RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

ÉCOLE NATIONALE POLYTECHNIQUE

المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique

Département d'Électronique

End of Studies Project Report

For the obtaining of State Engineer degree in Electronics

Multimodal Neurophysiological Signals Analysis for Stress Assessment

---

Israa BOULEFAAT & Mohamed Abdelhadi CHERFOUHI

Under the Supervision of :

Pr. Taous Meriem LALEG & Pr. Latifa HAMAMI

Presented and publicly defended on : June 21st, 2025

Jury:

President:	Pr. Rachida Touhami	ENP, Algiers
Examiner:	Pr.Hicham Bousbia-Salah	ENP, Algiers
Supervisor:	Pr. Taous Meriem LALEG	INRIA-Saclay, Paris
Supervisor:	Pr. Latifa HAMAMI	ENP, Algiers

ENP 2025

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

ÉCOLE NATIONALE POLYTECHNIQUE

المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique

Département d'Électronique

End of Studies Project Report

For the obtaining of State Engineer degree in Electronics

Multimodal Neurophysiological Signals Analysis for Stress Assessment

---

Israa BOULEFAAT & Mohamed Abdelhadi CHERFOUHI

Under the Supervision of :

Pr. Taous Meriem LALEG & Pr. Latifa HAMAMI

Presented and publicly defended on : June 21st, 2025

Jury:

President:	Pr. Rachida Touhami	ENP, Algiers
Examiner:	Pr. Hicham Bousbia-Salah	ENP, Algiers
Supervisor:	Pr. Taous Meriem LALEG	INRIA-Saclay, Paris
Supervisor :	Pr. Latifa HAMAMI	ENP, Algiers

ENP 2025

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

ÉCOLE NATIONALE POLYTECHNIQUE

المدرسة الوطنية المتعددة التقنيات
Ecole Nationale Polytechnique

Département d'Électronique

Mémoire de Projet de Fin d'Études

En vue de l'obtention du diplôme d'Ingénieur d'État en Électronique

---

Analyse Multimodale des Signaux Neurophysiologiques pour l'Évaluation du Stress

---

Israa BOULEFAAT & Mohamed Abdelhadi CHERFOUHI

Encadré par :

Pr. Taous Meriem LALEG & Pr. Latifa HAMAMI

Présenté et soutenu publiquement le: 21 Juin 2025

Composition du jury :

Présidente:	Pr. Rachida Touhami	ENP, Alger
Examinateur:	Pr. Hicham Bousbia-Salah	ENP, Alger
Encadrante:	Pr. Taous Meriem LALEG	INRIA-Saclay, Paris
Encadrante:	Pr. Latifa HAMAMI	ENP, Alger

ENP 2025

ملخص

هذا المشروع يدرس استخدام الإشارات الفسيولوجية العصبية مثل تخطيط الدماغ EEG، وتخطيط القلب ECG، وتخطيط ضوئي للدم PPG لتقييم مستوى التوتر لدى 23 مشاركا أثناء تنفيذهم لمهام مختلفة، حيث يتم تسجيل مستوى التوتر لكل مهمة. تُجزأ هذه الإشارات وتُفلتر، ثم تحول إلى صور باستخدام طريقتين مختلفتين: تمثيل الصور عبر الرسم البياني للرؤية (Visibility Graph) وتمثيل الصور بحقول الزوايا الغرامية (Gramian Angular Field).

تسمح هذه المقاربة متعددة الأنماط (Multimodal) باستخلاص معلومات متكاملة من إشارات فسيولوجية مختلفة. تُستخرج الخصائص بعد ذلك من الصور باستخدام طريقتين متكاملتين: تحويل حزم المويجات (wavelet packet transform) مع لحظات زيرنيك(Zernike's moments) وهو (Hu's moments)، وتحليل الإشارة شبه الكلاسيكي (SCSA).

بعد الانتهاء من جميع مراحل المعالجة، يتم تدريب نموذج تعلُّم آلي خاضع للإشراف باستخدام تسميات التوتر، بهدف تقييم ومقارنة فعالية طرق استخراج الخصائص المختلفة. بعد ذلك، يتم اختيار الطريقة الأكثر كفاءة لكل نوع من الإشارات، ثم تُدمج النتائج من أجل تحسين أداء نظام تقييم التوتر بالاعتماد على مبدأ التعددية في مصادر الإشارة (Multimodality).

الكلمات المفتاحية: التوتر — الإشارات العصبية الفسيولوجية — تحويل الصور — استخراج الخصائص — الشبكات العصبية.

Résumé

Ce projet étudie l'utilisation des signaux neurophysiologiques, notamment l'**EEG**, l'**ECG** et le **PPG**, pour l'évaluation du stress chez **23 participants** effectuant différentes tâches, chacune étant associée à un niveau de stress spécifique. Les signaux sont segmentés, filtrés, puis convertis en images selon deux méthodes distinctes : **la représentation par graphe de visibilité** et **la représentation par champ angulaire de Gramian**. Ce traitement multimodal permet de capturer des informations complémentaires provenant de différentes modalités physiologiques.

L'extraction de caractéristiques repose sur deux approches complémentaires : **la transformée en paquets d'ondelettes**, combinée aux **moments de Zernike et de Hu**, et **l'analyse semi-classique des signaux (SCSA)**. Une fois les signaux transformés et les caractéristiques extraites, un modèle d'apprentissage supervisé est entraîné à l'aide des étiquettes de stress afin d'évaluer et de comparer l'efficacité des différentes méthodes d'extraction.

Pour chaque type de signal, l'approche la plus performante est sélectionnée. Ces meilleures configurations sont ensuite combinées afin d'**améliorer les performances du système d'évaluation du stress**, en tirant parti de la complémentarité offerte par la **multimodalité**.

Mots-clés : Stress — Signaux neurophysiologiques — Multimodalité — Apprentissage automatique — Traitement d'image — Classification.

Abstract

This project investigates the use of neurophysiological signals, specifically EEG, ECG, and PPG, for stress assessment in 23 participants performing various tasks, with corresponding stress levels recorded for each activity. The signals are segmented and filtered, then converted into images using two distinct techniques: **Visibility Graph** and **Gramian Angular Field Image Representations**.

This multimodal approach enables the integration of diverse and complementary information from different physiological sources. Feature extraction is subsequently performed using two complementary strategies: **Wavelet Packet Transform combined with Zernike and Hu Moments**, and **Semi-Classical Signal Analysis (SCSA)**.

Once the full processing pipeline is completed, a supervised machine learning model is trained using the stress labels in order to evaluate and compare the performance of each feature extraction method. For each signal type, the most effective strategy is selected, and their outputs are then fused to **enhance the overall performance of the stress assessment system** by leveraging the benefits of **multimodality**.

Keywords: Stress — Neurophysiological Signals — Multimodality — Machine Learning — Image Processing — Classification.

Acknowledgments

We would like to express our deepest gratitude to all those who contributed to the successful completion of our graduation project.

First and foremost, we extend our sincere thanks to Professor LALEG, our supervisor, for her unwavering support, invaluable guidance, and insightful advice throughout the entire duration of this work.

We also express our heartfelt appreciation to Professor HAMAMI, our co-supervisor, for her availability, constructive feedback, and continuous encouragement at every stage of our journey.

A special thanks goes to Professor DJEDDOU, his guidance was crucial in the signal filtering phase of our project, providing us with both theoretical and practical tools that enriched our understanding.

We are equally grateful to our colleague and friend Juan, for his constant help and technical support. His experience and dedication greatly contributed to overcoming the many challenges we faced during our project.

Finally, we thank all the professors who accompanied us throughout our academic path for the knowledge and inspiration they shared, shaping the engineers we are becoming today.

Israa & Abdelhadi

Dedication

To Engineer Israa Boulefaat, For your strong courage, steady perseverance, and quiet victories through these 18 years. ***Engineer against all odds.***

To my mother, whose boundless love and whispered prayers have been the soul of my journey. Your faith in me lit the way even in my darkest hours.

To my dear friend Meriem, for your unwavering presence, your support in every moment of doubt, and the warmth that never failed me, you are deeply treasured.

To my second mother, whose love gave me the courage to keep going. Your kindness, wisdom, and strength have quietly shaped who I am. You are a blessing beyond words.

And to my teammate Abdelhadi for his loyalty and valuable help at every step of this journey.

Thank you all, from the deepest place in my heart.

Israa BOULEFAAT

Dedication

To My Parents, to my mother, my first teacher, my endless source of love and sacrifice, you have given me everything without asking for anything in return. To my father, my pillar of strength, my encourager, your wisdom and support have shaped who I am. Without you both, I would be lost. May Allah bless you with endless happiness and health.

To My Teacher, Hamdi, you are more than a teacher, you are a mother, a mentor, and a guiding light. Your patience, wisdom, and kindness have inspired me in ways words cannot express. May Allah protect you, reward you abundantly, and keep you with us for many more years to come.

To My Best Brothers, Khaled, Chegrani and bahaa-eddine, your unwavering support, advice, and belief in me mean more than I can say. I am endlessly grateful to have you in my life. Thank you for everything.

To my sister, Maroua, thanks for being here every time with me.

To My Partner, My Teammate, Israa, you are a blessing I will never take for granted. Your kindness, dedication, and unwavering support, in both good times and bad, make you truly irreplaceable. You are the kind of person everyone dreams of working with, and I am endlessly grateful to have you by my side. May Allah reward you with all the goodness you bring to others.

Abdelhadi Mohamed CHERFOUHI

Contents

List of Figures

List of Tables

Acronyms list

- **ANS** : Autonomic Nervous System
- **ECG** : electrocardiography
- **EEG** : electroencephalography
- **ET** : Extra Trees
- **GAF** : Gramian angular field
- **HRV** : Heart Rate Variability
- **LGBM** : Light Gradient-Boosting Machine
- **NN** : Neural Network
- **P1** : Feature concatenation of Zernike moments, Hu moments, and Wavelet Packet Transform coefficients
- **PASS** : Physical Activity and StresS
- **PPG** : photoplethysmography
- **QDA** : Quadratic Discriminant Analysis
- **RF** : Random Forest
- **SCSA** : Semi-classical signal Analysis
- **SVM** : Support Vector Machine
- **VG** : Visibility graph
- **WPT** : Wavelet Packet Transform

General Introduction

"**Integrating multiple modalities provides a more comprehensive understanding of stress, given that stress manifests differently across different people.**"

Hosseini et al., Multimodal Stress Detection Using Facial Landmarks and Biometric Signals, 2023

Stress is a pervasive condition that significantly impacts both mental and physical health, contributing to cognitive decline, cardiovascular diseases, and reduced quality of life [45]. Traditional stress assessment methods, such as self-reported questionnaires, are subjective and often fail to capture real-time physiological responses. Recent advances in wearable technology and signal processing have enabled the use of neurophysiological signals, such as electroencephalography (EEG), electrocardiography (ECG), and photoplethysmography (PPG), to objectively quantify stress levels. However, relying on a single modality can lead to incomplete or biased assessments due to the complex, multisystem nature of stress responses.

This thesis addresses the critical need for a robust, multimodal approach to stress detection by integrating EEG, ECG, and PPG signals. Our work leverages modern signal processing techniques, machine learning, and data fusion strategies to improve classification accuracy and enable real-time stress monitoring. The proposed methodology not only advances the state-of-the-art in stress assessment but also lays the foundation for practical applications in healthcare, workplace wellness, and human-computer interaction.

Despite extensive research on stress detection, existing systems face several limitations:

- **Unimodal Bias:** Most studies focus on a single signal type (e.g., EEG or ECG), ignoring the complementary information provided by multimodal data.
- **Class Imbalance:** Real-world stress datasets are often skewed, leading to models that perform poorly on underrepresented classes.
- **Real-Time Constraints:** Many methods rely on computationally expensive features, making them unsuitable for embedded or wearable devices.

Our work bridges these gaps by:

- Developing a multimodal fusion framework that combines EEG, ECG, and PPG signals for higher accuracy.
- Introducing data augmentation and balancing techniques to improve model generalizability.
- Optimizing feature extraction and classification pipelines for future real-time deployment on hardware such as FPGAs.

