacBook Pro 16-inch computer. The Polar H10 is considered the gold standard for measuring heart rate during physical activities and has even been found to outperform a Holter monitor [114]. The device is attached to the chest with the Polar Chest Strap Pro, and its durability and comfort make it ideal for nonrestrictive SLP speech interaction settings. Additionally, the device is equipped with a BLE module that allows communication between a host and the device, allowing the capabilities to capture raw ECG and acceleration data. The Polar H10 has been used in several studies within the same medical domain, both for monitoring heart rate during physical activities and working with children [112, 114, 115]. Speer (2020) [115] stated that the Polar H10 is recommended for HRV measurement of young children in an educational real-world setting, and it provides reliable HRV parameters that correlate well with parasympathetic activities.

The hardware architecture of the Apple computer is summarised in Table 3.7. The entirety of the study was executed on the CPU of the device.

Table 3.7: Hardware architecture of the host computer utilised for evaluation.

Hardware	Description
System on Chip (SoC)	Apple M2 Pro
CPU	8 Performance cores (up to 3.7 GHz) 4 Efficiency cores (up to 3.4 GHz)
GPU	19-cores (0.4 - 1.4 GHz)
Memory	16 GB

3.8 Integration

With the use of two separate systems that influence one another's operations, there is likely to be some set of protocols. A list of rules and standards to ensure seamless integration. A set of protocols for the integration of the analyser and recorder applications is provided in this section.

3.8.1 UI Integration

A governing protocol is created for the structure of saving data within the recorder application. This governing protocol consists of saving all the recorded data under a single folder with the inclusion of a single audio file, two MAT files, and a CSV file. This in turn allows the analyser application to import the folder with all the required data for its analyses. An illustration of this folder is provided in Figure 3.25. By establishing this protocol, and the subprotocols described below, there exists a seamless integration between the recorder and analyser application. The recorder application solely has to focus on capturing raw input data and save it for the analyser application to use later on. The

analysers application then imports a single folder to conduct its analyses and provide the SLP with not only statistical measurements but also visual and audio feedback.

3.8.2 Audio

The audio data plays a vital role in the diarisation of the SLP's sessions. It allows for the execution of the speaker diarisation algorithm that divides a session into 'person-specific' speech segments. It is therefore crucial for the recorder application and analyser to be on par. A summary of the protocols used for audio is given in Table 3.8 below.

Table 3.8: Audio protocol

Protocol	Description
Format	.wav, .flac
Sampling rate	16 kHz
Resolution	16-bit
Channel	Mono

The recorder application is set to record audio in a '.wav' format with a sampling rate of 16 kHz and a resolution of 16-bit. The analyser application can accept audio files in different formats, including WAVE (.wav), and FLAC (.flac). The analyser currently uses only a single audio channel for speaker diarisation. However, advanced systems like the ICSI RT-09 speaker diarisation system [36] can use multiple distant microphones (multiple channels) to improve diarisation accuracy through a beamforming submodule. This is beyond the scope of this study. There is no limit on the recording length and merely depends on the storage and memory available on the host device. Lastly, the recorder application is required to store the starting time of the audio recording as a MAT file containing a single `datetime` variable in the format of `yyyy.MM.dd HH:mm:ss.SSSSSSSSS` with an epoch of 2000-01-01 at nanosecond precision (1×10^9 ticks per second).

3.8.3 Behavioural

The behavioural integration protocols are designed to capture and utilise ECG and triaxial accelerometer data. These protocols are crucial as they help to identify stress through ECG data and movements through accelerometer data.

The recording protocols are limited to the capabilities of the Polar H10's ECG and accelerometer sensors, as stated in Section 3.2.2. The specific protocols for collecting behavioural data are summarised in Table 3.9 for easy reference.

Table 3.9: Behavioural protocol

Sensor	Protocol	Description
ECG	Sampling rate	130 Hz
	Resolution	14-bit
Accelerometer	Sampling rate	50 Hz
	Resolution	16-bit
	Range	$\pm 2G$

The recorder application and the Polar H10 establish a Bluetooth communication protocol, which enables the transmission of ECG and accelerometer data to the recorder application. Section 3.2.2 provides more details on this. The raw ECG and accelerometer data are stored in a single MAT file, along with their corresponding timestamps. The file contains a single n – by – 5 table, where n is the number of patients equipped with a Polar H10. The columns include ‘*pID*’, ‘*ECG*’, ‘*ecgTimestamps*’, ‘*ACC*’, and ‘*accTimestamps*’. That is the patient identification as given by the SLP in the recorder application, the raw ECG data and its corresponding timestamps, and the raw accelerometer data and its corresponding timestamps, respectively. The ‘*pID*’ column possesses a *character* datatype (cast depending on characters entered), the ‘*ECG*’ column consists of a single-column *double* datatype array, and the ‘*ecgTimestamps*’ and ‘*accTimestamp*’ columns each consist of a single-column *datetime* datatype array (same as mentioned in the audio section above), and the ‘*ACC*’ column consists of a 3-column *double* datatype matrix. An example of this format is provided in Figure 3.31 below.

	1 pID	2 ECG	3 ecgTimestamps	4 ACC	5 accTimestamps
1	1	100655x...	100655x1 datetime	38660x3 ...	38660x1 datetime
2	2	100665x...	100665x1 datetime	38668x3 ...	38668x1 datetime

Figure 3.31: Behavioural data format.

The analyser application is made aware of this format and can extract the required data that corresponds to their respective algorithm (e.g., ‘*ECG*’ and ‘*ecgTimestamps*’ are extracted for the stress identification algorithm). By utilising a fixed set of protocols the recorder application and analyser application can work seamlessly together without having to be executed consecutively.

3.9 Scalability and Efficiency

Most of the focus in terms of scalability and efficiency is placed on the speaker diarisation system, given its high demand in both memory and execution time. On the other hand, the stress and movement identification algorithms can analyse an hour’s worth of data without significant delay. The stress classifier can predict up to 3600 observations per second, while the movement classifier can predict up to 12472 observations per second, based on the current feature sets. For an hourly session, the stress identification and movement identification have execution times of 12.5 ms and 1.11 s, respectively, based on their respective feature segmentation parameters. However, it’s worth noting that the behavioural classifiers cannot execute an infinitely large recording session since the algorithms are executed on the entire session and are not compartmentalised into smaller segments, as done within the speaker diarisation algorithm.

For the speaker diarisation algorithm, the entire audio recording is kept in storage and only the segments that are required for analysis are extracted and used. With this said, the

audio data is extracted in 5-minute segments, analysed and replaced by the next 5-minute segment. This ensures that the algorithm does not exceed the memory capabilities of the host device (and of MATLAB's internal memory restrictions) while being executed. This approach not only ensures memory efficiency but also the use of parallel computation. The efficiency of the parallel computation is based on the host device's central processing unit, the number of cores it possesses and the amount of memory it can allocate to each core. Lastly, the structure of the code used to execute the speaker diarisation algorithm not only allows parallel computing but also cloud-based computing.

3.10 Challenges Faced

Two of the challenges faced within this study were the determination of the forgetting factors and constants used within the R-peak detection algorithm needed for HRV creation (see Section 3.4.3), and secondly, the creation of the GMMs needed for the speaker diarisation system (see Section 3.3.3).

The speaker diarisation system's results are reliant on the conversion of the GMMs. The error that occurred is that an '*ill-conditioned covariance*' matrix was created stopping the conversion process of the `fitgmdist()` function (see Section 3.3.3). The error may have occurred due to:

1. The feature dimensionality is relatively high and there are not enough observations.
2. Some of the features are highly correlated.
3. Some or all the features are discrete.
4. Too many Gaussian components were used to fit the data.

To resolve the issue, the 'CovType' was set to 'diagonal', and the 'Replicate' option was utilised to repeat the EM algorithm used to develop the GMMs with a new set of initial values. The algorithm was iterated three times with the 'Replicate' option set to 3 iterations before determining a model based on the highest log-likelihood.

Chapter 4

Evaluation

This study aims to automate the analysis process of a speech-language pathologist (SLP). The focus is placed on providing an automated means of identifying speaker-specific speech regions as well as the behaviour of an individual throughout the observation period. These objectives led to the creation of a speaker diarisation system for detecting who spoke when and behavioural identifiers in the form of stress and movement identification algorithms. Moreover, this study introduces a data-capturing counterpart containing an audio recording module and a BLE system that obtains raw ECG and acceleration data from Polar H10 devices. The respective systems then use this data - audio, ECG, and acceleration - to accomplish the automation process.

Here, a description of the evaluation metrics, datasets, and experimental setup for each system is provided. Thereafter, the results and analyses are provided to verify each system, and lastly, conclusions are drawn from the findings to complete the validation process of this study.

4.1 Evaluation Metrics

There are two sets of evaluation metrics to be utilised, one for the speaker diarisation system and one for the behavioural system. All the metrics, nonetheless, are created from the following base criteria:

- True Positive (TP) - Number of correctly identified true observations.
- False Positive (FP) - Number of incorrectly identified true observations.
- True Negative (TN) - Number of correctly identified negative observations.
- False Negative (FN) - Number of incorrectly identified negative observations.

4.1.1 Speaker Diarisation

For the evaluation of the speaker diarisation system, the diarisation error rate (DER) metric is employed, indicating the percentage of time incorrectly attributed to a speaker or non-speech. The DER metric is a summation of the three error types that can fall under a diarisation system: false alarm (FA), missed detection, and confusion between speaker labels, divided by the total speech duration present. The DER equation is illustrated by Equation 2.1 in Section 2.3.3 and is re-iterated below.

$$\text{DER} = \frac{FA + MISSED + SpeakerConfusion}{Total\ Duration\ of\ Speech}$$

The errors in the speaker diarisation system are either a result of the VAD module or the segmentation and clustering module. The errors resulting from the VAD module include FA and missed speech. Whereas, errors resulting from the segmentation and clustering module include missed speech due to the quantisation error (identified by VAD but excluded in segmentation and clustering due to the minimum duration requirement) and incorrect speaker identification. The ‘MISSED’ metric is, therefore, divided into two parameters: missed speech caused by the VAD module and missed speech caused by the segmentation and clustering module.

For the errors resulting from the VAD module, a 250 ms tolerance window is incorporated on both sides of a hypothesised speech segment. This means only speech exceeding 250 ms on either side of a hypothesised speech segment will be considered either as FA or missed detected speech.

A summary of the evaluation metrics used for the speaker diarisation system is illustrated in Table 4.1.

Table 4.1: Speaker diarisation evaluation metrics

Error Type	Description [Unit]
FA	False Alarm (False Positive). [s]
MISS_VAD	Missed speech caused by VAD module (False Negative). [s]
SE_MISS	Speaker Error - Missed speech caused by quantisation error resulting from segmentation and clustering module. [s]
SE_IncCLUST	Speaker Error - Incorrectly identified speaker. [s]
DER	Diarisation Error Rate - Summation of the above errors divided by the total speech duration. [%]

4.1.2 Behavioural

The behavioural system holds three key algorithms: a beat detection algorithm, a stress identification algorithm, and a movement identification algorithm. The three algorithms will be assessed using some of the following evaluation metrics: sensitivity, specificity,

PPV, and accuracy. The evaluation set’s TP, FP, FN, and total size will also be provided, as other metrics can be derived from them. The correct peaks (TP), incorrect peaks (FP), and missed peaks (FN) for the beat detection algorithm are calculated within a 30 ms tolerance window, allowing a comparison to be drawn with the algorithm created by [128]. The evaluation of the stress and movement identification algorithms will not incorporate a tolerance window.

Sensitivity indicates the percentage of correctly identified true positives, specificity is the percentage of correctly identified true negatives, PPV, also known as precision, is the percentage of correctly identified positive values, and accuracy is the percentage of the total correctly identified observations. Sensitivity is illustrated in Equation 2.3 in Section 2.5.5 alongside specificity, PPV, and accuracy in Equations 2.4, 2.5, and 2.7, respectively.

Among these metrics is the detection error rate (DER_{Beat}) metric commonly utilised for beat detection algorithms. Its mathematical expression is given by Equation 4.1.

$$\text{DER}_{\text{Beat}} = \frac{FP + FN}{TP} \quad (4.1)$$

4.2 Dataset Description

When assessing any classification or identification algorithm, it is common to compare the number of correctly and incorrectly identified observations with a reference set (ground truth). This section will evaluate all the algorithms, each with its own dataset.

The datasets used, and the processing thereof to enable the evaluation of the algorithms are described next.

4.2.1 Speaker Diarisation

For the evaluation of the speaker diarisation system, the ‘LibriSpeech ASR corpus’, [2], was utilised. It is a corpus of English-read speech extracted from LibriVox’s public-domain audiobooks, sampled at 16 kHz. The corpus is divided into various subsets, of which the ‘development-clean’ and ‘development-other’ subsets were utilised for this study. The ‘development-clean’ subset contains cleaner (on average) audio files of various US-accented English speakers. On the other hand, the ‘development-other’ subset consists of harder-to-recognise speech resulting from additional noise. These subsets consist of multiple short clips of single-read passages, sometimes as short as 3 seconds, making it ill-suited to evaluate a speaker diarisation system in its current format.

A script was created to extract various clips from 5 randomly chosen speakers: ‘84’, ‘174’, ‘251’, ‘652’, and ‘700’, to be merged into a single 9-minute audio file. Speaker ‘700’ is the only speaker extracted from the ‘development-other’ subset. The clips from the various speakers are randomly chosen and merged, providing a similarity to a real-world conversation where any individual, including the current speaker, can continue the conversation

or start anew. This approach, however, does exclude the simulation of overlapping speech. The exclusion thereof is standard when evaluating the DER of a speaker diarisation system. An annotation file was also created alongside the audio file to provide the ground truth.

The annotation of the ground truth signal was initially done by hand, however, this proved quite time-consuming. Instead, the annotation was done by automatically detecting speech regions using MATLAB's built-in `detectSpeech()` function. Each speaker's audio segment was overlapped with the speech regions obtained to indicate who spoke when. However, the function struggled to detect some of the more softly spoken short uttered phrases in the audio file (see Figure 4.1). These utterances accounted for a small portion of the audio file and were left unaltered for the evaluation of the speaker diarisation system.

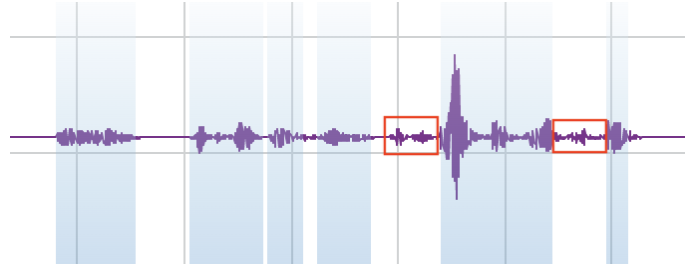


Figure 4.1: Missed speech annotations during the creation of the ground truth signal for speaker diarisation.