Project Objectives

The primary objective of this work is to develop a high-accuracy multimodal stress classification system that integrates EEG, ECG, and PPG signals. With:

- Achieving robust performance in both binary (stress/non-stress) and three-class (low/medium/high stress) scenarios.
- Addressing dataset imbalance through downsampling and SMOTE-based augmentation.
- Optimizing the pipeline for future deployment on FPGA hardware to enable real-time and low-latency stress monitoring.

By combining advanced signal processing with multimodal machine learning, this research contributes to the development of wearable, real-time stress detection systems with applications in healthcare, occupational safety, and mental wellness.

Contributions

This thesis makes the following contributions:

Novel Signal Processing Pipeline

- Advanced filtering techniques (methods from neurokit) to improve signal quality.
- Transformation of 1D signals into 2D representations (Visibility Graphs and Gramian Angular Fields) to capture temporal and structural patterns.

Feature Engineering

- Hybrid feature extraction using spectral methods (SCSA) and image descriptors (Zernike moments, Hu moments, wavelet packets transform coefficients).
- Dimensionality reduction and significant feature selection via correlation analysis and ANOVA test.

Multimodal Fusion

- Integration of best feature's paths from EEG, ECG, and PPG data to enhance stress classification performance.
- Comparative analysis of unimodal vs. multimodal approaches, demonstrating the superiority of fusion.

Data Balancing and Augmentation

- Application of SMOTE to address class imbalance, significantly improving recall for minority classes.
- Validation of augmentation techniques in both binary and three-class stress classification.

Machine Learning Optimization

- Evaluation of multiple classifiers (Random Forest, XGBoost, Neural Networks) for stress detection.
- Permutation tests to confirm the statistical significance of results.

Methodology Overview

Our methodology is structured into four stages:

1. **Data Acquisition and Preprocessing:** Using the PASS dataset [41], we segment and filter EEG, ECG, and PPG signals to remove noise and artifacts.
2. **Signal-to-Image Representation:** Convert time-series signals into 2D images (VG and GAF) to leverage spatial and temporal correlations.
3. **Feature Extraction and Fusion:** Extract discriminative features from each modality and fuse them for multimodal classification.
4. **Model Training and Validation:** Train machine learning models on both balanced and imbalanced datasets and evaluate performance using accuracy, precision, recall, and F1-score.

Thesis Structure

This dissertation is organized as follows:

- **Chapter 1:** Reviews the state-of-the-art in stress assessment, covering definitions, physiological impacts, and existing multimodal techniques.
- **Chapter 2:** Details the acquisition and processing of neurophysiological signals, including preprocessing, feature extraction, and machine learning methods.
- **Chapter 3:** Presents the implementation and experimentation, including dataset preparation, classification pipelines, and results.

  Finally a conclusion with essential findings, limitations, and future directions for real-time stress monitoring systems.

Chapter 1

State-of-the-Art on Stress Assessment

1.1 Introduction

In the first chapter, we will explore stress, its impact on health, and the classical methods used to assess it. After that, we will examine the contribution of neurophysiological signals to stress evaluation, review several multimodal databases, and conclude with a discussion on the concept of multimodality.

1.2 Definition of Stress

According to Papathanasiou et al.[40], stress is a universal phenomenon that can be defined in multiple ways depending on the theoretical approach. It can be understood as a reaction, a stimulus, or a transaction between an individual and their environment.

1.2.1 Stress as a Reaction

For this case, stress is a biological and psychological response of the body to an internal or external demand. This reaction involves specific physiological changes that can affect an individual's well-being. The human body activates defense mechanisms when facing stressful events. This activation manifests through hormonal changes, increased heart rate, muscle tension, and other physiological responses [48]. Stress is therefore seen as an adaptive reaction that helps the body cope with potentially threatening situations.

1.2.2 Stress as a Stimulus

In this approach, stress is defined as a series of events or environmental changes that trigger a reaction in the individual. These events, known as stressors, can be acute (sudden and intense events such as accidents or bad news), chronic (long-term stressors such as work pressure or family conflicts), recurrent (periodic stressful events such as exams or deadlines), or cumulative (the buildup of small stressors leading to mental and physical exhaustion) [22]. In this framework, stress is considered an external force that disrupts an individual's psychological and physiological balance

1.2.3 Stress as a Transaction

This model, developed by Lazarus and Folkman [14], views stress as an interaction between an individual and their environment. It emphasizes the subjective perception of stress rather than stress itself. The same event may be stressful for one person but insignificant for another. How a person evaluates an event and their available resources to cope with it determines whether they experience stress. The evaluation process involves three stages: primary appraisal, where the individual determines whether the event is a threat, a challenge, or irrelevant; secondary appraisal, where they assess their resources to handle the situation; and third appraisal, where they adjust their perception based on new information or experiences.

1.3 Impact of Stress on Health and Cognition

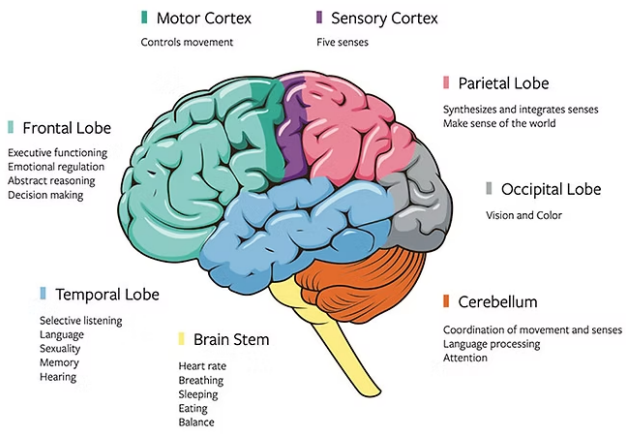
Stress aggravates health issues, contributing to cognitive decline and significantly increasing the risk of Alzheimer's disease and dementia.[45].

It's not uncommon to feel disorganized and forgetful when under significant stress. However, long-term stress may alter brain function in ways that impair memory. Studies in both animals and humans demonstrate that stress can affect cognitive processing. According to Dr. Kerry Ressler, chief scientific officer at McLean Hospital and professor of psychiatry at Harvard Medical School, stress—whether real-life or experimentally induced—impairs cognition, attention, and memory.

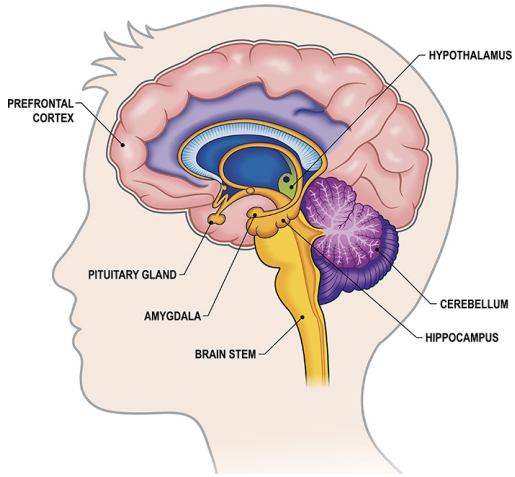
Beyond its effects on memory, stress influences various brain functions, including mood and anxiety regulation. It also promotes inflammation, negatively impacting cardiovascular health. Additionally, research suggests that stress may affect men and women differently [45].

To understand why stress affects cognition, it is essential to consider brain function (Fig 1.1a). The brain consists of multiple interconnected regions (Fig 1.1b) responsible for different tasks. Under stress, the amygdala—responsible for survival instincts—becomes highly active, diverting resources away from memory-related areas such as the hippocampus. This shift explains why stress can cause forgetfulness and memory lapses, particularly in high-pressure or traumatic situations.

These findings highlight the critical role of stress management in preserving cognitive function and overall brain health.



(a) Brain functions [12]



(b) Brain regions [26]

Figure 1.1: Interconnected regions responsible for different tasks

1.4 Classical Stress Measurement Methods

Stress measurement methods can be broadly categorized into subjective questionnaires and objective biomarkers.

1.4.1 Questionnaires

Questionnaires are widely used to assess stress levels based on self-reported data. These methods rely on individuals' perceptions of their stress levels, which can be influenced by personal

biases and external factors. Despite their limitations, they remain a simple and accessible means of stress evaluation.

NASA Task Load Index (NASA-TLX)
The NASA-TLX is a subjective workload assessment tool commonly used in psychological and human performance studies. It evaluates six subscales: mental demand, physical demand, temporal demand, performance, effort, and frustration [41].

BORG Rating of Perceived Exertion (BORG Scale)
The BORG scale measures perceived exertion, which is often correlated with physiological stress responses. Participants rate their exertion levels on a scale from 6 (no exertion) to 20 (maximum exertion). This method is particularly useful in physical and occupational stress research [41].

State-Trait Anxiety Inventory (STAI)
The STAI assesses both temporary (state) and long-term (trait) anxiety levels. Stress is strongly linked to anxiety, making this tool valuable for stress research. It consists of two 20-item subscales that measure how individuals feel at a particular moment versus their general tendency toward anxiety [5].

Perceived Stress Scale (PSS)
The PSS is a widely used psychological instrument for measuring the perception of stress. It assesses how unpredictable, uncontrollable, and overloaded respondents find their lives. This scale helps identify chronic stress levels in everyday life [44].

1.4.2 Biomarkers

Biomarkers provide objective measures of stress by analyzing physiological responses. These methods help overcome the subjectivity of self-reported questionnaires and offer insight into the biological impact of stress.

Salivary Cortisol
Cortisol, the primary stress hormone, is released by the adrenal glands in response to stress. Measuring salivary cortisol levels provides a non-invasive way to assess stress responses over time [34].

Salivary Alpha-Amylase (SAA)
Alpha-amylase is an enzyme that increases in response to stress, particularly in acute stress situations. Salivary alpha-amylase levels correlate with the activation of the sympathetic nervous system, making it a reliable biomarker for stress assessment [34].

Heart Rate Variability (HRV)
HRV refers to the variation in time intervals between consecutive heartbeats (Fig 1.2). A lower HRV is associated with higher stress levels and reduced autonomic nervous system flexibility. Stress-related changes in HRV are commonly assessed using electrocardiography (ECG) [41].

Electrodermal Activity (EDA)
EDA, also known as galvanic skin response (GSR), measures sweat gland activity controlled by the sympathetic nervous system. Increased skin conductance is a well-established physiological response to stress [41].

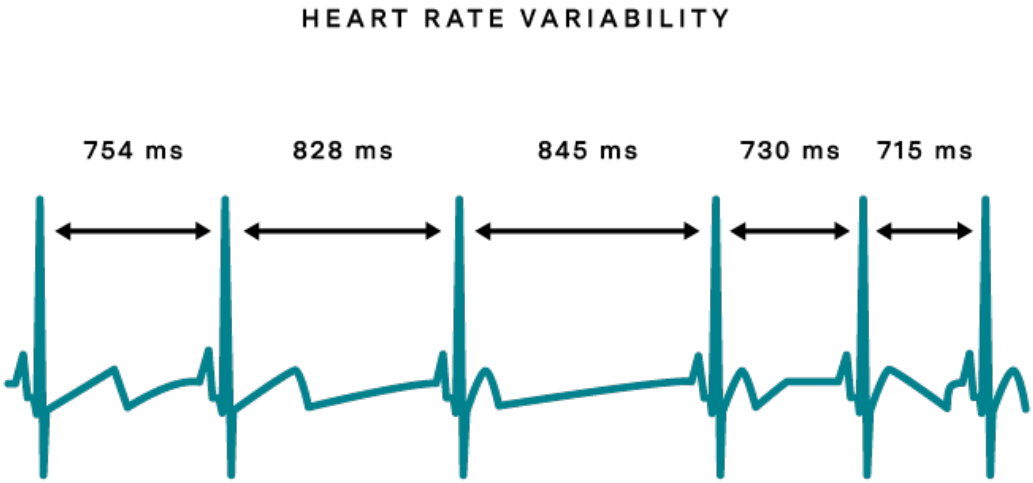


Figure 1.2: HRV

Functional Near-Infrared Spectroscopy (fNIRS)
fNIRS is a non-invasive method that measures changes in cerebral oxygenation related to stress. This technique is often combined with EEG to assess stress-related brain activity in real time [4].