Note that the dataset used in creating the speaker diarisation system, namely the CommonVoice training set, [1], differs from the dataset used for evaluation. The evaluation set is chosen to provide clean audible speech for testing while simultaneously providing audio data that mimics a task often employed by an SLP during a parent-child contact session, namely reading out loud to the child. With the audio clips being equally intelligible, the speaker diarisation system can be evaluated on the sole premise of the difference arising from inter-speaker vocal characteristics.

4.2.2 Beat Detection

The evaluation of the beat detection algorithm utilises the MIT-BIH Arrhythmia dataset, [3]. It contains over 48 half-hour segments of two-channel ambulatory ECG (Holter monitor) recordings, which include over 110'000 annotations of various beat and non-beat orientations. Of the 48 recordings, 13 are of healthy persons, and 35 are of persons with various cardiac diseases. The dataset has been standardised to evaluate beat detection algorithms, allowing for comparability with other studies. It also presents various difficulties in beat detection, as some ECG signals consist of various arrhythmias, making it increasingly challenging to detect a heartbeat accurately. The algorithm is evaluated on the following recordings illustrated in Table 4.2.

Table 4.2: Beat detection - Evaluated recordings

Type of beats in signal	Signal
Normal	101m
	103m
	105m
	113m
	123m
Diseased	102m
	109m
	116m
	200m
	210m

4.2.3 Stress Identification

The stress identification algorithm is evaluated using the WESAD dataset, [4], which contains stress-annotated ECG signals from 15 subjects with a mean age of 27.5 ± 2.4 years. From feature extraction (see Section 3.4.2), 184 observations (45 stress) were randomly divided into a 90 – 10 training and testing set, respectively. The training data consists of a 165×6 feature set holding 40 stress observations. The testing data consists of an 18×6 feature set with 5 stress observations (a single observation was excluded due to the training-testing set division). The features include *mean RR*, *skewness RR*, *pNN50*, *absolute power LF*, and *RMSSD*. The last column of the feature sets contains the classification labels.

4.2.4 Movement Identification

The movement identification algorithm was evaluated using the HAPT dataset, [5]. The dataset is composed of 30 subjects (61 experiments) wearing a waist-mounted smartphone that provides triaxial accelerometer data and annotations of various activities.

For this study, the annotation list was simplified to movement and non-movement. The conversion of this dataset is described in Section 3.5.4, and based on the feature extraction procedure described in Section 3.5.2, the dataset produces a 15718×4 feature set, which is used for both training and testing of the movement classifier. The dataset is divided into a random non-stratified partition with an 80 – 20 ratio for the training and testing sets. The training set has a 12575×4 feature set holding 6400 movement observations. The testing data comprises a 3143×4 feature set holding 1491 movement observations. The last column of the feature sets contains the classification labels.

4.3 Experimental Setup

Each system and algorithm is evaluated independently on the Apple MacBook Pro 16-inch computer (see Section 3.7) and executed on cleared memory and cache. The device is continuously kept on charge and maintained under cooled conditions to prevent any decrease in performance.

The Polar H10 **data capturing** is evaluated against that of CamNtech’s Actiheart 5. The Actiheart 5 is a single-lead ECG and activity recorder designed for research purposes. The device is used in conjunction with the Actiheart Software package (Version 5.1.24), allowing the visualisation and analysis of recorded data. It has been widely used by researchers studying heart health and activity levels, [159–161]. To evaluate the accuracy of the data, both devices are attached to an individual’s chest, as depicted in Figure 4.2, and simultaneously record ECG and acceleration data. The data is then visually inspected to determine the correlation between the recordings.

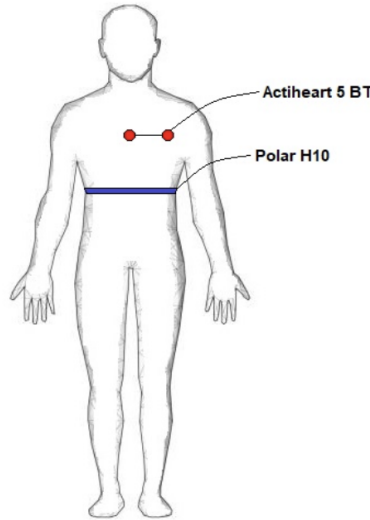


Figure 4.2: Sensor placement for data capturing evaluation.

The **speaker diarisation system** is evaluated five times on the 9-minute audio file described in Section 4.2.1 to obtain an averaged evaluation result. The VAD module is incorporated in the evaluation of the speaker diarisation system, meaning the audio file is not pre-processed into speech and non-speech segments.

The **beat detection algorithm** is evaluated on multiple ECG recordings from the MIT-BIH Arrhythmia dataset, [3], providing the opportunity to compare the algorithm to others within the same domain. The recordings used for evaluation are represented in Table 4.2. In addition to these recordings, a single ECG recording is also collected from the Polar H10 device. The heartbeats are manually annotated and compared to those identified by the beat detection algorithm. A conclusion will be drawn from evaluating the ‘normal’ recordings and the Polar H10 device’s recording.

The **stress identification algorithm** is evaluated on the test set created from the WE-

SAD dataset, [4]. The evaluation is executed using the ‘Classification Learner’ application in MATLAB. A 5-fold cross-validation is also implemented within the training set, providing additional validation of the model’s accuracy while also protecting the model against overfitting. The beat detection algorithm is incorporated in the evaluation of the stress identification algorithm as it was initially employed in creating the test set from raw ECG signals.

Lastly, the **movement identification algorithm** is evaluated on the test set created from the HAPT dataset. The evaluation of the algorithm is conducted in the ‘Classification Learner’ application in MATLAB. It also provides validation for the model through the use of a 5-fold cross-validation method.

4.4 Results and Analyses

The evaluation results of the data capturing system, speaker diarisation system, beat detection algorithm, stress identification algorithm, and movement identification algorithm are provided in this section. The results are provided in the form of tables and other graphical illustrations. Trends and challenges faced are also discussed.

4.4.1 Verification: Data Capturing

The following text describes the verification of the audio, ECG and acceleration data captured.

To verify the audio data, the record button on the main page of the recorder application was pressed, and the playback was listened to afterwards. This process is shown in Figures 3.26 and 3.28.

To evaluate the captured data, the raw ECG and accelerometer data from the Polar H10 device were visually assessed and compared to the corresponding data captured by the Actiheart 5 device. The Actiheart Software (Version 5.1.24) was used in processing the Actiheart 5’s data. In Figure 4.3, an event was noted in both the Actiheart 5 and Polar H10 data. Additionally, Figure 4.4 shows that the heart rate trends of both devices are similar, with an average heart rate of 83 bpm. Note that the heart rate data provided by the Actiheart Software is post-processed to remove outliers and mitigate sudden changes.

There is a time difference of approximately 3 seconds between the Actiheart 5’s data and that of the Polar H10, which may have been caused by their respective real-time clocks. The Actiheart Software is hosted on a Windows laptop, while the recorder application is hosted on the host computer.

This comparison between the Actiheart 5 data and the Polar H10 indicates that the BLE protocol was correctly created and provides real-time ECG and accelerometer data.

Lastly, simultaneous data capturing is verified. This study aimed to simultaneously cap-

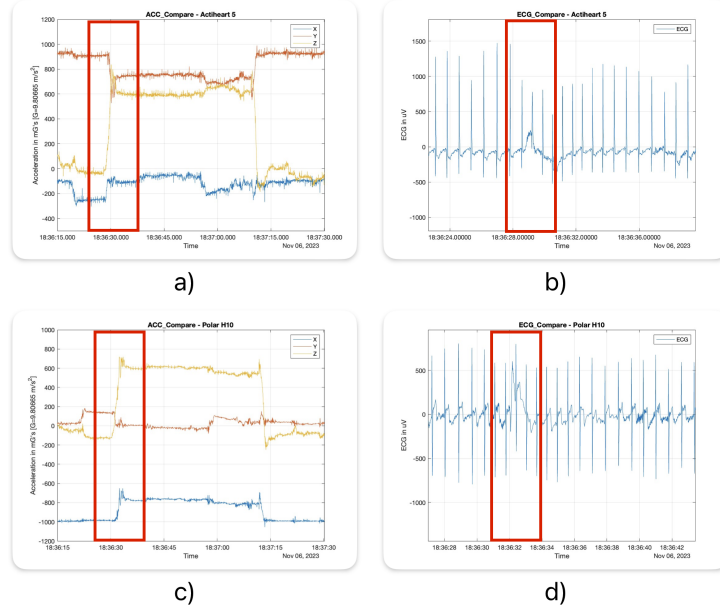


Figure 4.3: Data capturing evaluation - Polar H10 changes correlate to that of the Actiheart 5. a) Accelerometer data (Actiheart 5); b) ECG data (Actiheart 5); c) Accelerometer data (Polar H10); d) ECG data (Polar H10).

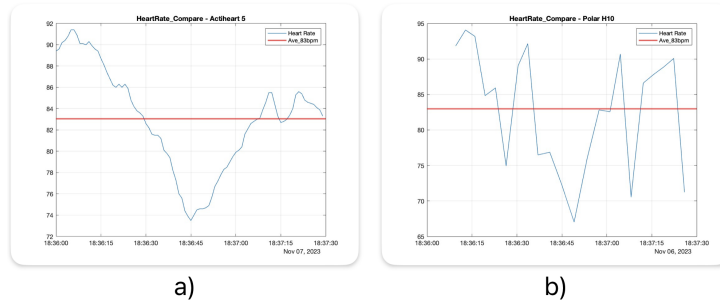


Figure 4.4: Data capturing evaluation - Polar H10 heart rate trend correlates to that of the Actiheart 5, with the same average heart rate of 83 bpm. a) Heart rate (Actiheart 5); b) Heart rate (Polar H10).

ture audio data and raw ECG and accelerometer data from multiple devices. ‘Simultaneously’ is loosely used as the system is merely required to provide visually comparable results among the time axis. Simultaneous data capturing of the behavioural data (between Polar H10 devices) can be seen by visually comparing the plots within Figure 3.30 in Section 3.6. Simultaneous data capturing between the audio and behavioural data can be seen by comparing the time axes end values of the plots within Figure 3.28 to that of Figure 3.30 in Section 3.6.

4.4.2 Verification: Speaker Diarisation

An illustration of the speaker diarisation output is provided in Figure 4.5. Here, it can be seen how the hypothesised output correlates with the ground truth. It must be said that since the speaker diarisation system randomly assigns a cluster name (e.g., ‘1’) to a speaker, the cluster names had to be rearranged to match that of the ground truths.

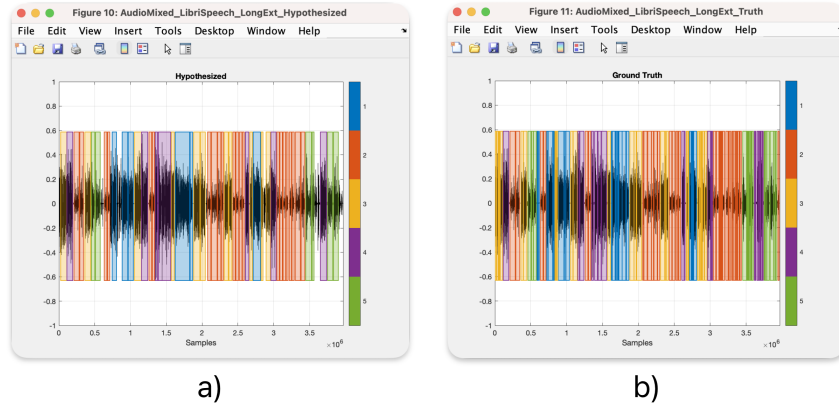


Figure 4.5: Speaker diarisation output verification. a) Hypothesised; b) Ground truth

The evaluation metrics' results for both a single-threaded and multi-threaded (3-cores) execution are provided in Table 4.3 and Table 4.4, respectively. The executions are based on a 9-minute and 29-second audio file created from the LibriSpeech dataset (see Section 4.2.1). Between entries 1 and 3 of the single-threaded execution, Table 4.3, the VAD parameters were adjusted to improve FA without affecting the MISS_VAD and SE_MISS metrics. The `detectSpeech()` function's *mergeDistance* parameter was changed from 0.3 to 0.2 s between entries 1 and 2, improving the FA but significantly affecting the SE_MISS metric. The parameter was then changed to 0.25 s from entry 3, partially increasing the FA error again but dramatically improving the SE_MISS error. This setup provided the best DER result and was used throughout the rest of the evaluations (including the parallel evaluations). The FA errors that are, however, still present can also be a result of miss misclassified speech regions in the annotated ground truth signal. This is evident in Figure 4.1.

From Table 4.3, it can be seen that when using the same system parameters the results stay the same. The only difference is the execution time. This is to be expected as the system makes use of the Expectation Maximisation (EM) algorithm within the `fitgmdist()` function to converge GMMs and thereby determine the number of speakers present. This in turn affects the HMM structure and therefore the Viterbi algorithm which is the system's largest time-consuming operation.

Table 4.3: Speaker diarisation evaluation results - Single-threaded execution

FA [%]	MISS_VAD [%]	SE_MISS [%]	SE_IncCLUST [%]	DER [%]	Time [s]
1.847	0.000	3.711	0.619	5.558	678.874
1.100	0.000	5.618	0.000	6.718	626.124
1.346	0.000	3.861	0.000	5.207	471.602
1.346	0.000	3.861	0.000	5.207	601.240
1.346	0.000	3.861	0.000	5.207	535.423

For the parallel evaluations, the *mergeDistance* parameter was set to 0.25 s. Three cores were assigned to execute the speaker diarisation algorithm, as most computers possess at least four cores. Given a 9-minute audio file, the speaker diarisation algorithm would

Table 4.4: Speaker diarisation evaluation results - Parallel (3-cores) execution

FA [%]	MISS_VAD [%]	SE_MISS [%]	SE_IncCLUST [%]	DER [%]	Time [s]
1.346	0.000	3.861	0.000	5.207	910.059
1.346	0.000	3.861	0.000	5.207	874.092
1.346	0.000	21.658	17.798	23.004	557.070
1.346	0.000	3.861	0.000	5.207	341.736
1.346	0.000	3.861	0.000	5.207	812.085

produce two audio segments - a 5-minute segment and an approximately 4-minute segment - for initial diarisation. After that, the system would merge the diarised segments and re-iterate to produce the final output. During the initial diarisation stage, the segments are independent of one another, providing the opportunity to execute them simultaneously. Therefore, two of the three cores are assigned to this operation for accelerated execution times.

The results of the parallel evaluations, however, differ from what was expected. For instance, entries 1, 2, and 5 provided slower execution times than the single-threaded executions, while entry 4 provided consistent results under accelerated times. This may be due to a structure error in the parallel setup, as the code executer waits for the parallel operation even after it has finished.

Apart from the execution times, there were instances where the system struggled while computing the GMMs, leading to incorrect speaker classifications and ultimately speaker merges. An example hereof is provided in Figure 4.6, representing the third entry in Table 4.4.

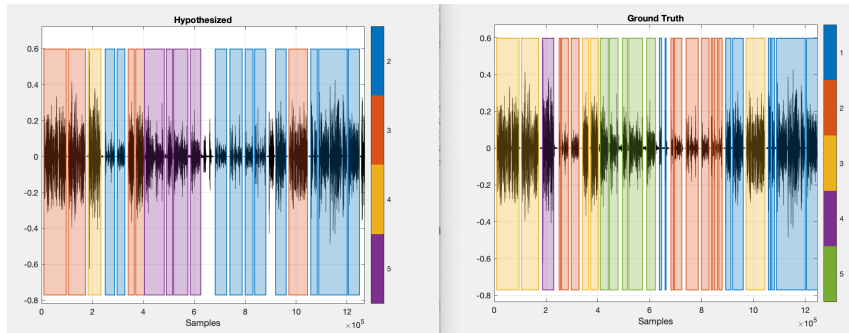


Figure 4.6: Incorrect speaker clustering result - Parallel execution entry 3

Figure 4.7 illustrates the time consumed by a single-threaded execution of the speaker diarisation system. The figure shows that most of the time is spent creating the GMMs and executing the Viterbi algorithm. The VAD subsystem has a very short execution time in this instance due to the use of the `detectSpeech()` function after confirming that the audio segment does not contain any non-speech audio in the form of non-speech sounds (noise). While this is expected under ideal conditions, as in which the test file was created, in practical situations, the VAD subsystem's execution time may increase. Of course, the system's time consumption also depends on the amount of speech present in the audio recording. This is because most of the calculations are applied to speech

segments extracted by the VAD subsystem. On average, the single-threaded execution takes around 9 minutes for an audio file containing approximately 8 minutes of speech.

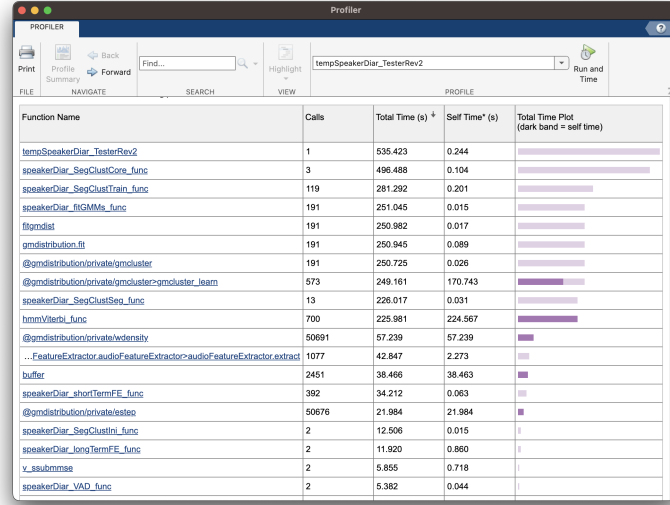


Figure 4.7: Time analysis for speaker diarisation execution (09:29 duration).

Besides diarising the audio data into person-specific regions, this study also provides commonly extracted **speech measurements** for speech interaction analyses. These measurements include turn-taking, total speaking duration, longest phrase, and longest spoken turn, and are reliant on the accuracy of the speaker diarisation system. For instance, even the smallest misclassification of a speaker segment between two consecutive segments can significantly alter the measurement of the longest spoken turn. In the test file, this was the case for 'Speaker 1' as shown in Table 4.5 and Table 4.6, which respectively provide the ground truth and hypothesised calculations of the speaker diarisation measurements for the given test file. The missed turn-takings noted in these tables stem from quantisation errors where the speaker diarisation system cannot assess speech segments less than 1.5 seconds.

Moreover, the inaccuracies of the spoken duration, longest phrase, and longest spoken turn are a result of either missed or falsely detected speech segments. Table 4.7 provides an overview of the average accuracy of each measurement.

Table 4.5: Speaker diarisation measurement results - Ground Truth

Speaker	Turn-taking [observations]	Spoken Duration [s]	Longest Phrase [s]	Longest Spoken Turn [s]
1	10	88.831	6.360	34.328
2	17	135.330	5.016	25.320
3	19	85.826	5.520	8.208
4	11	103.050	7.032	29.525
5	6	51.366	4.824	14.864

Lastly, a single-threaded execution of the speaker diarisation system consumed, at worst, approximately 3.37 GB of memory, with 2 GB pre-allocated to the launch of MATLAB.

Table 4.6: Speaker diarisation measurement results - Hypothesised

Speaker	Turn-taking [observations]	Spoken Duration [s]	Longest Phrase [s]	Longest Spoken Turn [s]
1	8	77.499	7.900	18.240
2	16	126.160	4.980	22.029
3	17	81.038	5.470	8.030
4	11	98.538	11.490	17.780
5	6	51.829	8.610	15.350

Table 4.7: Speaker diarisation measurement accuracies.

Measurement	Average Accuracy [%]
Turn-taking	92.718
Spoken Duration	93.922
Longest Phrase	66.457
Longest Spoken Turn	78.984

4.4.3 Verification: Beat Detection

The beat detection algorithm was evaluated on randomly selected ECG signals from the MIT-BIH Arrhythmia dataset (five normal and five diseased) [3]. The evaluation results are summarised in Table 4.8. Figure 4.8 indicates false and missed detected beats.

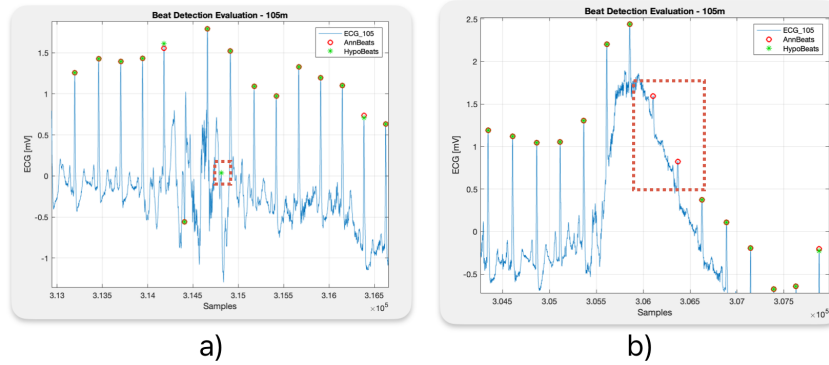


Figure 4.8: Beat detection errors. MIT-BIH - 105m a) False detected beat; b) Missed detected beats

The beat detection algorithm obtained an average DER of 0.910% for normal beat signals and 10.463% for diseased beat signals. Obtaining perfect and near-perfect results for 3 out of the 5 normal beat signals. These results, however, fail in comparison to state-of-the-art beat detection algorithms, such as that of [128], obtaining an average DER of 0.135% for the entire MIT-BIH Arrhythmia dataset [3]. As a result, the beat detection algorithm created in this study was also evaluated against an ECG signal obtained from a Polar H10 device. The individual was in a seated position for most of the recording, speaking from time to time, and stood up once. There was no difference between the manually annotated signal and the beat detection algorithm, obtaining zero misclassified beats (DER=0%). An illustration of the annotated Polar H10 ECG signal is presented in Figure 4.9, with the results indicated in Table 4.9.

Table 4.8: Beat detection evaluation results.

Type of beats in signal	Signal	TP	FP	FN	Total Size	Sensitivity [%]	PPV [%]	DER [%]	Time Consumption [s]
Normal	101m	1863	5	2	1865	99.893	99.732	0.376	0.960
	103m	2083	0	1	2084	99.952	100.000	0.048	0.789
	105m	2545	78	27	2572	98.950	97.026	4.126	0.721
	113m	1795	0	0	1795	100.000	100.000	0.000	0.817
	123m	1518	0	0	1518	100.000	100.000	0.000	0.716
Diseased	102m	1949	233	238	2187	89.118	89.322	24.116	0.706
	109m	2460	47	72	2532	97.156	98.125	4.837	0.704
	116m	2372	14	40	2412	98.342	99.413	2.277	0.723
	200m	2392	195	209	2601	91.965	92.462	16.890	0.720
	210m	2582	22	68	2650	97.434	99.155	4.197	0.737

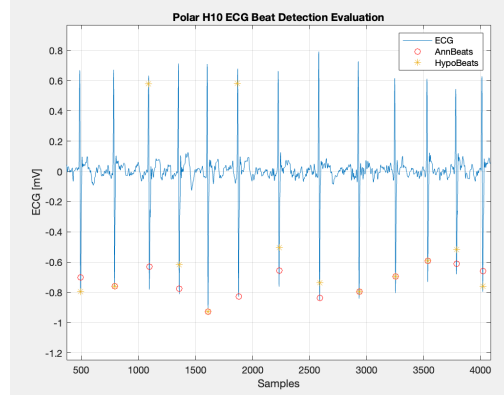


Figure 4.9: ECG beat detection evaluation - Polar H10

Table 4.9: Beat detection evaluation results - Polar H10 ECG measurement

TP	FP	FN	Total Size	Sensitivity [%]	PPV [%]	DER [%]
957	0	0	957	100.000	100.000	0.000

The following heart measurements, min, max, and average beats per minute, were evaluated based on the Polar H10 measurement described above, with the results summarised in Table 4.10.

Table 4.10: Beat detection measurement results.

Category	Min BPM	Max BPM	Average BPM
Hypothesised	56	98	75
Ground Truth	56	98	75
Accuracy	100.000	100.000	100.000

4.4.4 Verification: Stress Identification

The stress identification classifier is evaluated using MATLAB's 'Classification Learner' application. It provides validation for the classifier training via 5-fold cross-validation, testing accuracy, confusion matrices, and the ROC curve. The confusion matrix and ROC curve for validation are provided in Section 3.4.3, Figure 3.18 and 3.19, respectively. The confusion matrix and ROC curve for the testing set are provided in Figure 4.10 and Figure 4.11, respectively. The evaluation results are summarised in Table 4.11.

PPV	100.0%	100.0%
FDR		
	0	1
	Predicted Class	

Figure 4.10: Stress identification - Test confusion matrix

Next, the stress observation measurement was evaluated for the validation and testing sets. Table 4.12 presents the hypothesised and ground truth stress observations.

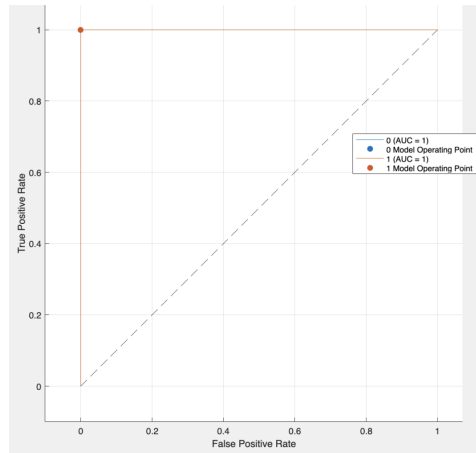


Figure 4.11: Stress identification - Test ROC

Table 4.11: Stress identification - Evaluation metrics results.

Category	TP	TN	FP	FN	Total Size	PPV [%]	Specificity [%]	Accuracy [%]
Validation	32	123	2	8	165	94.118	93.893	93.939
Testing	5	13	0	0	18	100.000	100.000	100.000

Table 4.12: Stress identification measurement results - Stress observations

Category	Measurement	Hypothesis	Ground Truth	Accuracy [%]
Validation	Stress Observations	32	40	80.000
Testing	Stress Observations	5	5	100.000

Lastly, a randomly chosen ECG recording from the WESAD dataset was chosen, namely S3, to compare the stress identification algorithm to that of the Kubios software. Kubios is a research-grade HRV analyser and is capable of not only identifying stress but also indicating PNS and SNS activities. HRV data extracted by this study's beat detection algorithm was provided to the software package for evaluation. The stress regions in the Kubios application are highlighted in blue as depicted by Figure 4.12. This stress region correlates to that obtained by the stress identification algorithm (see Figure 4.13).

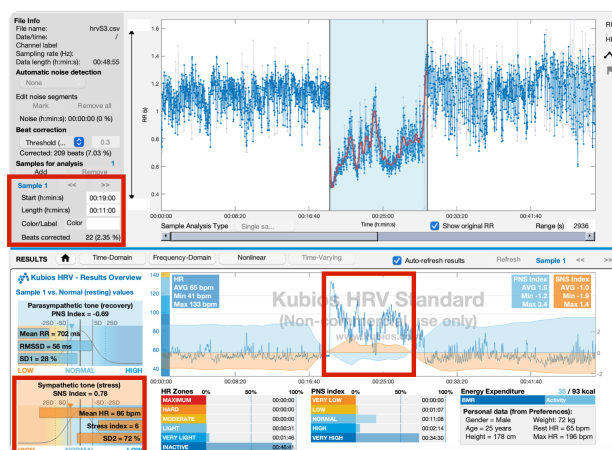


Figure 4.12: Stress identification Kubios - WESAD S3 Stress region between 19 minutes and 30 minutes. Stress index of 6. SNS is more active than PNS.

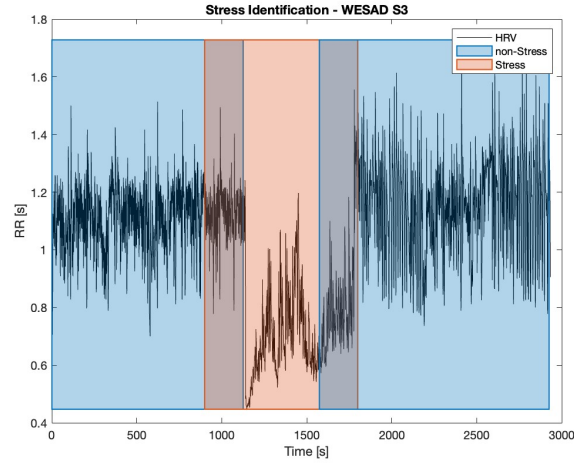


Figure 4.13: Stress identification algorithm - WESAD S3 Stress region identified between 900 seconds and 1800 seconds using 5-minute windows with 75% overlap.

4.4.5 Verification: Movement Identification

The evaluation of the movement identification classifier resembles that of the stress identification. Table 4.13 summarises the evaluation results for both the validation and testing sets. The confusion matrices and ROC curves for the validation and testing set are also illustrated in Figure 4.14 and Figure 4.15, respectively.