Thermal Imaging
Stress can cause peripheral vasoconstriction, reducing temperature in the extremities while increasing facial temperature. Thermal imaging (Fig 1.3) detects these temperature changes and has been used in stress assessment, particularly in security and military applications.

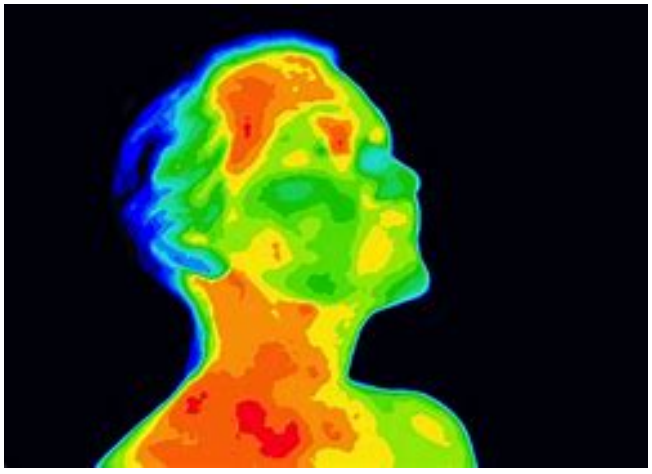


Figure 1.3: Thermal imaging

Face Images
When a person is under stress, eye, mouth, and head movements are different from normal situations, and research on stress recognition using general images is also being conducted. In these studies, many methods were used to recognize stress such as extracting hand-crafted features from the eyes, nose, and mouth areas or using eye size, mouth movements, and head movements as features [40].

Fig 1.4 shows a part from a database that used to evaluate stress with three levels

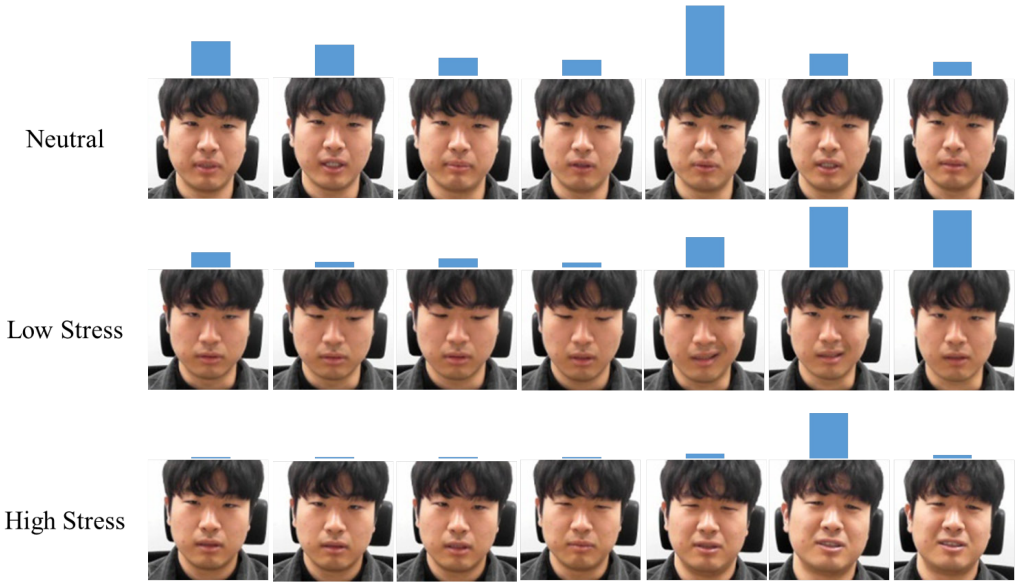


Figure 1.4: Face images for stress level detection [40]

Table 1.1, summarizes the comparison between classical stress measurement methods,

Table 1.1: Comparative Analysis of Classical Stress Measurement Methods

Method	**Advantages**	**Disadvantages**
Questionnaires		
NASA-TLX	Easy to administer; assesses multiple stress dimensions (mental, physical).	Subjective; prone to response bias.
BORG Scale	Simple, quick; useful for physical stress assessment.	Limited to perceived exertion, not cognitive/emotional stress.
STAI	Distinguishes between temporary and chronic anxiety (linked to stress).	Self-reported; may not correlate with physiological stress markers.
PSS	Measures perceived stress in daily life; widely validated.	Subjective; lacks real-time or biological validation.
Biomarkers		
Salivary Cortisol	Objective; reflects HPA axis activity; non-invasive collection.	Delayed response (peaks 20-30 mins post-stress); diurnal rhythm affects levels.
Salivary Alpha-Amylase	Rapid response (sympathetic activation); non-invasive.	Sensitive to food/drink; less specific to stress alone.
HRV	Real-time; indicates autonomic nervous system balance.	Requires precise ECG measurement; confounded by physical activity.

EDA	Directly measures sympathetic arousal; high temporal resolution.	Affected by temperature, humidity, and motion artifacts.
fNIRS	Non-invasive; measures brain oxygenation during stress tasks.	Expensive; limited spatial resolution compared to fMRI.
Thermal Imaging	Contactless; captures stress-induced temperature changes.	Sensitive to environmental conditions; requires controlled settings.
Face Image Analysis	Non-invasive; uses standard cameras; potential for real-time apps.	Privacy concerns; accuracy depends on lighting/pose variability.

1.5 Contribution of Neurophysiological Signals to Stress Assessment

In recent years, neurophysiological signal-based methods have gained attention due to their reliability in tracking stress responses in real-time. These include cardiovascular activity monitoring through electrocardiography (ECG) and photoplethysmography (PPG), cognitive and cerebral activity tracking using electroencephalography (EEG) and functional near-infrared spectroscopy (fNIRS), as well as skin conductance and temperature monitoring. Recent advancements have focused on portable, real-time stress monitoring systems that integrate machine learning models with physiological signals [4]. These developments aim to improve stress detection accuracy, especially in non-laboratory settings, enabling practical applications in everyday life.

A valuable contribution in this domain is the **SAM40** dataset, which consists of EEG recordings from 40 subjects undergoing various cognitive tasks designed to induce stress, such as the Stroop color-word test, arithmetic problem-solving, and mirror image recognition [21]. This dataset provides a structured approach to evaluating stress responses using EEG signals and serves as a benchmark for developing machine learning models for stress classification.

Methodology and Data Collection

The SAM40 dataset was collected using a 32-channel *Emotiv Epoc Flex gel kit*, with data recorded at a sampling rate of **128 Hz**. Participants were exposed to the following stress-inducing tasks:

- **Stroop Color-Word Test:** Measures cognitive interference and response inhibition by asking subjects to name the color of words, which can be either congruent or incongruent.
- **Mirror Image Recognition Task:** Requires participants to identify whether two presented images are symmetric or asymmetric, inducing cognitive load.
- **Arithmetic Problem Solving:** Subjects mentally solve mathematical problems and indicate whether a displayed solution is correct, increasing cognitive stress.

Each task was performed for **25 seconds**, followed by a short relaxation period. EEG signals were preprocessed using **Savitzky-Golay filtering** to remove baseline drifts and **wavelet thresholding** to eliminate artifacts.

EEG data plays a crucial role in stress assessment due to its high temporal resolution and direct measurement of brain activity. The SAM40 dataset enables researchers to analyze stress-related patterns in EEG signals and offers several advantages for stress detection:

- **Identification of Stress-Related Frequency Bands:** EEG signals reveal that stress is associated with increased **beta and gamma activity** (high cognitive load and alertness), while relaxation correlates with enhanced **alpha and theta activity** (reduced cognitive load and relaxation).
- **Objective Stress Measurement:** Unlike self-reported stress assessments, EEG provides a continuous and quantitative measure of stress levels by analyzing brainwave patterns.
- **Real-Time Monitoring:** The fast response time of EEG allows for real-time tracking of stress variations, which is crucial for applications in *workplace monitoring*, *mental health tracking*, and *cognitive load assessment*.
- **Machine Learning Integration:** The EEG data from SAM40 can be used to train deep learning models for automated stress classification, improving accuracy over traditional questionnaire-based methods.

1.6 Databases

In this section, we will briefly discuss several databases identified during our research.

1.6.1 SAM 40

SAM 40 is a dataset consisting of EEG recordings from 40 subjects (14 females, 26 males, with a mean age of 21.5 years) collected to monitor stress induced by cognitive tasks. The tasks performed include the Stroop color-word test, arithmetic problem-solving, and mirror image recognition. EEG signals were recorded using a 32-channel Emotiv Epoc Flex gel kit at a sampling frequency of 128 Hz. The data were processed to remove artifacts using wavelet thresholding and the Savitzky-Golay filter [21].

The primary objective of this dataset is to analyze short-term stress responses elicited by cognitive tasks. Subjects were exposed to different stress-inducing stimuli, and their EEG signals were segmented into 25-second epochs for further analysis. Behavioral stress ratings on a scale of 1 to 10 were also collected, allowing for a correlation between EEG patterns and perceived stress levels [21].

1.6.2 PASS

PASS (Physical Activity and Stress) is a multimodal database designed to support research on stress assessment in mobile environments. The dataset comprises data from 48 participants who performed tasks under varying stress levels while engaging in different levels of physical activity. To induce stress, participants played video games designed to evoke either calmness (TIMEframe) or stress (Outlast). Physiological signals, including EEG, electrocardiography (ECG), electrodermal activity (EDA), respiration, and skin temperature, were recorded throughout the experiment [41].

This dataset is particularly valuable for investigating the impact of movement artifacts and physical activity on stress detection. By incorporating wearable sensors, PASS provides a

realistic approach to stress monitoring in everyday scenarios, contributing to the advancement of mobile passive body/brain-computer interfaces [41].

1.6.3 Face Landmark-Based Stress Recognition Dataset

This dataset was developed for stress recognition using face images and facial landmarks. Unlike traditional stress detection methods that rely on biological signals or thermal imaging, which require specialized equipment, this dataset allows stress recognition using general camera-acquired images. A deep neural network was designed to process facial landmarks, focusing on eye, mouth, and head movements, which change under stress. Experimental results demonstrated the effectiveness of this approach in stress recognition [28].

1.7 Multimodal Signal Analysis

The assessment of stress through physiological and behavioral signals has gained significant attention in recent years. While single-modal approaches, such as EEG-based or ECG-based stress detection, provide valuable insights, multimodal signal analysis has demonstrated superior accuracy and robustness. By combining multiple physiological and behavioral signals, researchers can extract complementary information and improve stress recognition systems.

1.7.1 Multimodal Neurophysiological Signals

Multimodal approaches leverage a combination of neurophysiological and peripheral physiological signals, such as:

- **Electroencephalography (EEG)**: Captures brain activity and provides insight into cognitive load and emotional states. Stress is often associated with increased beta and gamma power and reduced alpha power in specific brain regions [21].
- **Electrocardiography (ECG)**: Measures heart rate variability (HRV), where lower HRV is typically linked to higher stress levels [41].
- **Electrodermal Activity (EDA)**: Reflects changes in skin conductance caused by autonomic nervous system activation during stress responses.
- **Respiration and Skin Temperature**: Variations in breathing rate and peripheral temperature can indicate stress-induced physiological changes.

Several datasets, such as **SAM 40** [21] and **PASS** [41], have been developed to facilitate multimodal stress analysis, providing EEG, ECG, EDA, and other signals recorded under controlled experimental conditions.

1.7.2 Multimodal Behavioral Analysis

Beyond physiological measurements, behavioral signals also play a crucial role in stress detection. These include:

- **Facial Expressions and Microexpressions**: Stress alters facial muscle activity, which can be analyzed using deep learning models applied to face images [28].
- **Posture and Body Movements**: Stress-induced changes in posture and movement patterns can be detected using motion sensors and cameras.

- **Speech and Voice Features**: Variations in pitch, speech rate, and vocal tremors are indicative of emotional and physiological stress responses.