Table 4.13: Movement identification - Evaluation metrics results

Category	TP	TN	FP	FN	Total Size	Sensitivity [%]	Specificity [%]	Accuracy [%]
Validation	6355	6099	76	45	12575	99.297	98.769	99.038
Testing	1475	1630	22	16	3143	98.927	98.575	98.791

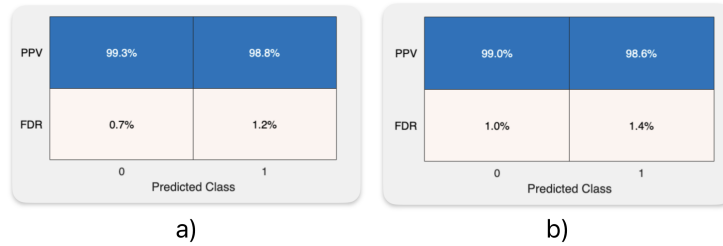


Figure 4.14: Movement identification - Confusion matrices a) Validation; b) Testing

The number of movement observations was also evaluated, as it resembles one of the movement measurements provided in the statistics overview page of the analyser application (see Section 3.6.2). Table 4.14 displays the validation and testing sets' results.

Table 4.14: Movement identification measurement results - Movement observations

Category	Measurement	Hypothesis	Ground Truth	Accuracy [%]
Validation	Movement Observations	6355	6400	99.297
Testing	Movement Observations	1475	1491	98.927

The maximum and average acceleration measurements, 'Max Gs' and 'Average Gs', are

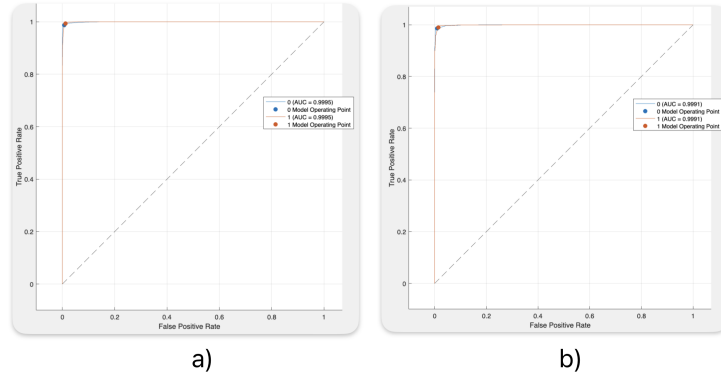


Figure 4.15: Movement identification - ROCs a) Validation; b) Testing

calculated through the entire recording, and not just from the movement regions. Therefore, the movement identification classifier's accuracy does not affect its calculations. Table 4.15 merely represents the maximum and average acceleration from the test dataset. The results were visually inspected and were confirmed to represent the true values (see Figure 4.16).

Table 4.15: Movement identification measurement results - Maximum and average acceleration

Max G's [$\text{g's} = 9.807 \text{ m/s}^2$]	Average G's [$\text{g's} = 9.807 \text{ m/s}^2$]
2.936	0.221

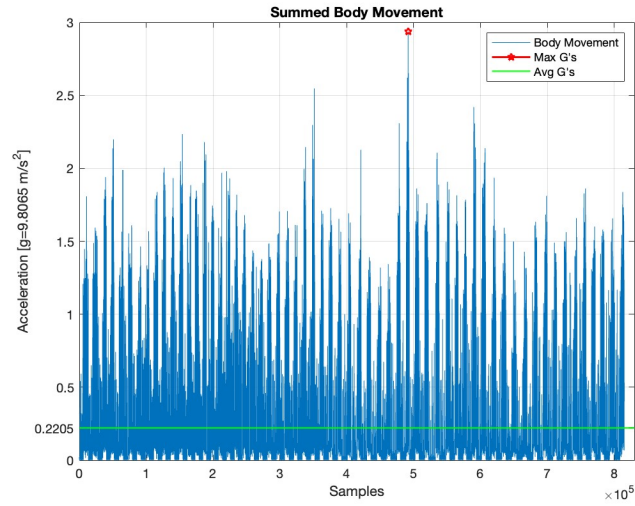


Figure 4.16: Movement measurement - Evaluation

4.5 Conclusion

This study aimed to automate speech-language pathologists' (SLPs) speech interaction analysis procedures. Two systems were developed to accomplish this: a speaker diarisation system and a behaviour analysis system. This section addressed how well each subsystem performed in its respective task. As well as, the recorder and analyser applications that encapsulate them. Lastly, the overall approach of this study is validated.

4.5.1 Validation: Speaker Diarisation

With the evaluation of the **speaker diarisation system** limited to a single self-made audio file, it can not be accurately compared to other systems. It did, however, indicate being continuously successful in diarising five unknown speakers with the absence of overlapping speech. The algorithm proved quite time-consuming, with approximately 1 minute of execution per 1 minute of speech present. The time consumption is insignificant as the analyses can be conducted in the background, separate from the SLP's contact sessions. It is thereby confirmed that the speaker diarisation system can provide an automated means of identifying who spoke when without prior knowledge of the speakers present. It also offers pre-calculated metrics such as turn-taking and longest phrase, all of which provide essential information to an SLP when conducting speech interaction analyses.

4.5.2 Validation: Beat Detection

The **beat detection algorithm** was evaluated on a small subset of the MIT-BIH Arrhythmia dataset [3]. The algorithm struggled compared to other state-of-the-art algorithms when it comes to 'diseased' (containing arrhythmias) signals. However, the algorithm did prove to be sufficient when extracting HRV data from healthy and clean ECG signals, such as those within the MIT-BIH Arrhythmia dataset (e.g., recording 113m) and those collected by a well-attached Polar H10 device (see Table 4.9). This is important to ensure accurate stress identification for this study.

Accurate heart rate measurements, such as minimum heart rate, maximum heart rate, and average heart rate, were also provided, allowing for quick person-specific evaluation.

4.5.3 Validation: Stress Identification

The **stress identification algorithm** was evaluated on a small dataset containing public speaking and arithmetic stress, closely mimicking the stress that might arise during an SLP contact session. The dataset is, however, limited not just in size but also in age diversity. Most of the subjects were in their mid-twenties, providing little to no information on the usability of the stress identification algorithm among young children (Age<13) and older adults (Age>65). Algorithmic adjustments were, however, made to minimise the effect of age on stress identification (see Section 3.4), but require further research in this regard.

The stress identification algorithm obtained high validation and testing accuracies with

minimum time consumption given the subset. With that said, the stress identification algorithm provides SLPs with an automated means of identifying person-specific momentary stress regions, allowing them to draw conclusions about the individual's behaviour at any given time.

The number of stress observations is also provided as a quick measurement for evaluation in this regard. For instance, if an individual has a high number of stress observations, one would conclude that the session was stressful for the individual.

4.5.4 Validation: Movement Identification

The **movement identification algorithm** proved to detect movement successfully using merely three features. It was evaluated on a relatively large dataset containing various basic activities, such as sitting, walking, and transitions between the two, mimicking expected activities within an SLP's contact session. The algorithm exceeds due to its simplicity - binary classification of movement and non-movement. This algorithm provides an automated means of identifying momentary activeness, allowing the SLP to conclude about an individual's listening/concentration capabilities.

The number of movement observations, maximum acceleration, and average acceleration measurements were also provided to allow for quick person-specific evaluations.

4.5.5 Validation: Applications

The **data-capturing application** provides an easy means of recording audio and behavioural (ECG and accelerometer) data, allowing it to be stored and easily analysed later within the **analyser application**. The analyser application encapsulates the speaker disambiguation, stress identification, and movement identification subsystems and provides visual representations of their results and metrics to draw conclusions easily. Furthermore, it allows the user to listen to the entire recording session or solely to a specific speaker. These audio files, alongside graphical illustrations of the results, can be extracted and saved offline to be reassessed at any moment without having to repeat the analysis process.

4.5.6 Validation: Overall

The applications created in this study provide an automated means of recording and analysing speech interactions. With the success they obtained, the procedures usually conducted by an SLP for speech interaction analyses can now be done by a computer, freeing them to address other responsibilities.

Chapter 5

Conclusion

This chapter provides the general results of this study, its limitations, future improvements, and finally, its possible benefits.

5.1 Summary and Results Overview

This study set out to automate speech-language pathologists' procedures to assess speech interactions between a parent and their child. Special focus was placed on speech analysis and behavioural analysis, identifying who spoke when and identifying person-specific stress and movement, respectively.

To accomplish this, a speaker diarisation system and stress and movement identification algorithms were created. They are encapsulated by an easy-to-use application called the 'analyser'. The **analyser application** is provided with patient data, namely patient information, raw ECG and accelerometer data extracted from Polar H10 devices, and a single-channel audio recording. This data is recorded by a separate 'recorder' application. The **recorder application** stores the data offline in a single folder to be used by the analyser application whenever applicable.

The **speaker diarisation system** can identify who is speaking when without any prior knowledge of the speakers or their speech characteristics. It does so with perfect clustering accuracy (no speaker is mistaken for someone else) and a diarisation error rate (DER) of 5.207%, stemming mainly from quantisation errors - speech segments ignored because they are shorter than the minimum utterance duration. Moreover, the speaker diarisation system allows for the calculation of commonly used evaluation metrics, typically applied by SLPs, to assess an individual's speech complexity and efficiency. Some of these metrics are pre-calculated by the analyser application and readily made available to its user, including turn-taking, total speech duration, longest phrase, and longest spoken turn.

Using a **beat detection algorithm**, the stress identification algorithm first extracts

HRV data from the raw ECG signals. The beat detection algorithm, in return, detects the absolute maximum peaks within the QRS complexes of an ECG signal. By doing this, the heart rate and, more importantly, the heart rate variability can be calculated. The algorithm obtained an average detection error rate (DER) of 0.877% on a selected few of the ‘normal’ recordings of the MIT-BIH Arrhythmia dataset, [3], and a perfect DER of 0% on a single test conducted with the Polar H10 device. Furthermore, beat detection allows for calculating a few useful heart rate metrics, such as minimum, maximum, and average heart rate.

After the HRV data has been extracted, the stress can be identified. The **stress identification algorithm** possesses a cubic SVM classifier to distinguish between stress and non-stress. It obtained a validation accuracy of 93.939% from a subset containing 165 observations (40 stress observations) and a testing accuracy of 100% from a subset containing 18 observations (5 stress observations). The algorithm’s results were, furthermore, correlated to those obtained by the Kubios software, getting matching results. A quick reference to the number of stress observations identified per individual is also provided to the user of the analyser application.

The **movement identification algorithm**, on the other hand, possesses a linear SVM classifier to distinguish between movement and non-movement. It obtained a validation accuracy of 99.038% from a subset containing 12575 observations (6400 movement observations) and a testing accuracy of 98.791% from a subset containing 3143 observations (1491 movement observations). Furthermore, the movement identification algorithm allows for calculating the maximum and average acceleration and the number of movement observations per individual.

5.2 Limitations

The limitations of the systems and algorithms used in this study are described here.

5.2.1 Speaker Diarisation System

The speaker diarisation system identifies who spoke when without any prior knowledge of the speakers. This is one of the system’s most significant benefits. However, it has a drawback. The system randomly assigns cluster names to the speakers and has to be reassigned manually. Since the system is unaware of who is who, it requires manual intervention to assign names to each speaker correctly. For example, given two speakers named John and Jane, the system may randomly assign the number 1 to Jane’s speech segments and the number 2 to John’s. To correctly identify each speaker, the assigned numbers must be updated manually to state that cluster name ‘1’ is Jane and cluster name ‘2’ is John. (**Manually reassigning cluster names to their respective speakers.**)

Despite having to reassign cluster names, the speaker diarisation system performed exceptionally well. However, it should be noted that the evaluation of the system is somewhat

limited. The evaluation was conducted on a single audio file comprising a 9-minute recording of 5 speakers reading from books without overlapping speech. This assessment restricts the evaluation of the system in more complex settings. Additionally, since a standardised dataset was not used, the system cannot be compared with others. (**Limited evaluations.**)

The system is conducted offline and consumes approximately 1 minute of execution time to 1 minute of speech to be processed. Stagnating (parts of) the central processing unit of the host device and consuming significant amounts of memory. (**Computationally expensive.**)

The speaker diarisation system automates most speech analysis procedures an SLP conducts; however, it can not produce transcripts. Transcriptions often come into account when performing language sampling. To produce transcriptions of an SLP's contact session, a more complex algorithm would have to be employed alongside the speaker diarisation system, namely automatic speech recognition (ASR). (**Cannot produce transcripts - not ASR.**)

5.2.2 Behavioural Algorithms

The ability to detect stress and movement is limited to the number of individuals wearing a Polar H10 device within proximity (BLE range; less than 100m). Additionally, these individuals must be present at the beginning of the session to connect their devices to the recorder application. Once an individual leaves the session, their device cannot be reconnected to resume their measurements. (**Limited by the wearable Polar H10 devices.**)

5.2.3 Applications

There are occurrences where the speaker diarisation system or the stress and movement algorithms might miss and falsely identify labels (i.e., speaker, stress, and movement). Unfortunately, to correct the mistakes, the system would have to be processed again, and hopefully, it would provide a better result. The analyser application does not offer a means to label the data manually. (**Does not provide manual labelling.**)

The data collected by the recorder, though stored locally, does not ensure data security. The data collected in an SLP's office can be sensitive and should be protected. (**Data security was excluded from this study.**)

5.3 Future Improvements

Improvements that could be adapted to this study are described next.