The **Face Landmark-Based Stress Recognition Dataset** [28] provides a novel approach by utilizing facial landmarks to detect stress, eliminating the need for specialized biosensors.

1.7.3 Fusion of Multimodal Data

Multimodal stress assessment integrates multiple data streams using various fusion techniques:

- **Feature-Level Fusion**: Extracted features from different modalities are combined into a single feature vector before classification.

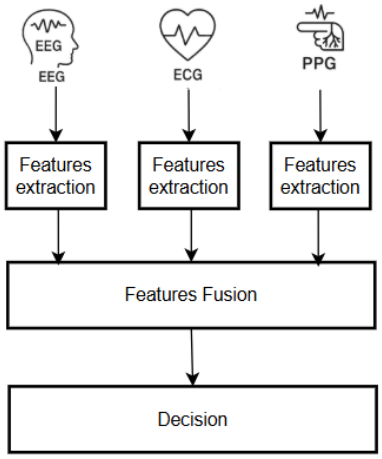


Figure 1.5: Features fusion

- **Decision-Level Fusion**: Independent classifiers are applied to each modality, and their outputs are combined to improve accuracy.

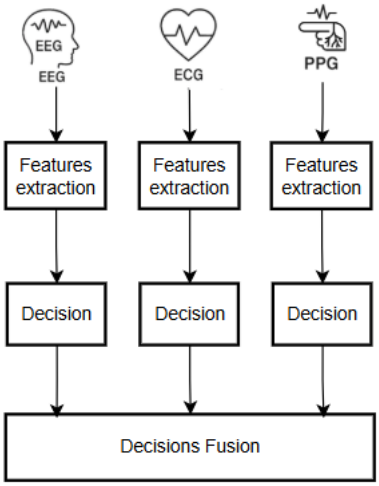


Figure 1.6: Decision fusion

- **Deep Learning-Based Fusion**: Convolutional and recurrent neural networks are increasingly used to automatically learn multimodal feature representations.

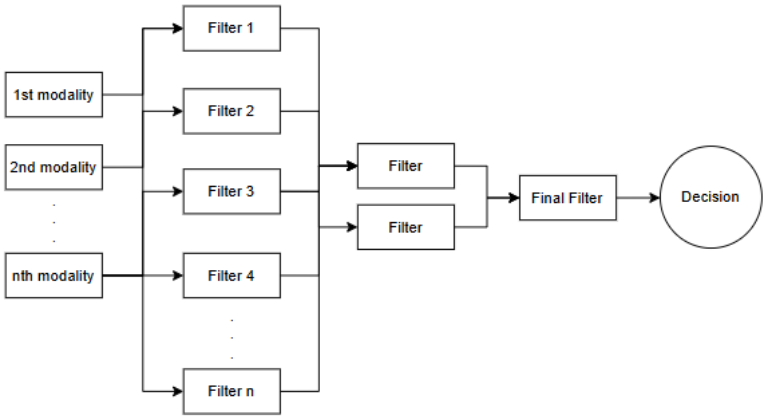


Figure 1.7: Deep Learning-Based Fusion

Recent studies have demonstrated that deep learning models trained on multimodal physiological and behavioral data can achieve higher stress detection accuracy than unimodal approaches.

1.7.4 Some multimodal signal analysis approaches

To assess stress using neurophysiological signals, research suggests that combining multiple signals yields better results than using them individually.

Starting with **Cognitive Stress Detection during Physical Activity using Simultaneous, Mobile EEG and ECG signals** (Maria et al.)[46], after segmenting the PASS EEG and ECG signals, a visibility graph representation was applied. Subsequently, shape-based features (Hu's moments and Zernike's moments) and frequency-based features (wavelet packet features) were extracted to train a machine learning model. The models used include LightGBM (LGBM), Random Forest (RF), and Gaussian Naïve Bayes (GNB). The classification results are summarized in Table 1.8.

Signal	Model					
	LGBM		RF		GNB	
	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score
EEG	0.91	0.90	0.81	0.77	0.77	0.69
ECG	0.85	0.89	0.75	0.74	0.73	0.71
EEG+ECG	0.93	0.92	0.85	0.85	0.82	0.82

Figure 1.8: Stress classification results for Cognitive Stress Detection during Physical Activity using Simultaneous Mobile EEG and ECG signals [46].

From these results, it is evident that combining signals leads to the highest accuracy. Among the tested models, LGBM achieved the best performance, confirming its effectiveness in this stress classification method.

Now, with **Stress Recognition using Face Images and Facial Landmarks** (Jung et al.) [27], the proposed algorithm begins with face and facial landmark detection for stress recognition. A deep learning-based three-stage network enhances face detection accuracy, while facial landmarks are extracted using a cascade method combining random-fern features and a regression tree classifier. The overall framework is illustrated in Fig 1.9.

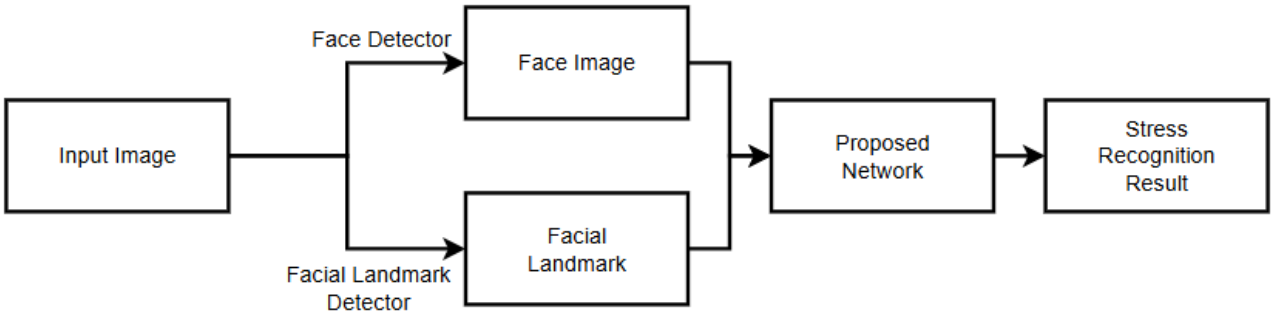


Figure 1.9: Flowchart of the overall framework from the stress recognition using face images [27]

The network classifies stress into three levels: no stress, weak stress, and strong stress (Fig 1.10). To optimize performance, shortcut mapping enhances deep network training, while a bottleneck architecture reduces internal parameters and computational cost without sacrificing accuracy.

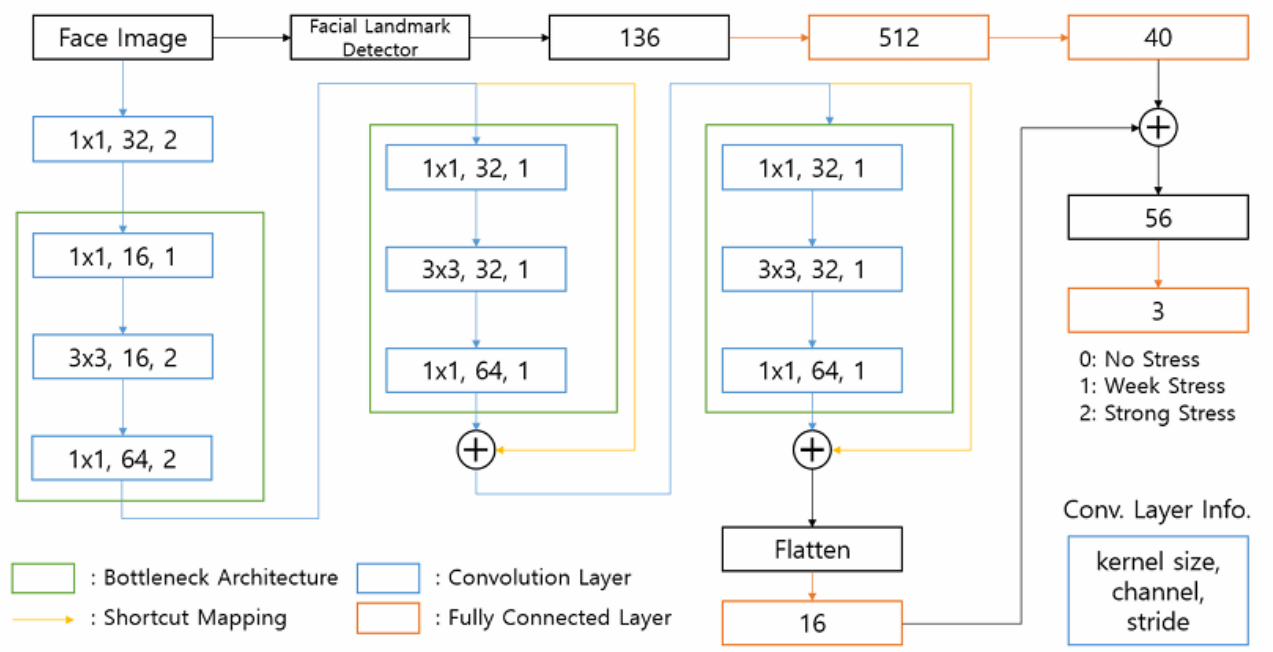


Figure 1.10: Structure of the proposed network of stress recognition using face images [27]

Performance evaluation was conducted using k-fold cross-validation, where one subject's data was used for testing while the remaining subjects' data trained the model. Results demonstrated that including facial landmarks significantly improved accuracy, with the best performance achieved using grayscale face images of an optimal size (Table 1.11).

Input (Image Size)	Accuracy
Color face image (84)	33.68 %
Color face image (84) and facial landmarks	56.14 %
Gray face image (84)	57.92 %
Gray face image (84) and facial landmarks	63.29 %
Gray face image (140) and facial landmarks	**64.63 %**

Figure 1.11: Accuracy from Jung et al. [27]

Jung et al. [27] confirmed that facial landmarks enhance stress recognition by capturing key stress-related facial movements, such as those of the eyes, mouth, and head. Their findings also highlight the effectiveness of grayscale images in improving classification accuracy by better isolating stress-related features, but, the Maria et al [46] method shows better accuracy than the Jung et al. [27] method, in this case, we can absolutely confirm that neurophysiological signals play the best role in stress assessment, especially when it is multimodal.

1.8 Conclusion

Numerous studies have explored stress detection and assessment using various modalities. As highlighted in this chapter, neurophysiological signals have demonstrated the highest accuracy for cognitive stress evaluation, particularly when combined in a multimodal approach. The integration of multiple physiological and behavioral signals enhances the reliability and robustness of stress recognition systems.

In the next chapters, we will focus on signal acquisition and processing methods, along with implementation and experimental validation. By refining feature extraction techniques and exploring the potential of incorporating additional modalities, we aim to further improve stress classification accuracy and advance real-time stress assessment methodologies.

Chapter 2

Computational Methods for Biomedical Signal Processing and Classification

2.1 Introduction

Previously, we explored stress, its various types, and its impact on health and cognition. We examined how stress negatively affects our well-being, daily life, mood, and mental processes, as well as its influence on neurophysiological signals in the human body. Additionally, we discussed classical methods for assessing and measuring stress and reviewed existing databases that aid in stress evaluation.

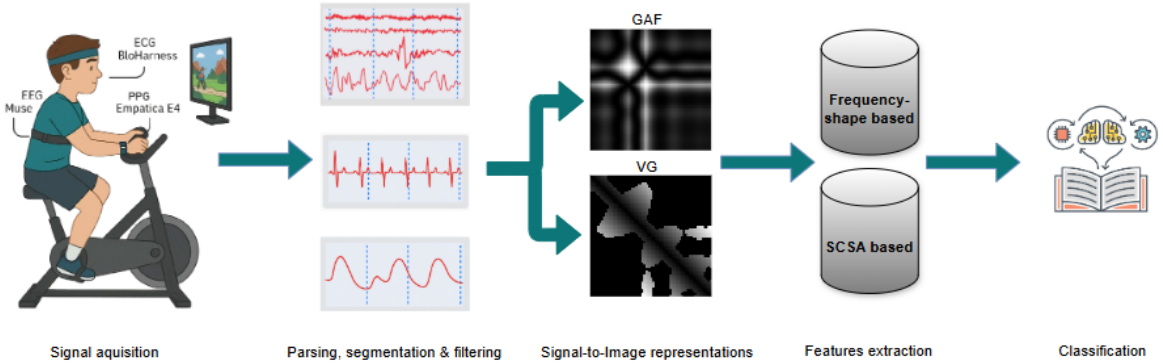


Figure 2.1: Methodology's full pipeline

In this chapter, we will explore existing techniques for noise reduction and artifact removal, methods for signal representation through imaging, and feature extraction strategies. Furthermore, we will delve into multimodal data fusion and the application of machine learning models.