5.3.1 Speaker Diarisation System

It is worth noting that there are more advanced speaker diarisation systems available that might be better suited to achieving the study's end goals. For example, deep neural networks (DNN) embeddings, known as x-vectors, have proven to be more robust than most systems in recent years and have been widely used in state-of-the-art speaker diarisation systems, [40, 41]. **(Replacing the speaker diarisation system with a system based on x-vectors.)**

Furthermore, the system utilises a single-channel audio recording to avoid additional sensor captures. However, suppose a multi-channel approach is adopted by placing lapel microphones on each significant individual. In that case, a more complex speaker diarisation approach can be developed, which is less likely to suffer from overlapping speech. This is because other noises or speakers can be filtered out by employing a technique known as time difference of arrival (TDOA), which can exclude sounds based on their time latency. **(Multi-channel approach to improve robustness against noise and overlapping speech.)**

Lastly, once the speaker diarisation system has been settled and enough speech data has been collected, the system can be adapted to an ASR system. This will provide speaker diarisation results and transcriptions of an SLP's sessions. **(Add ASR.)**

5.3.2 Behavioural Algorithms

Although the beat detection algorithm performed satisfactorily on healthy ECG signals, more complex beat detection algorithms exist, which are more robust to artefacts and arrhythmias, such as [128]. Due to the importance of the beat detection algorithm in the overall classification of stress and calculation of other heart rate-related metrics, adopting an improved algorithm could significantly improve the overall robustness of this study. **(Improve beat detection algorithm to that of [128].)**

Movement is identified because it can provide information on the overall concentration span of an individual. Moreover, it also provides information on how trustworthy an identified stress region is. For instance, if the individual exerted a high amount of energy in a short period, the stress identifier could have incorrectly identified the change in HRV as stress. However, except for physical exertions, posture also plays a significant role in heart rate changes. For instance, there is a significant decrease in average heart rate when changing posture from standing to seated. Adapting the movement identification algorithm to a postural classification algorithm could provide additional information on the individual's movements during the session and offer a more complex stress identification algorithm based on postural changes. **(Movement identification to postural classification; Postural orientated stress identification.)**

5.3.3 Applications

The idea behind the applications was not only to automate the analysis procedures of an SLP but also to improve continuously. The goal was to capture annotated data that can be used to improve the automation algorithms further and to provide additional data that could enhance the research in the field of speech-language pathology. However, given that the current algorithms are not always perfect, adding the ability to annotate the recorded data manually could correct these imperfections, providing improved annotated data to be stored and used later to improve the applications and algorithms. (**Add the ability to annotate recorded data manually.**)

5.4 Possible Benefits

Automating the speech interaction analysis procedures of an SLP can potentially save wasted time spent on prerequisites such as identifying when each individual/patient speaks. By employing a machine to do this automatically, SLPs would theoretically devote more time to patient care by extending sessions or increasing the number of sessions they conduct daily. Moreover, quick person-specific metrics and annotated data could lead to faster and more accurate diagnoses. This could result in earlier interventions, improving patient communication abilities and overall health outcomes. (**Provide SLPs with more time; increased patient care.**)

By providing synchronised annotated data, SLPs can for the first time, correlate stress, movement, and speech to one another in a single application. This allows for analyses such as matching stress to another individual's speech, for instance, comparing the stress of a child to the harsh words of an adult. (**New analysis options - Simultaneous assessments.**)

The applications also allow for the data and analyses thereof to be saved offline, providing the benefit of creating patient catalogues to be reassessed (or assessed as a whole) at any given point. (**Creating patient catalogues.**)

Appendix A

Polar H10 PMD GATT

A.1 PMD services

Compiled from ‘Polar Measurement Data Specification for 3rd Party’ document [144].

The PMD GATT service and its characteristics are provided in Table A.1, obtained from [144]. The PMD GATT service defines various measurement types, each with its own frame and setting types for the PMD control point characteristic.

Table A.1: PMD GATT Service and its Characteristics

Service Name	Characteristic Name	Property	Security Permission	UUID
PMD Service	NA			FB005C80-02E7-F387-1CAD-8ACD2D8DF0C8
	PMD Control Point	Read, Write, Indicate	None	FB005C81-02E7-F387-1CAD-8ACD2D8DF0C8
	PMD Control Point Client Characteristic Configuration Descriptor	Read, Write	None	
	PMD Data MTU Characteristic	Notify (opt. Indicate)	None	FB005C82-02E7-F387-1CAD-8ACD2D8DF0C8
	PMD Data MTU Client Characteristic Configuration Descriptor	Read, Write	None	

The ECG and acceleration measurements are referred to as measurement types 0 and

2, respectively, in the GATT PMD service. This is illustrated in Table A.2 with their corresponding frame types in Table A.3 and Table A.4.

Table A.2: PMD measurement types

Measurement Type	Description
0x0	ECG
0x2	Acceleration

Table A.3: ECG frame type

Frame type	Size	Unit	Description
0	3 Bytes	μV	Electrocardiogram
1...255			Reserved for future use (RFU)

Table A.4: Acceleration frame type

Frame type	Size	Unit	Description
0	3 Bytes	mG	x, y, z - 8-bit
1	6 Bytes	mG	x, y, z - 16-bit
2	9 Bytes	mG	x, y, z - 24-bit
128	n	mG	Delta frame
3...127, 129...255			Reserved for future use (RFU)

Possible PMD setting types for ECG and acceleration frame types are illustrated in Table A.5. This is used to set each frame type's parameters.

Table A.5: PMD setting types

Setting Type	Description	ECG	Acceleration
0	Sample rate	130 Hz	25, 50, 100, 200 Hz
1	Resolution	14 bits	16 bits
2	Range	-	2, 4, 8 G
3	Range (Milli-unit)	-	-
4	Channels	-	-
5	Factor	-	-

A.2 PMD commands

The PMD services have a set of procedures for communication and data retrieval from Polar devices. These procedures are explained below.

Figure A.1 illustrates the **prerequisite** stage required to establish a connection and to communicate with a Polar H10 device.

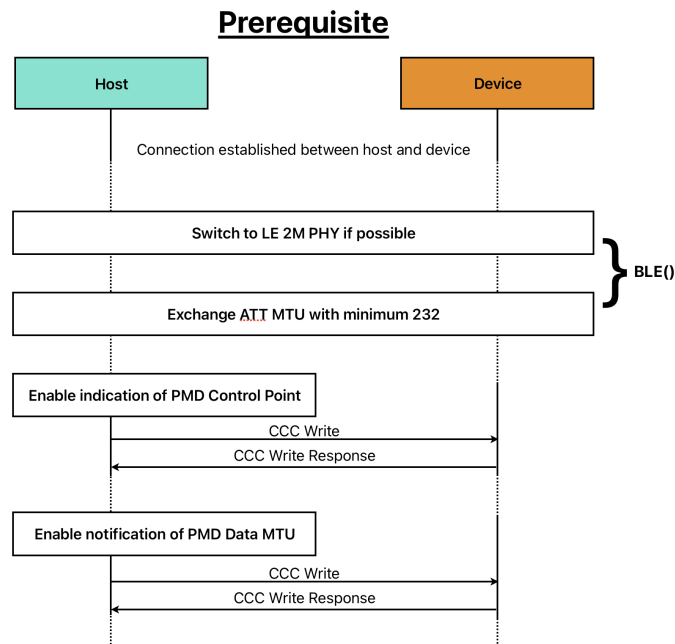


Figure A.1: PMD prerequisites.

The PMD service also allows a feature list request indicating which features are available given the Polar device. This is done by sending a read request to the PMD control point characteristic. An illustration for the Polar H10 feature request is provided in Figure A.2. Moreover, the command to obtain the available setting types for these features is illustrated in Figure A.3.

To begin transmitting a given measurement type, its corresponding "start stream" command is sent to the PMD Control Point of the required Polar device. An illustration of the **start** stream request is provided in Figure A.4.

To **stop** the streaming process the stop command is then sent to the PMD Control Point for the given measurement type. An illustration hereof is provided in Figure A.5.

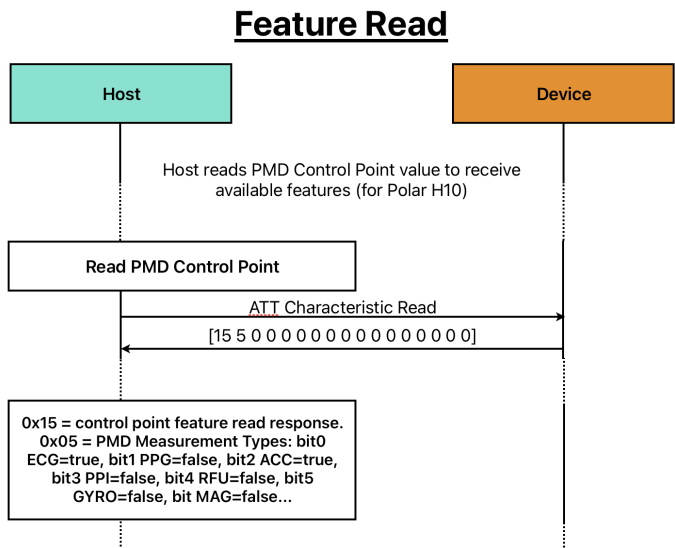


Figure A.2: PMD read available features.

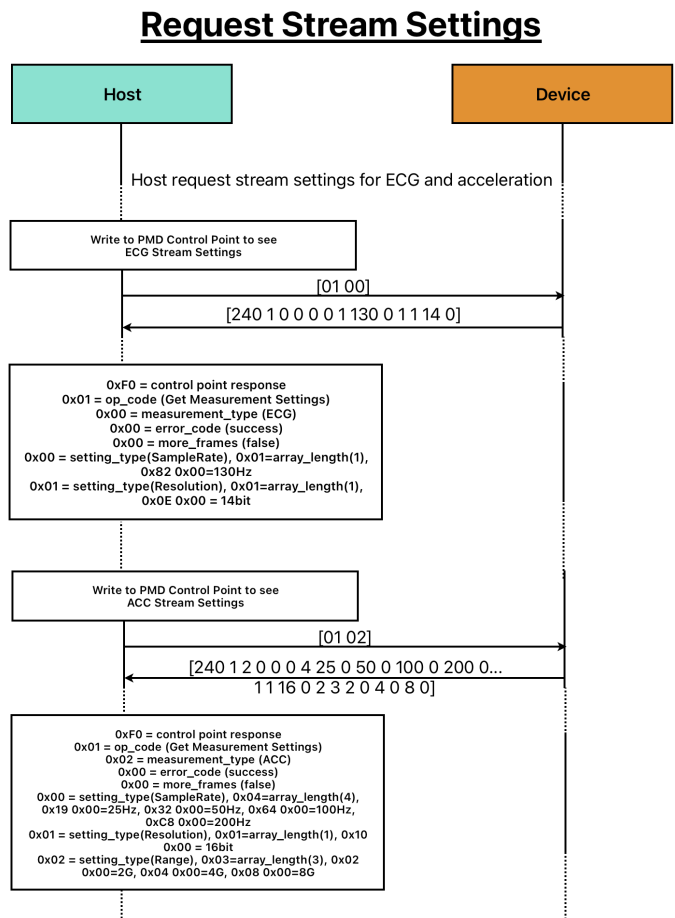


Figure A.3: PMD request available setting types.

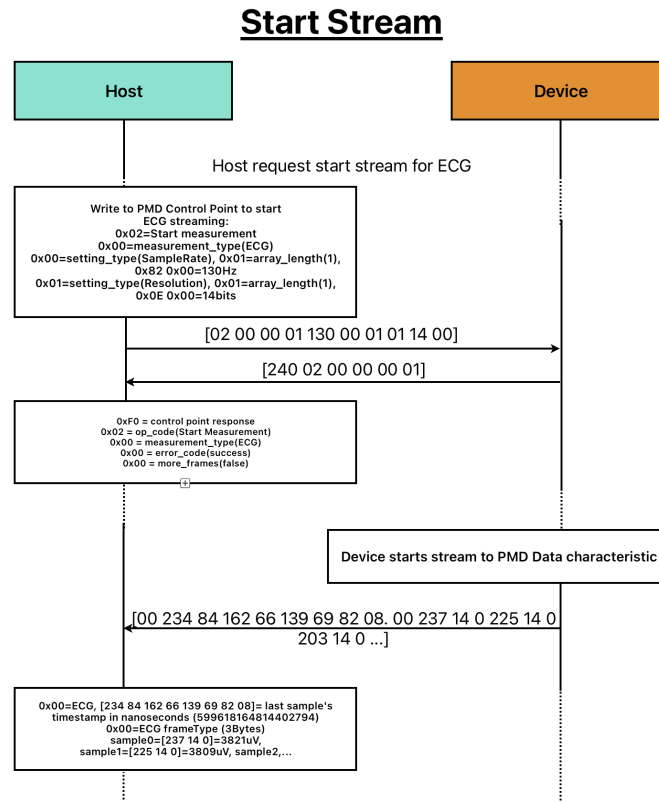


Figure A.4: PMD start streaming.

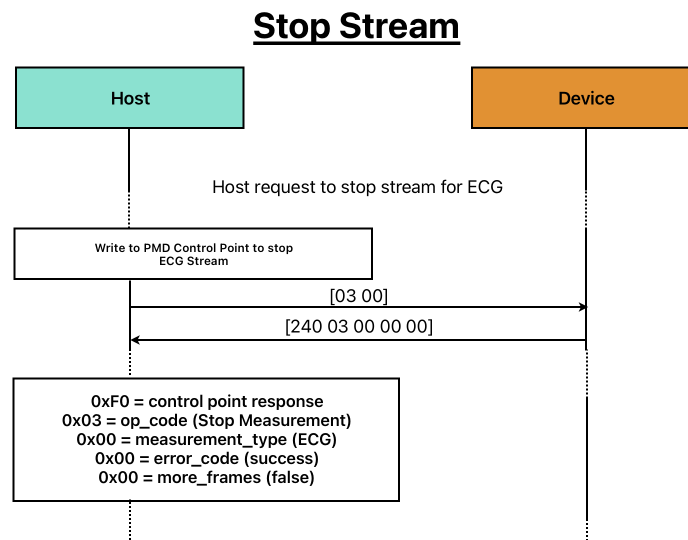


Figure A.5: PMD stop streaming.

Appendix B

Viterbi Algorithm (Log-interpretation)

Here the Viterbi-log algorithm is provided. The use of log-likelihoods provides the advantage of avoiding floating point underflow problems as well as performing exponentiations when computing Gaussians, creating a more computationally efficient algorithm.

HMM Parameters:

$\pi = \{\pi_1, \pi_2, \dots, \pi_N\} \rightarrow$ initial probability distribution over N states.

$Q = \{q_1, q_2, \dots, q_N\} \rightarrow$ a set of N-states.

$A = \{a_{11}, \dots, a_{ij}, \dots, a_{NN}\} \rightarrow$ transition probability matrix. $\sum_{j=1}^N a_{ij} = 1, \forall i$

$O = \{o_1, o_2, \dots, o_T\} \rightarrow$ a sequence of T observations.

$B = b_i(o_t) \rightarrow$ a sequence of observation probabilities, also called emission probabilities, each expressing the probability of an observation o_t being generated from a state i .

The HMM string architecture depicted in Figure 3.11 creates the transition matrix, A, and the initial probability distribution, π . The observation vector of each state is represented by a GMM (see Equation B.1 below). The matrices consist of T-observations and N-states, provided as rows and columns respectively.

$$b_j(\underline{y}) = \sum_{m=1}^M c_{jm} \mathcal{N}\left(\underline{y}; \underline{\mu}^{(jm)}, \underline{\Sigma}^{(jm)}\right); \quad (\text{B.1})$$

$c_{jm} \rightarrow$ prior probability for component m of state s_j . $\sum_{m=1}^M c_{jm} = 1, c_{jm} \geq 0$

$\underline{y} \rightarrow$ [feature vector] - single observation i.e. o_t .

$\underline{\mu}, \underline{\Sigma} \rightarrow$ mean and covariance matrix.

Viterbi Parameters:

$\delta_t(i) \rightarrow$ most likely path of arriving at state i in timestep t .