Fig 2.1 shows the full pipeline of detailed process in this chapter.

2.2 Preprocessing Methods

In this section, we will discuss the commonly used methods for signal preprocessing, noise removal, filtering, and artifact correction for neurophysiological signals, specifically EEG, ECG, and PPG. In the next chapter, we will explain why these three signals were chosen for stress assessment.

2.2.1 Independent Component Analysis (ICA)

Independent Component Analysis (ICA) is a statistical technique used for separating multivariate signals into independent non-Gaussian components as shown in Fig 2.2. ICA is widely applied in fields like audio, image processing, and biomedical signal analysis such as EEG signal to isolate distinct sources from mixed signals and to remove artifacts such as eye blinks, muscle movements, and power line noise.[17]

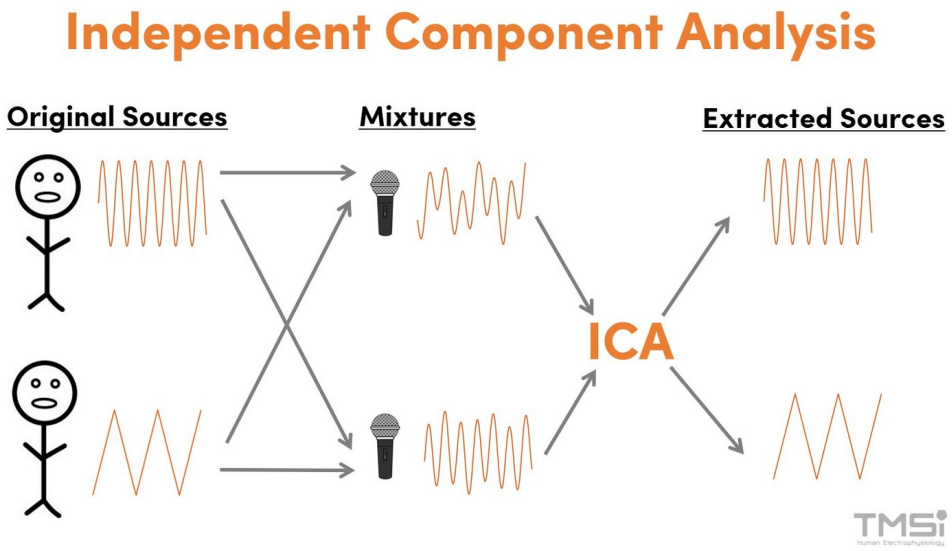


Figure 2.2: Independent Component Analysis illustration[25]

2.2.1.1 Basic concepts

Statistical independence refers to the idea that two random variables, X and Y, are independent if knowing one does not affect the probability of the other. Mathematically, this means the joint probability of X and Y is equal to the product of their individual probabilities:

$$P(X \text{ and } Y) = P(X) \cdot P(Y) \tag{2.1}$$

or equivalently,

$$P(X \cap Y) = P(X) \cdot P(Y) \tag{2.2}$$

In addition, ICA operates under some key assumptions, the source signals are statistically independent of each other, they have non-Gaussian distributions, propagation delay is negligible,the signals at the sensors are different linear combinations of the sources and finally the number of sources is equal to the number of sensors. These assumptions allow ICA to effectively separate mixed signals into independent components, a task that traditional methods like PCA cannot achieve.

2.2.1.2 The ICA algorithm for EEG signal

Data Preprocessing, EEG signals are bandpass filtered (typically 0.5–50 Hz) to remove low-frequency drifts and high-frequency noise before ICA is applied.

Signal Model (Linear Mixing), the observed EEG signals are modeled as a linear mixture of independent sources:

$$\mathbf{x} = \mathbf{As} \tag{2.3}$$

where:

- $\mathbf{x} = [x_1, x_2, ..., x_n]^T$ are the observed signals,
- $\mathbf{s} = [s_1, s_2, ..., s_n]^T$ are the independent source signals,

- **A** is the unknown mixing matrix.

Whitening the Data, to decorrelate the signals, we first center the data:

$$\mathbf{x}' = \mathbf{x} - E\{\mathbf{x}\} \tag{2.4}$$

Then, we whiten the data using the covariance matrix:

$$\mathbf{V}\mathbf{D}\mathbf{V}^T = E\{\mathbf{x}'\mathbf{x}'^T\} \tag{2.5}$$

where:

- **V** is the matrix of eigenvectors,
- **D** is the diagonal matrix of eigenvalues.

The whitening transformation matrix is:

$$\mathbf{P} = \mathbf{V}\mathbf{D}^{-1/2}\mathbf{V}^T \tag{2.6}$$

The whitened signals are:

$$\mathbf{x}^* = \mathbf{P}\mathbf{x}' \tag{2.7}$$

Finding Independent Components, we need to estimate a separation matrix **M** such that:

$$\mathbf{s} = \mathbf{M}\mathbf{x}^* \tag{2.8}$$

This is done by maximizing the non-Gaussianity of **s**. A common measure of non-Gaussianity is Kurtosis:

$$\text{Kurt}(y) = E\{y^4\} - 3(E\{y^2\})^2 \tag{2.9}$$

Iterative Algorithm for ICA, the FastICA algorithm by Hyvärinen and Oja is used:[37]

1. Choose an initial random weight vector **m**.
2. Update **m** using a nonlinear function $g(y)$:

$$\mathbf{m}^+ = E\{\mathbf{x}^* g(\mathbf{m}^T\mathbf{x}^*)\} - E\{g'(\mathbf{m}^T\mathbf{x}^*)\}\mathbf{m} \tag{2.10}$$

3. Normalize **m**:

$$\mathbf{m} = \frac{\mathbf{m}^+}{\|\mathbf{m}^+\|} \tag{2.11}$$

4. Repeat until convergence (when the weight vector change is below a threshold).

Reconstructing Independent Sources, after estimating **M**, the independent components are:

$$\mathbf{s} = \mathbf{M}\mathbf{x}^* \tag{2.12}$$

These sources correspond to the original EEG signals without artifacts, as we can see in Fig 2.3.

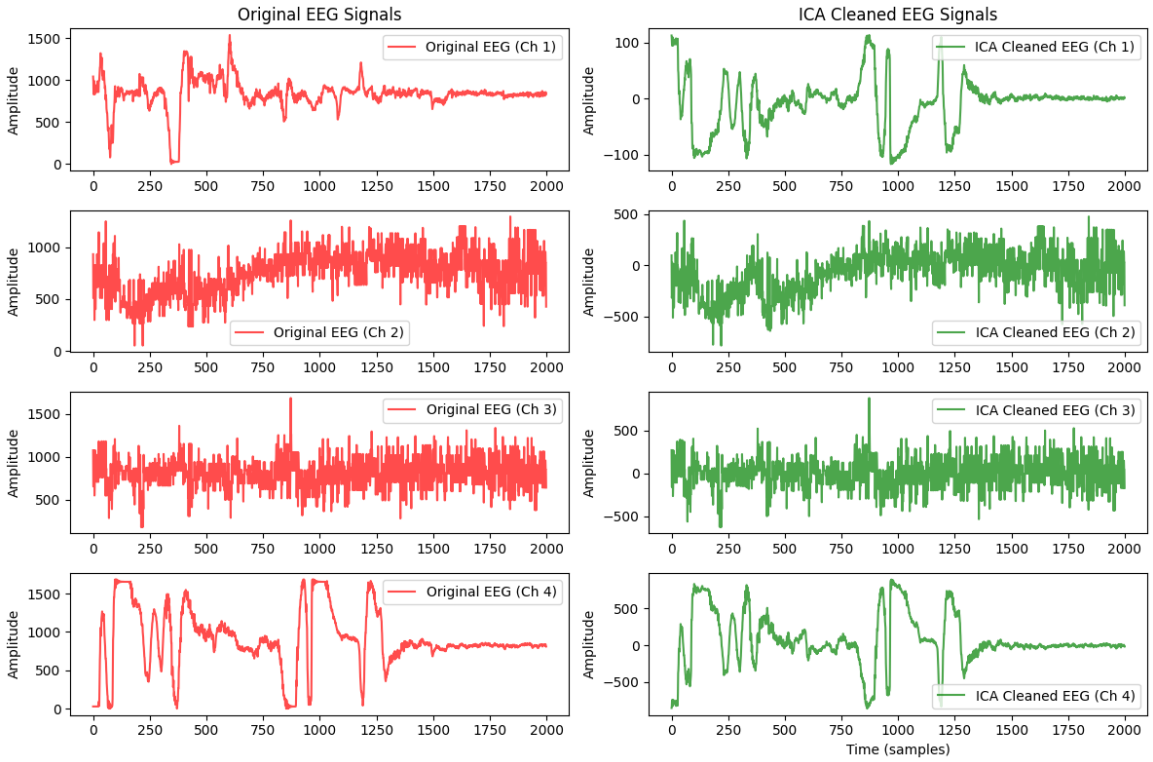


Figure 2.3: Example of ICA application for the PASS dataset

However, according to a study [11], ICA alone is not sufficient for artifact removal. In fact, it may also eliminate parts of the neural signal, leading to distortions. To address this issue, the article introduces wICA, an enhanced method specifically designed for preserving neural activity while effectively suppressing artifacts. In the next subsection, we will explore this approach in detail.

Fig 2.3 shows clearly that ICA is not sufficent, and we need to improve it by using wICA, so what is wICA.

2.2.2 wavelet-enhanced independent component analysis wICA

Let us now present the wICA method, which leverages Wavelet technique to enhance the performance of the ICA artifact suppression technique.[11]

When working with real EEG data, ICA (Independent Component Analysis) often estimates independent components that not only capture strong artifacts but also frequently include a significant amount of cerebral activity. This issue can arise due to factors such as the limitations on the maximum number of independent sources, temporal independence constraints, or suboptimal application of the algorithm [11]. Rejection of such components supposes a loss of a part of the cerebral activity and, consequently, distortion of the artifact free EEG

According to ICA assumptions, this component should not contain any artifacts independent of ocular sources. The component can be divided into a high-amplitude artifact, $a(t)$ and a low-amplitude residual neural signal, $n(t)$

$$s_1(t) = a(t) + n(t) \tag{2.13}$$

In the conventional ICA algorithm, the entire component is set to zero, i.e., $s_1(t) = 0$, before the

signal recomposition, as shown in Eq. 2.12. As a result, in the ICA-corrected EEG, a portion of the cerebral activity is lost:

$$\hat{x}_j(t) = r_j(t) - m_{j1}n(t), \tag{2.14}$$

where $r_j(t) = x_j(t) - m_{j1}a(t)$ represents the artifact-free signal, and m_{j1} is the corresponding weight from the mixing matrix, M.

From Eq. 2.13, the decomposition of the independent component into artefactual and neural activity is unknown. However, using properties of the signals a(t) and n(t) we can estimate them.

The artifact, a(t), has high magnitude (power) and is localized in the time and/or in frequency domains, while n(t) is of low amplitude and has a broad band spectrum. These properties fit well with the wavelet decomposition technique that provides an optimal resolution both in the time and frequency domains, without requiring the signal stationarity, (we define in details the wavelet decomposition, in appendices).