$\psi_t(i) \rightarrow$ previous state. What state did we come from given the current state i ?

$P^* \rightarrow$ maximum probability

$q_T^* \rightarrow$ final state with a valid sequence of states to arrive here.

Initialisation:

$$\begin{aligned}\delta_1(i) &= \log(\pi_i b_i(O_1)) \\ &= \log(\pi_i) + \log(b_i(O_1))\end{aligned}\tag{B.2}$$

$$\psi_1(i) = 0, \text{ no previous state}\tag{B.3}$$

Inductive Step:

$$\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) + \log(a_{ij})] + \log(b_j(O_t)),\tag{B.4}$$

$$\psi_t(j) = \arg \max_{1 \leq i \leq N} [\delta_{t-1}(i) + \log(a_{ij})],\tag{B.5}$$

where, $2 \leq t \leq T$; $1 \leq j \leq N$.

Termination:

$$P^* = \max_{1 \leq i \leq N} [\delta_T(i)]\tag{B.6}$$

$$\begin{aligned}q_T^* &= \arg \max_{1 \leq i \leq N} [\delta_T(i)] \\ &= \psi_{t+1}(q_{t+1}^*)\end{aligned}\tag{B.7}$$

Appendix C

Algorithm

A pseudo representation of the speaker diarisation system is provided here.

```
%% PARAMETERS:
Fs = 16e3; (Sampling rate);
windowDuration = 30ms; hopDuration = 10ms;
overlap = windowDuration - hopDuration = 20ms;
SegmentSize = 5 min;
Create 5-minute audio segments (to be stored as a tall
    array);
Start Iteration for each 5-minute segment: IterChunks

%% PRE-PROCESSING:
% Gather a 5-minute audio segment (in-memory):
audioPre = gather(audioSegment{iter});
% Normalise:
audioPre = audioPre./max(abs(audioPre));
% Perform speech enhancement:
audioPre = v_ssubmmse(audioPre,Fs);
% Mitigate distant speakers (Non-linear function):
audioPre = sign(audioPre).*(abs(audioPre).^0.75);
% Add white Gaussian noise:
audioPre = awgn(audioPre,120);

%% VAD:
```

```

% Perform Voice Activity Detection:
VADIdx = VAD_func(audioPre, Fs, windowParameters);
speechIdx = VADIdx(Speech);
%% INITIAL SEGMENTATION:
% Perform Initial Segmentation:
[speakerIdx, Components, g] = iniSeg(audioPre, Fs,
    speechIdx, windowParameters, minDuration=1);

%% CORE ALGORITHM:
% Limit Speech segments >= 2.5 seconds:
filtSpeechIdx = speechIdx(Duration>=2.5 seconds);
Iterate till the stopping condition is met: IterMerge
% Train: Obtain Model for Viterbi Segmentation
HMM = trainFunc(audioPre, Fs, Comps, minSpeechDuration
    =2.5, speakerIdx, windowParameters, g);
% Segmentation:
Segments = segFunc(HMM, shortTermFeat, filtSpeechIdx,
    windowParameters);
% Merge:
Execute delta-BICs for each cluster pair;
% Stopping Condition:
% Extract delta-BICs that are less or equal to zero:
mergeClusts = deltaBIC(val<=0);
If mergeClusts is not empty: The stopping condition is
    not met
Merge smallest mergeClusts and re-iterate;
If mergeClusts is empty: The stopping condition is met
Break iteration;
End IterMerge
End IterChunks

%% FINAL SEGMENTATION:
Merge the audio segments and their results;
Run the Core Algorithm once more for the merged results;

```

```
% Adjust minimum speech duration:
minSpeechDuration = 1.5;
% Limit Speech segments >= 1.5 seconds:
filtSpeechIdx = speechIdx(Duration>=1.5 seconds);
% Train: Obtain Model for Viterbi Segmentation
HMM = trainFunc(audioPre, Fs, Comps, minSpeechDuration
    =1.5, speakerIdx, windowParameters, g);
% Segmentation:
Segments = segFunc(HMM, shortTermFeat, filtSpeechIdx,
    windowParameters);

%% OUTPUT:
speakerDiarisationOutput = Segments;
```

Appendix D

WESAD PKL-To-CSV Python Script

```
# Format WESAD Data: Pickel file to csv.
import pandas as pd
import numpy as np
import glob

listFiles = glob.glob('/Datasets/WESAD/S*/S*.pkl')
for i in listFiles:
    df = pd.read_pickle(i)

    data = np.concatenate((df['signal']['chest']['ACC'],
df['signal']['chest']['ECG'],
df['signal']['chest']['Resp']),axis=1)

    df2 = pd.DataFrame(data,
columns=('X-Axis','Y-Axis','Z-Axis','ECG','Resp'))
    df2['Labels'] = df['label']
    df2.drop(df2[(df2['Labels']==0) | (df2['Labels']==5) |
(df2['Labels']==6) | (df2['Labels']==7)].index,
inplace=True)
# The code below adds label names to the corresponding
# label integer:
```

```
#df2['Label_Names'] = np.where(df2['Labels']==1,  
# 'Baseline', np.where(df2['Labels']==2, 'Stress',  
# np.where(df2['Labels']==3, 'Amusement',  
# np.where(df2['Labels']==4, 'Meditation', False))))  
  
fileLoc = '/Datasets/WESAD/CSV_Files/'  
fileNum = df['subject'] + '.csv'  
fileName = fileLoc + fileNum  
  
df2.to_csv(fileName, index=False)
```

Appendix E

MIT-BIH-to-MATLAB Code

```
% Matlab script to create .mat files from wfdb .dat files of
    the MIT-BIH
% Arrhythmia Database
recordNum = load('RECORDS');
recordLoc = 'mitdb/';
recordName = convertStringsToChars(strcat(recordLoc,string(
    recordNum)));
BeatAnnList = {'N','L','R','A','a','J','S','V','F','e','j','E',
    '','/','f','Q'};
fileLoc = '/database/';
%%
%temp = cell(48,1);
for i=1:length(recordName)
    [Sig,Fs,Time] = rdsamp(recordName{i},1);
    [Ann,AnnType] = rdann(recordName{i},'atr');

    BeatAnnTable = table(Ann,AnnType,'VariableNames',{'
        AnnTime','AnnType'});
    BeatAnnTable(not(ismember(BeatAnnTable.AnnType,
        BeatAnnList))),:) = [];

    m = struct('Fs',Fs,'Sig',Sig,'Time',Time,'BeatAnnTable',
        BeatAnnTable);
```

```
    fileName = strcat(fileLoc,recordName{i},'m.mat');
    save(fileName,'-struct','m');
    %temp{i} = {AnnType(not(ismember(AnnType,BeatAnnList)))};
end
%%
% figure;
% plot(Time,Sig);
% hold on;
% grid on;
% plot(Time(BeatAnnTable.AnnTime),Sig(BeatAnnTable.AnnTime),'
    ro','MarkerSize',5);
% hold off;
```

Appendix F

HAPT-to-MATLAB Code

```
% HAPT_To_MAT creates a feature and label table for Human
    Activity
% Recognition (HAR) ML capabilities. The dataset used is the
    Smartphone-Based
% Recognition of Human Activities and Postural Transitions
    Data Set
% Version 2.1 from:
% =====
% Jorge L. Reyes-Ortiz(1,2), Davide Anguita(1), Luca Oneto(1)
    and Xavier Parra(2)
% 1 - Smartlab, DIBRIS - University degli Studi di Genova,
    Genoa (16145), Italy.
% 2 - CETpD - Universitat Politècnica de Catalunya. Vilanova
    i la Geltrú (08800), Spain
% har '@' smartlab.ws
% www.smartlab.ws
% =====
clear all;
%% Load Data From HAPT Dataset:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fileLoc = '/Datasets/HAPT Data Set/RawData/';
cd(fileLoc);
```

```

fileList = dir('acc_exp*.txt');
fileList = {fileList.name}';
fileName = strcat(fileLoc,fileList);

% Load label data:
fileLabel = 'labels.txt';
fileName2 = strcat(fileLoc,fileLabel);
rawLables = readmatrix(fileName2); % Obtain mixed labels and
    setpoints per file.

Fs = 50; % Sampling rate
rawAccData = [];

for i=1:length(fileName)
    temp = readmatrix(fileName{i});
    labelarr = zeros(length(temp),1);
    % Obtain only the required label data for the current
        file:
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    curExpSub = str2double(regexp(fileName{i},'\d*','Match'))
        '
    labels = rawLables(rawLables(:,1)==curExpSub(1) &
        rawLables(:,2)==curExpSub(2),3:5);
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Create label array: Match labels to corresponding data
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    for j=1:length(labels)
        labelarr(labels(j,2):labels(j,3)) = labels(j,1);
    end
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Remove unlabelled data:
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    temp(labelarr(:,1)==0,:) = [];
    labelarr(labelarr(:,1)==0,:) = [];

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
temp(:,end+1) = labelarr; % Add label to rawAccData
    matrix
rawAccData = cat(1,rawAccData,temp); % Concat raw
    triaxial accelerometer data (x,y,z)
end

clear fileLoc fileList fileName fileLabel fileName2 rawLables
    temp labelarr curExpSub labels
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Move to current running script:
filePath = matlab.desktop.editor.getActiveFilename;
cd(fileparts(filePath));
%% Perform pre-processing:
[totalAcc,bodyMovementAcc,gravityAcc,Fs] =
    preProcessingAcc_func(rawAccData(:,1:3),Fs);
%% Add Movement [1] and Non-movement [0] labels:
labelarr = rawAccData(:,4);
labelarr(labelarr~=4 & labelarr~=5 & labelarr~=6,2) = 1;
%% Create Acceleration Feature Dataset:
tic
[featAcc,~] = createFeaturesAcc_func(bodyMovementAcc(:,1:3),
    labelarr(:,2),Fs);
toc
cd('..');
save('database/HAPT/featAcc_Rev2.mat','featAcc');
cd(fileparts(filePath));

```

Appendix G

Master Code

The code to this study can be found at https://github.com/28762193/Master_Stress_SpeakerDiarisation

Bibliography

- [1] “Common Voice,” Available: <https://commonvoice.mozilla.org/en> (Accessed Jun. 06, 2023).
- [2] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Librispeech: An ASR corpus based on public domain audio books.” *IEEE*, 4 2015, pp. 5206–5210.
- [3] G. Moody and R. Mark, “The impact of the MIT-BIH Arrhythmia Database,” *IEEE Engineering in Medicine and Biology Magazine*, vol. 20, pp. 45–50, 2001.
- [4] P. Schmidt, A. Reiss, R. Duerichen, C. Marberger, and K. V. Laerhoven, “Introducing WESAD, a Multimodal Dataset for Wearable Stress and Affect Detection.” *ACM*, 10 2018, pp. 400–408.
- [5] J. Reyes-Ortiz, D. Anguita, L. Oneto, and X. Parra, “Smartphone-based recognition of human activities and postural transitions,” 2015.
- [6] N. S. N. M. Sainain, R. Omar, H. Ismail, N. Mamat, and R. Abdullah, “Parental knowledge and development of languages and literacy, communication and socializations in the early childhood education,” *International Journal of Psychosocial Rehabilitation*, vol. 24, pp. 2070–2080, 2020.
- [7] A. I. Pinto, M. Pessanha, and C. Aguiar, “Effects of home environment and center-based child care quality on children’s language, communication, and literacy outcomes,” *Early Childhood Research Quarterly*, vol. 28, pp. 94–101, 1 2013.
- [8] B. Hart and T. R. Risley, *Meaningful differences in the everyday experience of young American children*. Paul H Brookes Publishing, 1995.
- [9] M. A. Fields-Olivieri, P. M. Cole, and C. K. Roben, “Toddler emotion expressions and emotional traits: Associations with parent-toddler verbal conversation,” *Infant Behavior and Development*, vol. 61, 11 2020.
- [10] H. G. Kim, E. J. Cheon, D. S. Bai, Y. H. Lee, and B. H. Koo, “Stress and heart rate variability: A meta-analysis and review of the literature,” pp. 235–245, 3 2018.
- [11] American Speech-Language-Hearing Association, “Scope of Practice in Speech-

- Language Pathology [Scope of Practice],” Available: www.asha.org/policy/ (Accessed Jun. 30, 2023), 2016.
- [12] “Assessment Tools, Techniques, and Data Sources,” Available: <https://www.asha.org/practice-portal/clinical-topics/late-language-emergence/assessment-tools-techniques-and-data-sources/> (Accessed Jul. 07, 2023).
- [13] M. Pillay, R. Tiwari, H. Kathard, and U. Chikte, “Sustainable workforce: South african audiologists and speech therapists,” *Human Resources for Health*, vol. 18, p. 47, 12 2020.
- [14] —, “Sustainable workforce: South african audiologists and speech therapists,” *Human Resources for Health*, vol. 18, p. 47, 12 2020.
- [15] *Oxford Dictionary of English*, 3rd ed. Oxford University Press, 2015.
- [16] L. Richard and D. Charbonneau, “An introduction to E-Prime,” *Tutorials in Quantitative Methods for Psychology*, vol. 5, pp. 68–76, 9 2009.
- [17] “SALT Home Page,” Available: <https://www.saltsoftware.com/> (Accessed Jul. 12, 2023).
- [18] C. L. F. Boulder, “Clinical Applications of the Breakthrough LENA System: Monitoring Language in the Natural Environment of the Child,” Available: https://www.lena.org/wp-content/uploads/2016/07/LTR-09-1_Clinical_Applications.pdf, 11 2008.
- [19] “Understanding lena technology,” Available: <https://www.lena.org/technology/> (Accessed Apr. 03, 2023).
- [20] “Top 5 pediatric speech therapy software tools and apps — speech pathology degrees — find an online slp program,” Available: <https://www.speechpathologygraduateprograms.org/2017/12/top-5-pediatric-speech-therapy-software-tools-and-apps/> (Accessed Jul. 12, 2023).
- [21] “Nuance and Microsoft Announce the First Fully AI-Automated Clinical Documentation Application for Healthcare - Mar 20, 2023,” Available: <https://news.nuance.com/2023-03-20-Nuance-and-Microsoft-Announce-the-First-Fully-AI-Automated-Clinical-Documentation-Application-for-Healthcare/> (Accessed Jul. 18, 2023).
- [22] R. Prenger, R. Valle, and B. Catanzaro, “Waveglow: A Flow-based Generative Network for Speech Synthesis.” *IEEE*, 5 2019, pp. 3617–3621.
- [23] L. R. Bahl, F. Jelinek, and R. L. Mercer, “A Maximum Likelihood Approach to Continuous Speech Recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-5, pp. 179–190, 3 1983.