The method begins with the equation :

$$W_s(d,b) = W_a(d,b) + W_n(d,b) \tag{2.15}$$

where the wavelet transform of a signal $s_1(t)$ is defined by:

$$W_s(d,b) = \frac{1}{\sqrt{d}} \int s_1(t)\psi_{d,b}(t)\, dt, \quad \psi_{d,b}(t) = \psi\left(\frac{t-b}{d}\right) \tag{2.16}$$

And $W_s(d,b)$ is the wavelet transform of $s_1(t)$, and $W_a(d,b)$ and $W_n(d,b)$ are the wavelet coefficients obtained by the transformation (2.16) of the artificial and neural parts of the component, respectively. The function ψ represents the mother wavelet, with b and d defining the time localization and scale.

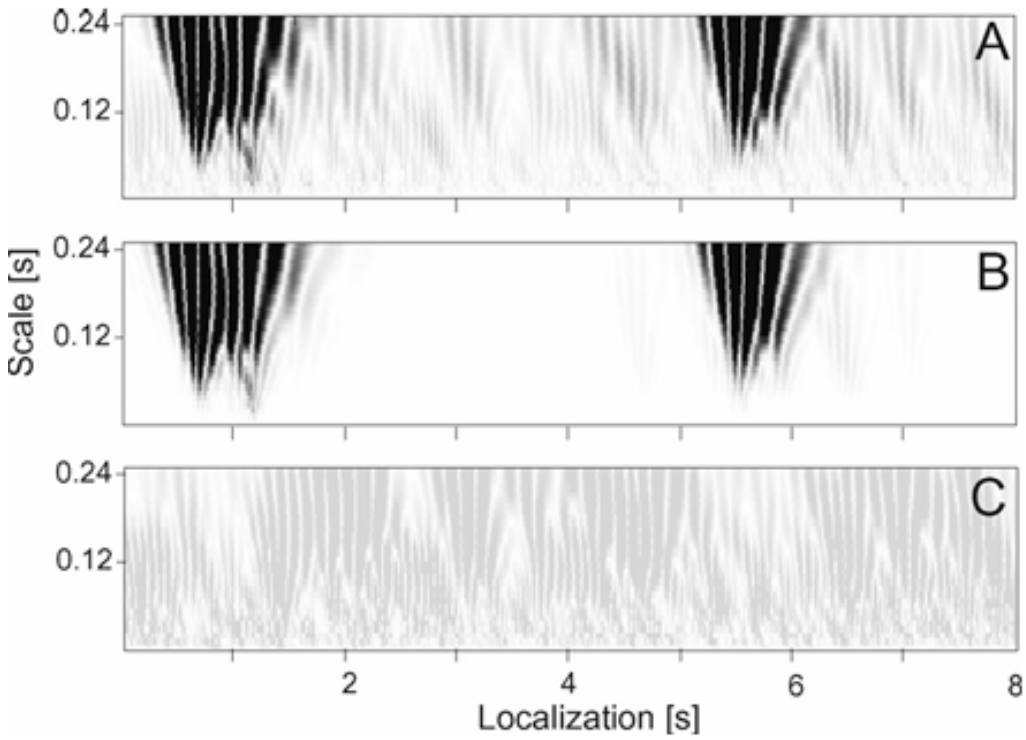


Figure 2.4: Example of wavelet representation of the independent component $s_1(t)$ (A) and its parts: artefactual $a(t)$ (B), and neural $n(t)$ (C).Gray intensity codifies the absolute value of wavelet coefficients. [11]

As shown bellow, in Fig 2.4A which is $W_s(d, b)$, the wavelet representation of $s(t)$, Fig 2.4B is $W_a(d, b)$, which is the wavelet representation of the artefactual $a(t)$ and Fig 2.4C is $W_n(d, b)$, which is the wavelet representation of the neural signal $n(t)$. Artifacts are suppressed by thresholding high-amplitude coefficients, allowing better preservation of the low-amplitude neural signal.

The selection of the threshold value, K, is a crucial element of the algorithm. Here, we use the simplest fixed-form threshold:

$$K = \sqrt{2 \log N}\, \sigma, \tag{2.17}$$

where N is the length of the data segment to be processed, and

$$\sigma^2 = \frac{\text{median}(|W(d, b)|)}{0.6745} \tag{2.18}$$

is the estimator of the magnitude of the neural wide-band signal part.

In resume, this is the steps for wICA algorithm:

1. **ICA Decomposition:** Apply conventional ICA to the raw EEG signals to obtain the mixing matrix M and a set of N independent components:

$$\{s_1(t), s_2(t), \ldots, s_N(t)\}.$$

2. **Wavelet Transformation:** Compute the wavelet transform of each independent component, obtaining the wavelet representations $W(j, k)$ for each $s_i(t)$.

3. **Thresholding:** Suppress artifact-dominated wavelet coefficients by setting:

$$W(j,k) = 0, \quad \text{for all } |W(j,k)| > K.$$

4. **Reconstruction of Neural Components:** Perform the inverse wavelet transform on the thresholded coefficients to reconstruct components containing primarily neural sources:

$$\{n_1(t), n_2(t), \ldots, n_N(t)\}.$$

5. **Recomposition of EEG Signals:** Reconstruct the artifact-suppressed EEG by applying the mixing matrix M to the cleaned neural components:

$$\tilde{\mathbf{X}} = M \cdot [n_1(t), n_2(t), \ldots, n_N(t)]^T.$$

Here is an example of the wICA application for an EEG data from the PASS database,

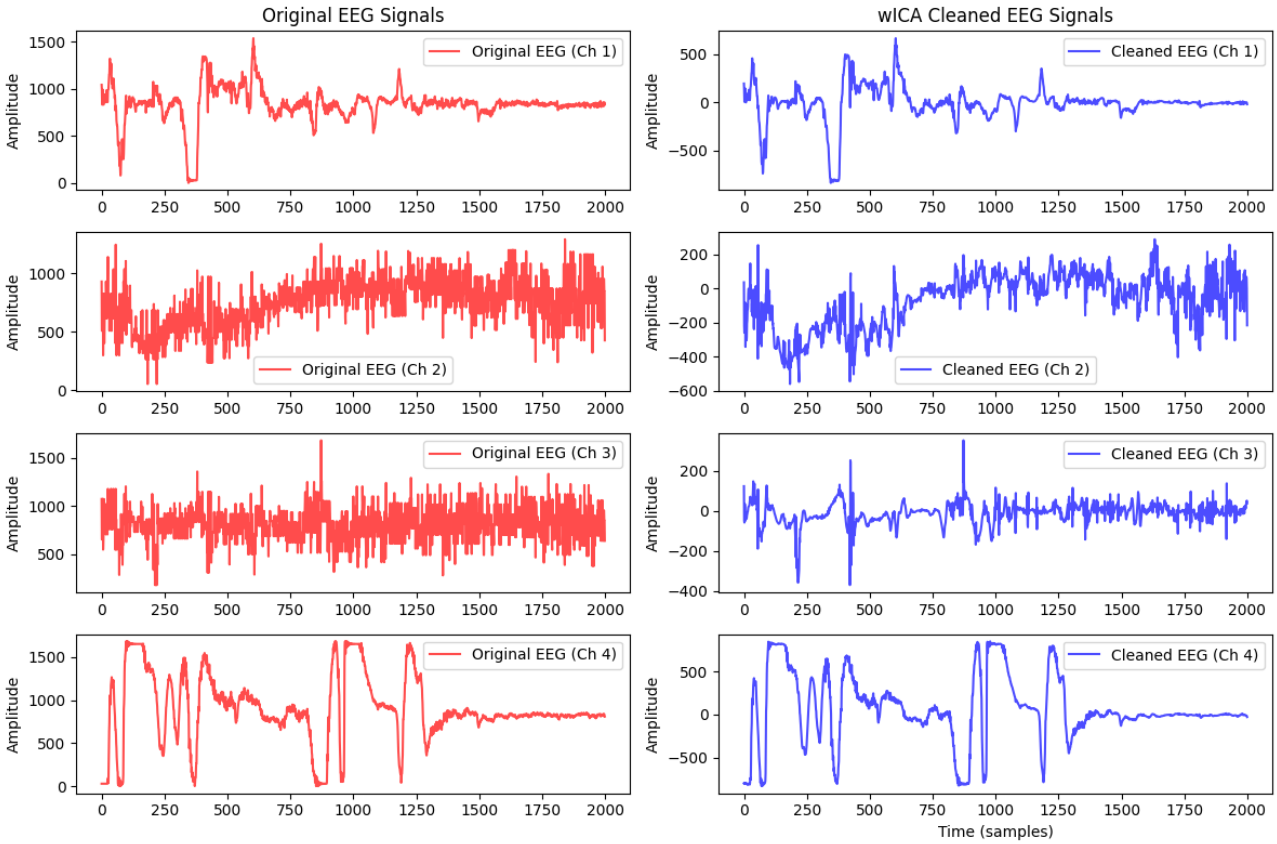


Figure 2.5: Example of wICA application for the PASS dataset

2.2.3 Semi-Classical Signal Analysis (SCSA)

The Semi-Classical Signal Analysis (SCSA) is a spectral decomposition method based on the Schrödinger operator. It has been widely applied for signal denoising and feature extraction in biomedical signals, particularly in ECG and PPG data processing [32, 33].

The main idea behind SCSA is to represent a signal $s(x)$ as a potential function in a Schrödinger operator:

$$H_h\psi(x) = -h^2\frac{d^2\psi(x)}{dx^2} - s(x)\psi(x) = \lambda\psi(x), \tag{2.19}$$

where:

- H_h is the Schrödinger operator,
- $s(x)$ is the input signal,
- h is the semi-classical parameter controlling the decomposition,
- λ are the eigenvalues, and $\psi(x)$ are the corresponding eigenfunctions.

Instead of using Fourier transforms, SCSA reconstructs the signal from the only negative eigenvalues of the Schrödinger operator.

Algorithm for 1D-SCSA

The SCSA algorithm consists of the following steps:

1. **Construct the Schrödinger Operator:** Using for example the ECG signal $s(x)$ as the potential function, and here is the discretised Schrödinger operator:

$$H_h = -h^2 D_2 - S, \tag{2.20}$$

   where D_2 is the second derivative operator, and S is the diagonal matrix $s(x)$.

2. **Solve the Eigenvalue Problem:** Compute the eigenvalues and eigenfunctions:

$$H_h \psi_n = \lambda_n \psi_n. \tag{2.21}$$

   Select only the eigenfunctions corresponding to negative eigenvalues $\lambda_n < 0$.

3. **Reconstruct the Signal $s_h(x)$:** The reconstructed signal is given by:

$$s_h(x) = 4h \sum_{n=1}^{N_h} \sqrt{|\lambda_n|} \psi_n^2(x), \tag{2.22}$$

   where N_h is the number of selected eigenfunctions.

4. **Select Optimal h:** Adjust h to balance noise removal and feature preservation.

   The parameter h plays a crucial role in SCSA:

   - **Low h (0.05 - 0.1):** Preserves more signal details with minimal denoising.
   - **Medium h (0.2 - 0.5):** Provides a balance between noise reduction and detail retention.
   - **High h (0.5 - 1.0):** Stronger noise filtering but may smooth out fine details.

The Semi-Classical Signal Analysis (SCSA) method not only denoises signals but also enables feature extraction by analyzing the eigenvalues and eigenfunctions of the Schrödinger operator. These extracted features can be applied to classification, pattern recognition, and biomedical signal analysis, such as PPG and ECG analysis [32, 33].

2D-SCSA for images [29]

The Two-Dimensional Semi-Classical Signal Analysis (2D-SCSA) is a spectral method based on the decomposition of an image using the Schrödinger operator and its associated eigenfunctions. It exploits the semi-classical quantization approach to extract relevant features while reducing noise and improving contrast.

Mathematically, the 2D-SCSA is formulated through the semi-classical Schrödinger operator eigenvalue problem:

$$\mathcal{H}_h(V)\psi_{k,h} = \mu_{k,h}\psi_{k,h}, \quad \text{with} \quad \mathcal{H}_h(V) = -h^2\Delta - V, \tag{2.23}$$

where:

- $V(x, y)$ represents the input image (positive potential),
- $\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$: the 2D Laplacian operator,
- h is the semi-classical parameter controlling the spectral resolution,
- $\mu_{k,h}$ are the negative eigenvalues ($\mu_{1,h} < \cdots < \mu_{K_h^\lambda,h} < \lambda$),
- $\psi_{k,h}$ are the associated L^2-normalized eigenfunctions.

The image reconstruction is performed by summing a weighted subset of eigenfunctions through the spectral expansion formula:

$$V(x, y) = -\lambda + \lim_{h\to 0}\left(\frac{h^2}{L_{2,\gamma}^{cl}}\sum_{k=1}^{K_h^\lambda}(\lambda - \mu_{k,h})^\gamma \psi_{k,h}^2(x, y)\right)^{\frac{1}{1+\gamma}}, \tag{2.24}$$

where $L_{2,\gamma}^{cl} = \frac{1}{4\pi}\frac{\Gamma(\gamma+1)}{\Gamma(\gamma+2)}$ is a semi-classical constant and Γ is the Gamma function,

$$h \in \mathbb{R}^{*+}, \quad \gamma \in \mathbb{R}^+, \quad \lambda \in \mathbb{R}^-$$

Algorithm Implementation:
1. *Operator Splitting*: The 2D problem is decomposed into 1D operators for rows and columns:

$$A_{i,h} = -h^2 D_2 - \text{diag}(\frac{1}{2}I[i, :]), \quad \text{(row operator)} \tag{2.25}$$

$$B_{j,h} = -h^2 D_2 - \text{diag}(\frac{1}{2}I[:, j]), \quad \text{(column operator)} \tag{2.26}$$

where D_2 is the 1D Laplacian discretization matrix.

2. *Eigenvalue problem Solution*: For each row i and column j, solve:

$$A_{i,h}\phi_{i,n,h} = \kappa_{i,n,h}\phi_{i,n,h}, \tag{2.27}$$

$$B_{j,h}\phi_{j,m,h} = \rho_{j,m,h}\phi_{j,m,h}. \tag{2.28}$$

3. *Tensor Product Reconstruction*: Reconstruct each pixel (i, j) via:

$$I_{rec}[i, j] = -\lambda + \left(\frac{h^2}{L_{2,\gamma}^{cl}}\sum_{n=1}^{N_h^\lambda}\sum_{m=1}^{M_h^\lambda}(\lambda - (\kappa_{i,n,h} + \rho_{j,m,h}))^\gamma \phi_{i,n,h}^2[j]\phi_{j,m,h}^2[i]\right)^{\frac{1}{1+\gamma}}. \tag{2.29}$$

The main advantages of this method are:
- *Noise reduction*: By truncating small eigenvalues, high-frequency noise is filtered while preserving structural information (PSNR improvements of 4-6 dB reported in [29]).
- *Contrast enhancement*: The spectral representation naturally highlights edges and textures through dominant eigenfunctions.
- *Data compression*: Efficient representation using $O(\sqrt{N})$ eigenmodes for $N \times N$ images.

This approach has been successfully applied in biomedical imaging (e.g., MRI denoising with 34.5 dB PSNR) and pattern recognition tasks. The semi-classical parameter h (typically $\in [0.1, 0.5]$) provides a trade-off between reconstruction accuracy and computational cost, while $\gamma = 1$ is commonly used for general applications.

Parameter	Value
Optimal h	0.2
γ	4
PSNR improvement	3 - 8 dB
$N_h^\lambda = M_h^\lambda$	≈ 400

Table 2.1: Typical 2D-SCSA performance on 512×512 images [29]

2.3 Signal-to-Image Representation

Signal-to-image representation is a technique used to transform time-series signals into visual formats, enabling the application of image-based deep learning models for classification and feature extraction. This transformation helps leverage convolutional neural networks (CNNs) and other vision-based AI models for stress assessment and biomedical signal analysis.

2.3.1 Visibility Graph

The Visibility Graph (VG) is a nonlinear time-series analysis method that converts a signal into a graph representation, where each data point in the time series is mapped to a node, and edges are formed based on a visibility criterion. This method enables the characterization of complex physiological patterns and enhances feature extraction for classification tasks.

Graph Construction: Each point in the time-series signal is considered a node, and edges are formed between nodes that have direct visibility based on a geometric criterion. Two data points and in a time series are visible to each other if every intermediate point satisfies:[46]

$$y_c < y_a + (y_b - y_a)\frac{t_c - t_a}{t_b - t_a} \tag{2.30}$$

With this equation we can easily generate a visibility graph from time series as Fig 2.6 shows

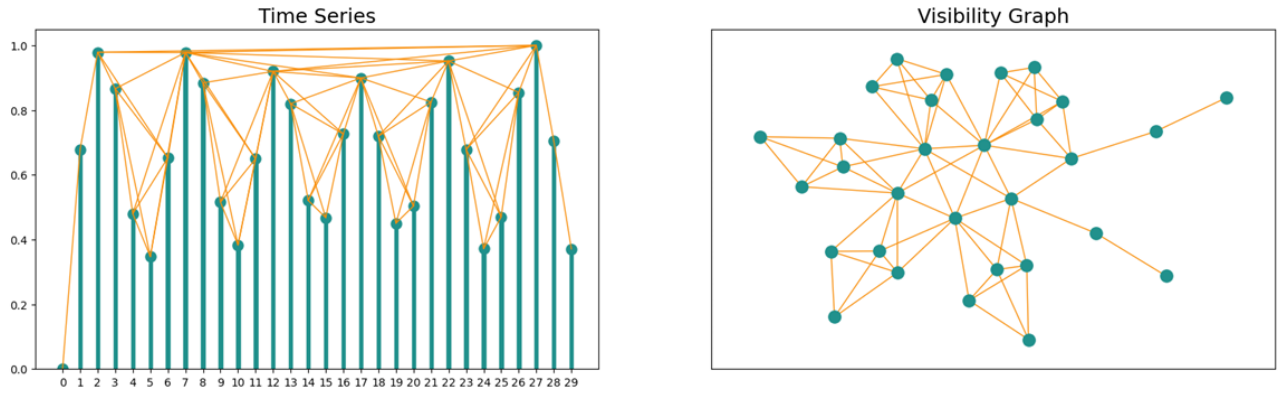


Figure 2.6: Illustration of Visibility Graph network mapped from an analytically generated time-series [46]

Limited Penetrable Visibility Graph (LPVG) Approach:

Based on the study from the IFAC stress paper, LPVG is an advanced version of VG that allows connections between nodes even if one or more intermediate data points exist between them. This makes LPVG more robust to noise in biomedical signals like EEG and ECG.[46]

Limited Penetration Parameter (p): The parameter allows two nodes to be connected even if some points exist between them, improving robustness to noise. Higher values of help to ignore intermediate data points that might be noise while maintaining essential signal structure.

Adjacency Matrix Representation: The adjacency matrix is constructed, where if two nodes are connected and otherwise. This matrix encodes the graph topology and is used for image-based analysis. The adjacency matrix is visualized as an image, enabling CNN-based analysis. The LPVG representation enhances the ability to detect stress-related patterns in EEG and ECG data.

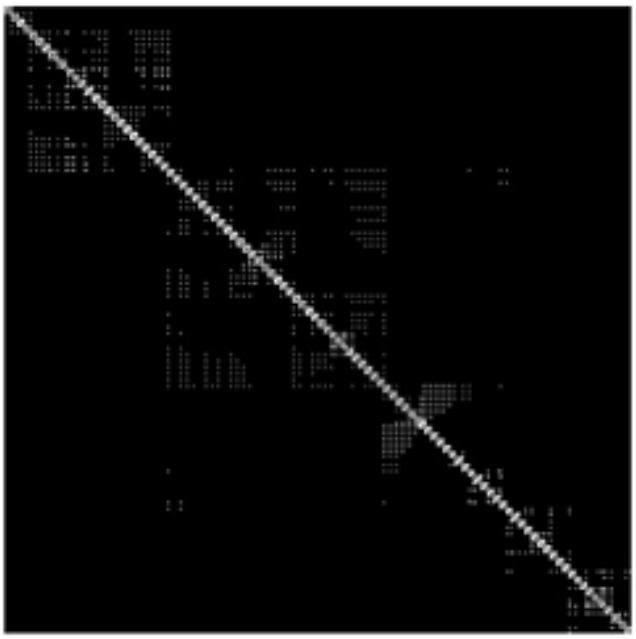


Figure 2.7: Visibility graph image representation [46]

Feature Extraction: Graph-based metrics such as degree distribution, clustering coefficient, betweenness centrality, and entropy are extracted to quantify the complexity of the physiological signal.

Finally, the LPVG approach, as detailed in the IFAC stress paper, has demonstrated effectiveness in stress classification by transforming physiological signals into structured images. This method captures long-term dependencies and non-linear interactions, making it highly suitable for analyzing cognitive stress states.

2.3.2 Gramian Angular Field

The *Gramian Angular Field (GAF)* is a technique for encoding **time-series data as images** by representing temporal correlations through an angular transformation [51]. This method converts a **1D time-series** into a **2D matrix**, enabling the application of computer vision and deep learning techniques for pattern recognition.

The GAF method transforms a normalized time series into a **polar coordinate system** to preserve the temporal dependencies in a geometric form.

Each time-series value is rescaled between and and then mapped to an angle in the **polar coordinate system** using:

$$x'_i = \frac{x_i - \min(X)}{\max(X) - \min(X)} \tag{2.31}$$

$$\phi_i = \arccos(x'_i) \tag{2.32}$$

where:

x'_i: is the normalized time-series value.

ϕ_i: is the angular representation.

There is two types of **Gramian Angular Fields (GAFs)** can be constructed:

1. Gramian Angular Summation Field (GASF)

$$GASF_{i,j} = \cos(\phi_i + \phi_j) \tag{2.33}$$

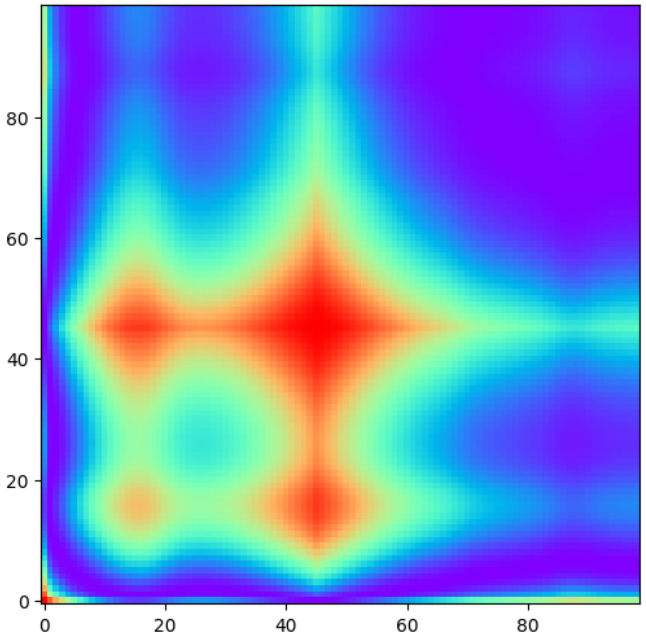


Figure 2.8: Gramian Angular Summation Field (GASF)

In Fig 2.8, we see an application of the Gramian Angular Summation Field in one of the ECG signals.

2. Gramian Angular Difference Field (GADF)

$$GADF_{i,j} = \sin(\phi_i - \phi_j) \tag{2.34}$$

These matrices encode the **temporal relations** between different time steps.