- [24] X. A. Miro, S. Bozonnet, N. Evans, C. Fredouille, G. Friedland, and O. Vinyals, “Speaker Diarization: A Review of Recent Research,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, pp. 356–370, 2 2012.
- [25] S. V. Blackley, J. Huynh, L. Wang, Z. Korach, and L. Zhou, “Speech recognition for clinical documentation from 1990 to 2018: a systematic review,” *Journal of the American Medical Informatics Association*, vol. 26, pp. 324–338, 4 2019.
- [26] H. Purwins, B. Li, T. Virtanen, J. Schlüter, S. yiin Chang, and T. Sainath, “Deep learning for audio signal processing,” 4 2019.
- [27] A. B. Nassif, I. Shahin, I. Attili, M. Azzeh, and K. Shaalan, “Speech recognition using deep neural networks: A systematic review,” *IEEE Access*, vol. 7, pp. 19 143–19 165, 2019.
- [28] “Linguistic Data Consortium,” Available: <https://catalog.ldc.upenn.edu> (Accessed Jul. 19, 2023), 6.
- [29] “Guidelines for practice in a culturally and linguistically diverse south africa 2019 1 professional board for speech, language and hearing professions guidelines for practice in a culturally and linguistically diverse south africa,” 2019.
- [30] T. J. Park, N. Kanda, D. Dimitriadis, K. J. Han, S. Watanabe, and S. Narayanan, “A Review of Speaker Diarization: Recent Advances with Deep Learning,” 1 2021.
- [31] “AMIDA: Augmented Multiparty Interaction with Distance Access,” Available: <https://www.cstr.ed.ac.uk/research/projects/amida/> (Accessed Jun. 20, 2023).
- [32] “Computers in the human interaction loop (chil) — chil — project — fact sheet — fp6 — cordis — european commission,” Available: <https://cordis.europa.eu/project/id/506909> (Accessed Feb. 06, 2024).
- [33] M. Kumar, S. H. Kim, C. Lord, and S. Narayanan, “Improving speaker diarization for naturalistic child-adult conversational interactions using contextual information,” *The Journal of the Acoustical Society of America*, vol. 147, pp. EL196–EL200, 2 2020.
- [34] P. C. Loizou, *Speech Enhancement*, 2nd ed. CRC Press, 2 2013.
- [35] M. Huijbregts and F. de Jong, “Robust speech/non-speech classification in heterogeneous multimedia content,” *Speech Communication*, vol. 53, pp. 143–153, 2 2011.
- [36] G. Friedland, A. Janin, D. Imseng, X. A. Miro, L. Gottlieb, M. Huijbregts, M. T. Knox, and O. Vinyals, “The ICSI RT-09 Speaker Diarization System,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, pp. 371–381, 2 2012.
- [37] A. Tritschler and R. A. Gopinath, “Improved speaker segmentation and segments clustering using the bayesian information criterion,” in *Sixth European Conference*

on Speech Communication and Technology, 1999.

- [38] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, “Speaker Verification Using Adapted Gaussian Mixture Models,” *Digital Signal Processing*, vol. 10, pp. 19–41, 1 2000.
- [39] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, “Front-End Factor Analysis for Speaker Verification,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, pp. 788–798, 5 2011.
- [40] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, “X-Vectors: Robust DNN Embeddings for Speaker Recognition.” *IEEE*, 4 2018, pp. 5329–5333.
- [41] F. Landini, S. Wang, M. Diez, L. Burget, P. Matějka, K. Žmolíková, L. Mošner, A. Silnova, O. Plchot, O. Novotný, H. Zeinali, and J. Rohdin, “BUT System for the Second DIHARD Speech Diarization Challenge,” 2 2020.
- [42] G. Sell, D. Snyder, A. McCree, D. Garcia-Romero, J. Villalba, M. Maciejewski, V. Manohar, N. Dehak, D. Povey, S. Watanabe *et al.*, “Diarization is hard: Some experiences and lessons learned for the jhu team in the inaugural dihard challenge.” in *Interspeech*, 2018, pp. 2808–2812.
- [43] A. Arora, D. Raj, A. S. Subramanian, K. Li, B. Ben-Yair, M. Maciejewski, P. Želasko, P. García, S. Watanabe, and S. Khudanpur, “The JHU Multi-Microphone Multi-Speaker ASR System for the CHiME-6 Challenge,” 6 2020.
- [44] A. Ng, M. Jordan, and Y. Weiss, “On spectral clustering: analysis and an algorithm,” *Advances in neural information processing systems*, vol. 14, pp. 849–856, 2001.
- [45] P. Kenny, D. Reynolds, and F. Castaldo, “Diarization of Telephone Conversations Using Factor Analysis,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, pp. 1059–1070, 12 2010.
- [46] Y. Fujita, N. Kanda, S. Horiguchi, Y. Xue, K. Nagamatsu, and S. Watanabe, “End-to-End Neural Speaker Diarization with Self-attention,” 9 2019.
- [47] Y. Fujita, T. Komatsu, R. Scheibler, Y. Kida, and T. Ogawa, “Neural Diarization with Non-autoregressive Intermediate Attractors,” 3 2023.
- [48] S. Horiguchi, Y. Fujita, S. Watanabe, Y. Xue, and K. Nagamatsu, “End-to-End Speaker Diarization for an Unknown Number of Speakers with Encoder-Decoder Based Attractors,” 5 2020.
- [49] Y. Fujita, N. Kanda, S. Horiguchi, K. Nagamatsu, and S. Watanabe, “End-to-End Neural Speaker Diarization with Permutation-Free Objectives,” 9 2019.
- [50] G. Friedland, O. Vinyals, Y. Huang, and C. Muller, “Prosodic and other long-

- term features for speaker diarization,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, pp. 985–993, 7 2009.
- [51] D. Imseng and G. Friedland, “Robust speaker diarization for short speech recordings.” *IEEE*, 12 2009, pp. 432–437.
 - [52] D. Imseng and G. Friedland, “An adaptive initialization method for speaker diarization based on prosodic features.” *IEEE*, 2010, pp. 4946–4949.
 - [53] Z. K. Abdul and A. K. Al-Talabani, “Mel frequency cepstral coefficient and its applications: A review,” *IEEE Access*, vol. 10, pp. 122 136–122 158, 2022.
 - [54] “Extract MFCC, log energy, delta, and delta-delta of audio signal - MATLAB mfcc,” Available: <https://www.mathworks.com/help/audio/ref/mfcc.html> (Accessed Jun 24, 2023).
 - [55] P. Boersma, “Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound,” pp. 97–110, 1993.
 - [56] H. W. Kuhn, “The Hungarian method for the assignment problem,” *Naval Research Logistics Quarterly*, vol. 2, pp. 83–97, 3 1955.
 - [57] N. Ryant, K. Church, C. Cieri, A. Cristia, J. Du, S. Ganapathy, and M. Liberman, “The Second DIHARD Diarization Challenge: Dataset, task, and baselines,” 6 2019.
 - [58] M. Przybocki and A. Martin, “2000 NIST Speaker Recognition Evaluation LDC2001S97,” Available: <https://doi.org/10.35111/ex24-j205>, 2001.
 - [59] J. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, M. Kronenthal, G. Lathoud, M. Lincoln, A. Lisowska, I. McCowan, W. Post, D. Reidsma, and P. Wellner, “The ami meeting corpus: A pre-announcement,” S. Renals and S. Bengio, Eds. Springer Berlin Heidelberg, 2006, pp. 28–39.
 - [60] A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke, and C. Wooters, “The ICSI Meeting Corpus,” in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03).*, vol. 1, 2003, pp. I–I.
 - [61] “Obtaining the chime-5 dataset — chime-6 challenge,” Available: <https://chimechallenge.github.io/chime6/download.html> (Accessed Feb. 07, 2024).
 - [62] J. Barker, S. Watanabe, E. Vincent, and J. Trmal, “The fifth ‘CHiME’ Speech Separation and Recognition Challenge: Dataset, task and baselines,” 3 2018.
 - [63] S. Watanabe, M. Mandel, J. Barker, E. Vincent, A. Arora, X. Chang, S. Khudanpur, V. Manohar, D. Povey, D. Raj, D. Snyder, A. S. Subramanian, J. Trmal, B. B. Yair,

- C. Boeddeker, Z. Ni, Y. Fujita, S. Horiguchi, N. Kanda, T. Yoshioka, and N. Ryant, “CHiME-6 Challenge: Tackling Multispeaker Speech Recognition for Unsegmented Recordings,” 4 2020.
- [64] J. Huh, A. Brown, J. weon Jung, J. S. Chung, A. Nagrani, D. Garcia-Romero, and A. Zisserman, “Voxsrc 2022: The fourth voxceleb speaker recognition challenge,” 2 2023.
- [65] “VoxSRC 2023 challenge,” Available: <http://mm.kaist.ac.kr/datasets/voxceleb/voxsrc/competition2023.html> (Accessed Jun. 27, 2023).
- [66] Z. Chen, T. Yoshioka, L. Lu, T. Zhou, Z. Meng, Y. Luo, J. Wu, X. Xiao, and J. Li, “Continuous speech separation: dataset and analysis,” 1 2020.
- [67] “The Third DIHARD Speech Diarization Challenge,” Available: <https://dihardchallenge.github.io/dihard3/> (Accessed Jun. 27, 2023).
- [68] “Rich Transcription Evaluation — NIST,” Available: <https://www.nist.gov/itl/iad/mig/rich-transcription-evaluation> (Accessed Jun. 27, 2023).
- [69] O. Cetin and E. Shriberg, “Speaker Overlaps and ASR Errors in Meetings: Effects Before, During, and After the Overlap.” IEEE, 2006, pp. I–357–I–360.
- [70] T. Yoshioka, H. Erdogan, Z. Chen, X. Xiao, and F. Alleva, “Recognizing Overlapped Speech in Meetings: A Multichannel Separation Approach Using Neural Networks,” 10 2018.
- [71] N. Kanda, C. Boeddeker, J. Heitkaemper, Y. Fujita, S. Horiguchi, K. Nagamatsu, and R. Haeb-Umbach, “Guided Source Separation Meets a Strong ASR Backend: Hitachi/Paderborn University Joint Investigation for Dinner Party ASR,” 5 2019.
- [72] H. Bounazou, N. Asbai, and S. Zitouni, “Speaker Diarization in Overlapped Speech.” IEEE, 5 2022, pp. 890–895.
- [73] L.-A. R. Sacrey, S. Raza, V. Armstrong, J. A. Brian, A. Kushki, I. M. Smith, and L. Zwaigenbaum, “Physiological measurement of emotion from infancy to preschool: A systematic review and meta-analysis,” *Brain and Behavior*, vol. 11, pp. 1–48, 2 2021.
- [74] S. D. Calkins, “ORIGINS AND OUTCOMES OF INDIVIDUAL DIFFERENCES IN EMOTION REGULATION,” *Monographs of the Society for Research in Child Development*, vol. 59, pp. 53–72, 2 1994.
- [75] N. Eisenberg, R. A. Fabes, B. Murphy, P. Maszk, M. Smith, and M. Karbon, “The Role of Emotionality and Regulation in Children’s Social Functioning: A Longitudinal Study,” *Child Development*, vol. 66, p. 1360, 10 1995.

- [76] H. Selye, *The stress of life*. McGraw-Hill, 1956.
- [77] M. Campkin, “Stress management in primary care,” *Family Practice*, vol. 17, pp. 98–99, 2 2000.
- [78] S. Rotenberg and J. J. McGrath, “Inter-relation between autonomic and HPA axis activity in children and adolescents,” *Biological Psychology*, vol. 117, pp. 16–25, 5 2016.
- [79] A. H. Marques, M. N. Silverman, and E. M. Sternberg, “Evaluation of Stress Systems by Applying Noninvasive Methodologies: Measurements of Neuroimmune Biomarkers in the Sweat, Heart Rate Variability and Salivary Cortisol,” *Neuroimmunomodulation*, vol. 17, pp. 205–208, 2010.
- [80] S. W. Porges, “Cardiac vagal tone: A physiological index of stress,” *Neuroscience & Biobehavioral Reviews*, vol. 19, pp. 225–233, 6 1995.
- [81] N. Michels, I. Sioen, E. Clays, M. D. Buyzere, W. Ahrens, I. Huybrechts, B. Vanaelst, and S. D. Henauw, “Children’s heart rate variability as stress indicator: Association with reported stress and cortisol,” *Biological Psychology*, vol. 94, pp. 433–440, 10 2013.
- [82] “An Introduction to the Sympathetic and Parasympathetic Nervous System,” Available: <https://imotions.com/blog/nervous-system/> (Accessed Jan. 21, 2022).
- [83] S. W. Porges, “The polyvagal perspective,” *Biological Psychology*, vol. 74, pp. 116–143, 2 2007.
- [84] S. Porges, *The polyvagal theory: Neurophysiological foundations of emotions attachment communication self-regulation*. WW Norton & Company, 2011.
- [85] J. T. Cacioppo and G. G. Berntson, *The Brain, Homeostasis, and Health: Balancing Demands of the Internal and External Milieu*. Oxford University Press, 9 2012, pp. 121–137.
- [86] M. Malik, J. T. Bigger, A. J. Camm, R. E. Kleiger, A. Malliani, A. J. Moss, and P. J. Schwartz, “Heart rate variability: Standards of measurement, physiological interpretation, and clinical use,” *European Heart Journal*, vol. 17, pp. 354–381, 3 1996.
- [87] J. Lacey, J. Kagan, B. Lacey, and H. Moss, “Situational determinants and behavioral correlates of autonomic response pattern,” *Expression of emotions in man*. International University Press, pp. 161–196, 1963.
- [88] U. R. Acharya, K. P. Joseph, N. Kannathal, C. M. Lim, and J. S. Suri, “Heart rate variability: a review,” *Medical & Biological Engineering & Computing*, vol. 44, pp. 1031–1051, 12 2006.

- [89] R. M. Sapolsky, L. M. Romero, and A. U. Munck, “How do glucocorticoids influence stress responses? integrating permissive, suppressive, stimulatory, and preparative actions*,” *Endocrine Reviews*, vol. 21, pp. 55–89, 2 2000.
- [90] G. G. BERNTSON, J. T. BIGGER, D. L. ECKBERG, P. GROSSMAN, P. G. KAUFMANN, M. MALIK, H. N. NAGARAJA, S. W. PORGES, J. P. SAUL, P. H. STONE, and M. W. V. D. MOLEN, “Heart rate variability: Origins, methods, and interpretive caveats,” *Psychophysiology*, vol. 34, 11 1997.
- [91] R. Mccraty and F. Shaffer, “Heart Rate Variability: New Perspectives on Physiological Mechanisms, Assessment of Self-regulatory Capacity, and Health Risk,” *Global Advances in Health and Medicine*, vol. 4, pp. 46–61, 1 2015.
- [92] F. Shaffer and J. P. Ginsberg, “An Overview of Heart Rate Variability Metrics and Norms,” *Frontiers in Public Health*, vol. 5, 9 2017.
- [93] “Trends in heart-rate variability signal analysis,” *Frontiers in Digital Health*, vol. 3, 2 2021.
- [94] E. Burns, “Pediatric ECG Interpretation,” Available: <https://litfl.com/normal-paediatric-ecg/>, 2 2021.
- [95] S. Laborde, E. Mosley, and J. F. Thayer, “Heart Rate Variability and Cardiac Vagal Tone in Psychophysiological Research – Recommendations for Experiment Planning, Data Analysis, and Data Reporting,” *Frontiers in Psychology*, vol. 08, 2 2017.
- [96] J. F. THAYER and E. STERNBERG, “Beyond Heart Rate Variability: Vagal Regulation of Allostatic Systems,” *Annals of the New York Academy of Sciences*, vol. 1088, pp. 361–372, 11 2006.
- [97] A. J. Masino, D. Forsyth, H. Nuske, J. Herrington, J. Pennington, Y. Kushleyeva, and C. P. Bonafide, “M-Health and autism: Recognizing stress and anxiety with machine learning and wearables data,” vol. 2019-June. Institute of Electrical and Electronics Engineers Inc., 6 2019, pp. 714–719.
- [98] M. P. Tarvainen, J.-P. Niskanen, J. A. Lipponen, P. O. Ranta-aho, and P. A. Karjalainen, “Kubios hrv – heart rate variability analysis software,” *Computer Methods and Programs in Biomedicine*, vol. 113, pp. 210–220, 1 2014.
- [99] R. E. Kleiger, P. K. Stein, and J. T. Bigger, “Heart Rate Variability: Measurement and Clinical Utility,” *Annals of Noninvasive Electrocardiology*, vol. 10, pp. 88–101, 1 2005.
- [100] L. K. Hill and A. Siebenbrock, “Are all measures created equal? Heart rate variability and respiration - biomed 2009,” *Biomedical sciences instrumentation*, vol. 45, pp. 71–6, 2009.

- [101] J. M. Karemaker, "Counterpoint: Respiratory sinus arrhythmia is due to the baroreflex mechanism," *Journal of Applied Physiology*, vol. 106, pp. 1742–1743, 5 2009.
- [102] J. W. Denver, S. F. Reed, and S. W. Porges, "Methodological issues in the quantification of respiratory sinus arrhythmia." *Biological psychology*, vol. 74, pp. 286–94, 2 2007.
- [103] P. Grossman and E. W. Taylor, "Toward understanding respiratory sinus arrhythmia: Relations to cardiac vagal tone, evolution and biobehavioral functions," *Biological Psychology*, vol. 74, pp. 263–285, 2 2007.
- [104] D. L. Eckberg, "Human sinus arrhythmia as an index of vagal cardiac outflow," *Journal of Applied Physiology*, vol. 54, pp. 961–966, 4 1983.
- [105] J. P. Fisher, T. Zera, and J. F. Paton, *Respiratory–cardiovascular interactions*, 2022, pp. 279–308.
- [106] D. S. Quintana, G. A. Alvares, and J. A. J. Heathers, "Guidelines for Reporting Articles on Psychiatry and Heart rate variability (GRAPH): recommendations to advance research communication," *Translational Psychiatry*, vol. 6, pp. e803–e803, 5 2016.
- [107] A. Boardman, F. S. Schlindwein, A. P. Rocha, and A. Leite, "A study on the optimum order of autoregressive models for heart rate variability," *Physiological Measurement*, vol. 23, pp. 325–336, 5 2002.
- [108] J. A. Hirsch and B. Bishop, "Respiratory sinus arrhythmia in humans: how breathing pattern modulates heart rate," *American Journal of Physiology-Heart and Circulatory Physiology*, vol. 241, pp. H620–H629, 10 1981.
- [109] L. Lambrechts and B. Fourie, "How to interpret an electrocardiogram in children," *BJA Education*, vol. 20, pp. 266–277, 8 2020.
- [110] M. A. Almeida-Santos, J. A. Barreto-Filho, J. L. M. Oliveira, F. P. Reis, C. C. da Cunha Oliveira, and A. C. S. Sousa, "Aging, heart rate variability and patterns of autonomic regulation of the heart," *Archives of Gerontology and Geriatrics*, vol. 63, pp. 1–8, 3 2016.
- [111] J. Koenig and J. F. Thayer, "Sex differences in healthy human heart rate variability: A meta-analysis," *Neuroscience & Biobehavioral Reviews*, vol. 64, pp. 288–310, 5 2016.
- [112] "Polar H10 Heart Rate Sensor System," 2019.
- [113] "Watch - apple (za)," Available: <https://www.apple.com/za/watch/> (Accessed Aug. 10, 2023).

- [114] R. Gilgen-Ammann, T. Schweizer, and T. Wyss, “RR interval signal quality of a heart rate monitor and an ECG Holter at rest and during exercise.” *European journal of applied physiology*, vol. 119, 7 2019.
- [115] K. E. Speer, S. Semple, N. Naumovski, and A. J. McKune, “Measuring Heart Rate Variability Using Commercially Available Devices in Healthy Children: A Validity and Reliability Study,” *European Journal of Investigation in Health, Psychology and Education*, vol. 10, 1 2020.
- [116] *ECG interpretation made incredibly easy*. Wolters Kluwer—Lippincott Williams & Wilkins, 2011.
- [117] M. Ghamari, “A review on wearable photoplethysmography sensors and their potential future applications in health care,” *International Journal of Biosensors & Bioelectronics*, vol. 4, 2018.
- [118] M. Zhao, F. Adib, and D. Katabi, “Emotion recognition using wireless signals.” *ACM*, 10 2016, pp. 95–108.
- [119] P. C. Petrantonakis and L. J. Hadjileontiadis, *EEG-Based Emotion Recognition Using Advanced Signal Processing Techniques*. Wiley, 2 2015, pp. 269–293.
- [120] Y. G. Yang and S. Yang, “Study of Emotion Recognition Based on Surface Electromyography and Improved Least Squares Support Vector Machine,” *Journal of Computers*, vol. 6, 8 2011.
- [121] T. Sharma, S. Bhardwaj, and H. B. Maringanti, “Emotion estimation using physiological signals.” *IEEE*, 11 2008, pp. 1–5.
- [122] F. Hönig, J. Wagner, A. Batliner, and E. Nöth, “Classification of user states with physiological signals: On-line generic features vs. specialized feature sets,” 2009, pp. 2357–2361.
- [123] G. Friesen, T. Jannett, M. Jadallah, S. Yates, S. Quint, and H. Nagle, “A comparison of the noise sensitivity of nine QRS detection algorithms,” *IEEE Transactions on Biomedical Engineering*, vol. 37, pp. 85–98, 1990.
- [124] P. M. Tripathi, A. Kumar, R. Komaragiri, and M. Kumar, “A Review on Computational Methods for Denoising and Detecting ECG Signals to Detect Cardiovascular Diseases,” *Archives of Computational Methods in Engineering*, 10 2021.
- [125] U. Satija, B. Ramkumar, and M. S. Manikandan, “A Review of Signal Processing Techniques for Electrocardiogram Signal Quality Assessment,” *IEEE Reviews in Biomedical Engineering*, vol. 11, pp. 36–52, 2018.
- [126] M. Tarvainen, P. Ranta-aho, and P. Karjalainen, “An advanced detrending method with application to HRV analysis,” *IEEE Transactions on Biomedical Engineering*,

vol. 49, pp. 172–175, 2002.

- [127] B. S. Shaik, G. Naganjaneyulu, T. Chandrasheker, and A. Narasimhadhan, “A method for qrs delineation based on stft using adaptive threshold,” *Procedia Computer Science*, vol. 54, pp. 646–653, 2015.
- [128] P. Sabherwal, L. Singh, and M. Agrawal, “Aiding the Detection of QRS Complex in ECG Signals by Detecting S Peaks Independently,” *Cardiovascular Engineering and Technology*, vol. 9, pp. 469–481, 9 2018.
- [129] M. Merah, T. Abdelmalik, and B. Larbi, “R-peaks detection based on stationary wavelet transform,” *Computer Methods and Programs in Biomedicine*, vol. 121, pp. 149–160, 10 2015.
- [130] “Discrete stationary wavelet transform 1-D - MATLAB swt,” Available: <https://www.mathworks.com/help/wavelet/ref/swt.html> (Accessed Jun. 09, 2022).
- [131] J.-C. Huang, C.-F. Chen, C.-C. Chang, S.-C. Chen, M.-C. Hsieh, Y.-P. Hsieh, and H.-C. Chen, “Effects of stroke on changes in heart rate variability during hemodialysis,” *BMC Nephrology*, vol. 18, p. 90, 12 2017.
- [132] R. Castaldo, L. Montesinos, P. Melillo, C. James, and L. Pecchia, “Ultra-short term hrv features as surrogates of short term hrv: a case study on mental stress detection in real life,” *BMC Medical Informatics and Decision Making*, vol. 19, p. 12, 12 2019.
- [133] H.-M. Cho, H. Park, S.-Y. Dong, and I. Youn, “Ambulatory and laboratory stress detection based on raw electrocardiogram signals using a convolutional neural network,” *Sensors*, vol. 19, p. 4408, 10 2019.
- [134] V. S. Kublanov, A. Y. Dolganov, D. Belo, and H. Gamboa, “Comparison of machine learning methods for the arterial hypertension diagnostics,” *Applied Bionics and Biomechanics*, vol. 2017, pp. 1–13, 2017.
- [135] F. Ma, J. Zhang, W. Liang, and J. Xue, “Automated classification of atrial fibrillation using artificial neural network for wearable devices,” *Mathematical Problems in Engineering*, vol. 2020, pp. 1–6, 4 2020.
- [136] A. J. Khalaf and S. J. Mohammed, “Verification and comparison of mit-bih arrhythmia database based on number of beats,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, p. 4950, 12 2021.
- [137] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, “Physiobank, physiotoolkit, and physionet,” *Circulation*, vol. 101, 6 2000.
- [138] “American heart association ecg database usb,” Available: <https://www.ecri.org/american-heart-association-ecg-database-usb> (Accessed Sep. 08, 2023).

- [139] P. Sabherwal, M. Agrawal, and L. Singh, “Automatic Detection of the R Peaks in Single-Lead ECG Signal,” *Circuits, Systems, and Signal Processing*, vol. 36, pp. 4637–4652, 11 2017.
- [140] M. Soleymani, J. Lichtenauer, T. Pun, and M. Pantic, “A multimodal database for affect recognition and implicit tagging,” *IEEE Transactions on Affective Computing*, vol. 3, pp. 42–55, 1 2012.
- [141] J. Healey and R. Picard, “Detecting stress during real-world driving tasks using physiological sensors,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, pp. 156–166, 6 2005.
- [142] A. Site, J. Nurmi, and E. S. Lohan, “Systematic Review on Machine-Learning Algorithms Used in Wearable-Based eHealth Data Analysis,” *IEEE Access*, vol. 9, pp. 112 221–112 235, 2021.
- [143] “Science - elite hrv,” Available: <https://elitehrv.com/science> (Accessed Jul. 10, 2023).
- [144] “Github - polarofficial/polar-ble-sdk,” Available: <https://github.com/polarofficial/polar-ble-sdk> (Accessed Jul. 09, 2023).
- [145] A. S. Perrotta, A. T. Jeklin, B. A. Hives, L. E. Meanwell, and D. E. Warburton, “Validity of the elite hrv smartphone application for examining heart rate variability in a field-based setting,” *Journal of Strength and Conditioning Research*, vol. 31, pp. 2296–2302, 8 2017.
- [146] D. Anguita, A. Ghio, L. Oneto, X. Parra, J. L. Reyes-Ortiz *et al.*, “A public domain dataset for human activity recognition using smartphones.” vol. 3, 2013, p. 3.
- [147] J.-L. Reyes-Ortiz, L. Oneto, A. Samà, X. Parra, and D. Anguita, “Transition-Aware Human Activity Recognition Using Smartphones,” *Neurocomputing*, vol. 171, pp. 754–767, 1 2016.
- [148] S. Narayanan and P. G. Georgiou, “Behavioral Signal Processing: Deriving Human Behavioral Informatics From Speech and Language,” *Proceedings of the IEEE*, vol. 101, pp. 1203–1233, 5 2013.
- [149] Y. Fujiki, “iphone as a physical activity measurement platform.” ACM, 4 2010, pp. 4315–4320.
- [150] D. Karantonis, M. Narayanan, M. Mathie, N. Lovell, and B. Celler, “Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring,” *IEEE Transactions on Information Technology in Biomedicine*, vol. 10, pp. 156–167, 1 2006.
- [151] P. Maziewski, A. Kupryjanow, K. Kaszuba, and A. Czyżewski, “Accelerometer signal

- pre-processing influence on human activity recognition,” 2009, pp. 95–99.
- [152] J. Pan and W. J. Tompkins, “A real-time qrs detection algorithm,” *IEEE transactions on biomedical engineering*, no. 3, pp. 230–236, 1985.
 - [153] “Heart rate service — bluetooth® technology website.”
 - [154] “Voicebox,” Available: <http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html> (Accessed Sep. 26, 2022).
 - [155] R. Snell and F. Milinazzo, “Formant location from lpc analysis data,” *IEEE Transactions on Speech and Audio Processing*, vol. 1, pp. 129–134, 4 1993.
 - [156] T. Giannakopoulos, “A method for silence removal and segmentation of speech signals, implemented in matlab,” *University of Athens, Athens*, vol. 2, 2009.
 - [157] “k-means clustering - matlab kmeans,” Available: <https://www.mathworks.com/help/stats/kmeans.html#bueftl4-1> (Accessed Oct. 19, 2023).
 - [158] J.-L. Reyes-Ortiz, L. Oneto, A. Ghio, A. Samá, D. Anguita, and X. Parra, “LNCS 8681 - Human Activity Recognition on Smartphones with Awareness of Basic Activities and Postural Transitions,” 2014.
 - [159] K. R. Evenson, D. J. Catellier, K. Gill, K. S. Ondrak, and R. G. McMurray, “Calibration of two objective measures of physical activity for children.” *Journal of sports sciences*, vol. 26, 12 2008.
 - [160] H. Henriksson, E. Forsum, and M. Löf, “Evaluation of Actiheart and a 7 d activity diary for estimating free-living total and activity energy expenditure using criterion methods in 1.5- and 3-year-old children,” *British Journal of Nutrition*, vol. 111, pp. 1830–1840, 5 2014.
 - [161] M. A. McNarry and K. A. Mackintosh, “Reproducibility of Heart Rate Variability Indices in Children with Cystic Fibrosis,” *PLOS ONE*, vol. 11, p. e0151464, 3 2016.