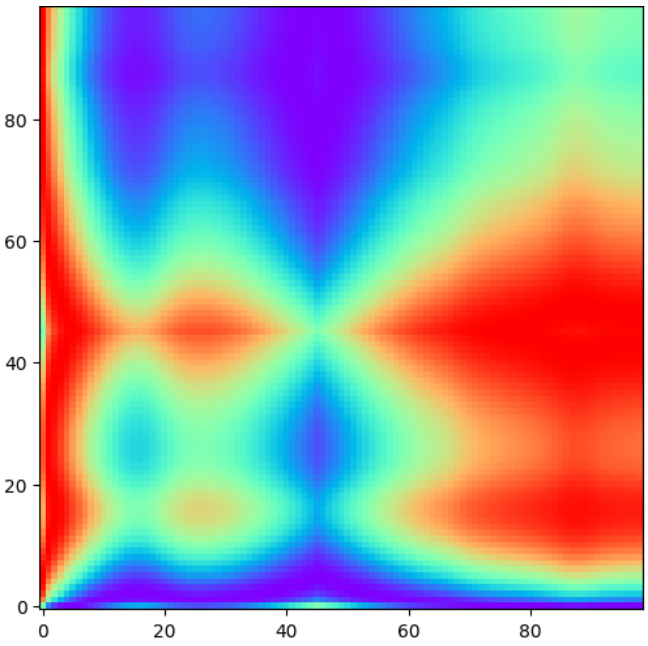


Figure 2.9: Gramian Angular Difference Field (GADF)

In Fig 2.9, we see an application of the Gramian Angular Difference Field in one of the ECG signals.

2.4 Feature Extraction

What's feature engineering? and feature extraction?

Feature engineering is the process of transforming raw data into meaningful features that improve machine learning model performance, involving techniques like scaling, encoding, and creating new variables. **Feature extraction**, a subset of feature engineering, focuses on automatically reducing data dimensionality by extracting the most relevant information from raw inputs, often converting complex data into compact, representative numerical formats.

2.4.1 Feature Extraction with SCSA

The spectral properties of the Schrödinger operator provide a unique way to extract meaningful features from signals. The key features derived from SCSA include:

- **Negative Eigenvalues** λ_n
    - These eigenvalues represent the **energy levels** of the signal.
    - The **distribution of** λ_n gives insight into signal complexity and structure.
    - Useful for EEG pattern classification (e.g., distinguishing normal vs. seizure signals).
- **Sum of Negative Eigenvalues** A global descriptor of the signal:

$$\sum_{n=1}^{N_h} |\lambda_n| \tag{2.35}$$

    - Represents the total spectral energy of the signal.
    - Helps in signal comparison and classification.

- **Number of Negative Eigenvalues** N_h
  - Indicates the **complexity** of the signal.
  - A higher N_h means the signal contains more oscillatory components.
- **Weighted Eigenvalues Moments** We compute higher-order moments of the eigenvalues:

$$M_k = \sum_{n=1}^{N_h} |\lambda_n|^k \tag{2.36}$$

  - $k = 1$ represents the total spectral energy.
  - $k = 2$ describes the spectral spread (how distributed the eigenvalues are).
- **Eigenfunction Norms**
  - The L_2-norm of eigenfunctions reflects how much an eigenfunction contributes to the signal:

$$\|\psi_n(x)\|^2 = \int |\psi_n(x)|^2 dx \tag{2.37}$$

  - Helps in identifying dominant signal components.
- **Spectral Entropy**

$$H = -\sum_{n=1}^{N_h} p_n \log p_n, \quad \text{where } p_n = \frac{|\lambda_n|}{\sum_{m=1}^{N_h} |\lambda_m|} \tag{2.38}$$

  - Measures how spread out the eigenvalues are.
  - **Higher entropy** indicates more randomness in the signal.
  - **Lower entropy** suggests a more structured signal.
- **Invariants for eigenvalues**

  These invariants are spectral moments derived from the negative eigenvalues $\lambda_{i,j} = -\kappa_{i,j}^2$ of the semi-classical Schrödinger operator used in the SCSA method.

$$\mathrm{INV}_1 = \frac{1}{N} \sum_{i=1}^{N} \left(4h \sum_j \kappa_{i,j} \right) \tag{2.39}$$

$$\mathrm{INV}_2 = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{16h}{3} \sum_j \kappa_{i,j}^3 \right) \tag{2.40}$$

$$\mathrm{INV}_3 = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{256h}{7} \sum_j \kappa_{i,j}^7 \right) \tag{2.41}$$

  - $\mathbf{INV_1}$ captures the overall spectral energy.
  - $\mathbf{INV_2}$ is sensitive to nonlinearities and sharper features.
  - $\mathbf{INV_3}$ accentuates fine-scale peaks and detailed structures.

- **Invariants for eigenfunctions**

  These invariants are derived from the eigenfunctions $\psi_{b,i,j}$ of the semi-classical Schrödinger operator.

$$\mathrm{INV}_1 = \frac{1}{B}\sum_{b=1}^{B}\left(\frac{1}{M}\sum_{i=1}^{M}\left(4h\sum_{j=1}^{N}\psi_{b,i,j}\right)\right) \tag{2.42}$$

$$\mathrm{INV}_2 = \frac{1}{B}\sum_{b=1}^{B}\left(\frac{1}{M}\sum_{i=1}^{M}\left(\frac{16h}{3}\sum_{j=1}^{N}\psi_{b,i,j}^{3}\right)\right) \tag{2.43}$$

$$\mathrm{INV}_3 = \frac{1}{B}\sum_{b=1}^{B}\left(\frac{1}{M}\sum_{i=1}^{M}\left(\frac{256h}{7}\sum_{j=1}^{N}\psi_{b,i,j}^{7}\right)\right) \tag{2.44}$$

  - $\mathbf{INV_1}$ reflects the general spatial content of the eigenfunctions.
  - $\mathbf{INV_2}$ captures more pronounced nonlinear features and sharp transitions.
  - $\mathbf{INV_3}$ focuses on fine-scale and detailed structure in the eigenfunctions.

- **Skewness :**

  - For the eigenvalues :

$$\mathrm{Skew} = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{1}{M}\sum_{j=1}^{M}\left(\frac{\kappa_{i,j}-\bar{\kappa}_i}{s_i}\right)^3\right] \tag{2.45}$$

  - For the eigenfunctions :

$$\mathrm{Skew} = \frac{1}{A}\sum_{k=1}^{A}\left[\frac{1}{B}\sum_{i=1}^{B}\left(\frac{1}{C}\sum_{j=1}^{C}\left(\frac{\psi_{k,i,j}-\bar{\psi}_{k,i}}{s_{k,i}}\right)^3\right)\right] \tag{2.46}$$

- **Kurtosis :**

  - For the eigenvalues :

$$\mathrm{Kurtosis} = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{1}{M}\sum_{j=1}^{M}\left(\frac{\kappa_{i,j}-\bar{\kappa}_i}{s_i}\right)^4\right] \tag{2.47}$$

  - For the eigenfunctions:

$$\mathrm{Kurtosis} = \frac{1}{A}\sum_{k=1}^{A}\left[\frac{1}{B}\sum_{i=1}^{B}\left(\frac{1}{C}\sum_{j=1}^{C}\left(\frac{\psi_{k,i,j}-\bar{\psi}_{k,i}}{s_{k,i}}\right)^4\right)\right] \tag{2.48}$$

SCSA provides unique signal features based on eigenvalues and eigenfunctions. These features can be used for biomedical signal classification, such as EEG-based brain activity analysis and brain-computer interfaces (BCIs).

2.4.2 Moments

2.4.2.1 Basic concepts and background

Moments are scalar quantities used to characterize the shape and intensity distribution of an image. They can be defined for both continuous and discrete images.

1. **Raw Moments**

   For a continuous image with intensity function $f(x, y)$, the raw moment of order $(p + q)$ is defined as:

   $$m_{pq} = \iint x^p y^q f(x, y)\, dx\, dy$$

   For a digital (discrete) image, the raw moments are given by:

   $$m_{pq} = \sum_x \sum_y x^p y^q f(x, y)$$

   where:

   - x, y: pixel coordinates,
   - $f(x, y)$: intensity at pixel (x, y),
   - m_{00}: the total intensity (i.e., area for binary images).

2. **Image Centroid**

   The centroid $(\bar{x}, \bar{y})$ of the image is the center of mass, calculated as:

   $$\bar{x} = \frac{m_{10}}{m_{00}}, \quad \bar{y} = \frac{m_{01}}{m_{00}} \tag{2.49}$$

3. **Central Moments**

   Central moments are invariant to translation and are defined as:

   $$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q f(x, y) \tag{2.50}$$

4. **Normalized Central Moments**

   To achieve scale invariance, the central moments are normalized as follows:

   $$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}}, \quad \text{with } \gamma = \frac{p + q}{2} + 1 \tag{2.51}$$

These normalized moments η_{pq} form the basis for constructing Hu's invariant moments, which are designed to be invariant under translation, scaling, and rotation.

2.4.2.2 Hu's moments

Hu (1962) introduced seven moment invariants derived from the normalized central moments η_{pq}, which are invariant under image translation, scaling, and rotation as as Huang and Leng explain in their paper[24]. These invariants are defined as follows:

$$\phi_1 = \eta_{20} + \eta_{02} \tag{2.52}$$

$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \tag{2.53}$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \tag{2.54}$$

$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \tag{2.55}$$

$$\begin{aligned}\phi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12}) \left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] \\ +(3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) \left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right]\end{aligned} \tag{2.56}$$

$$\begin{aligned}\phi_6 = (\eta_{20} - \eta_{02}) \left[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right] \\ +4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})\end{aligned} \tag{2.57}$$

$$\begin{aligned}\phi_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12}) \left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] \\ -(\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03}) \left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right]\end{aligned} \tag{2.58}$$

From (Rao et al.,2013)[43], we see that they calculate the Hu's moment for Fig 2.11, and they gave the results which we are showing in the table 2.2

Indices of invariants	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Hu's seven moment	1.0441	0.61145	0.48121	0.71147	1.0638	1.0399	0.012835

Table 2.2: Hu's seven moment invariants for the Telugu Alphabet,(Table 3 from Rao et al., 2013)[43]

Simple Python code for Hu's moments

```
import pyfeats
import cv2
img = cv2.imread('image.png', cv2.IMREAD_GRAYSCALE)
features_hu, _=pyfeats.hu_moments(img)
```

2.4.2.3 Zernike's moments

From (Rao et al.,2013)[43],Zernike's moments are computed using a set of orthogonal polynomials defined over the unit disc($x^2+y^2 <= 1$). They are known for being rotation-invariant and robust to noise, with minimal redundancy in shape representation.

1. **Zernike Polynomials**

   Zernike polynomials $V_n^m(\rho, \theta)$ are a set of complex polynomials orthogonal on the unit disk $\rho \leq 1$, defined in polar coordinates (ρ, θ) by:

   $$V_{nm}(\rho, \theta) = R_{nm}(\rho)\, e^{jm\theta} \tag{2.59}$$

   where:

- $n \in \mathbb{N}$, $m \in \mathbb{Z}$, $|m| \leq n$,
- $n - |m|$ is even,
- $j = \sqrt{-1}$,
- $\rho \in [0, 1]$, $\theta \in [0, 2\pi]$,
- $R_{nm}(\rho)$ is the radial polynomial.

2. **Radial Polynomial**

   The radial polynomial $R_{nm}(\rho)$ is defined as:

$$R_{nm}(\rho) = \sum_{s=0}^{\frac{n-|m|}{2}} (-1)^s \frac{(n-s)!}{s!\left(\frac{n+|m|}{2} - s\right)!\left(\frac{n-|m|}{2} - s\right)!} \rho^{n-2s} \tag{2.60}$$

3. **Zernike's Moments**

   For a digital image $f(x, y)$ defined over a finite set of pixels inside the unit disk, the Zernike moments are approximated as:

$$A_{nm} \approx \frac{n+1}{\pi} \sum_{x} \sum_{y} f(x, y)\, V_{nm}^{*}(\rho, \theta) \tag{2.61}$$

   where:

   - $D = \{(x, y) \mid x^2 + y^2 \leq 1\}$,
   - (ρ, θ) are the polar coordinates of (x, y),
   - Δx and Δy are the pixel resolutions.
   - $V_{n_m}^{*}(\rho, \theta)$ is the complex conjugate of the Zernike polynomial.

To calculate the Zernike's moments, the algorith first binarize the input image, normalize the image to the unit disk (translation and scale normalization), map image coordinates to polar form (ρ, θ),compute Zernike moments using the above formula 2.61.

Fig 2.10 shows the Block diagram of computing Zernike moments from Rao et al.[43]

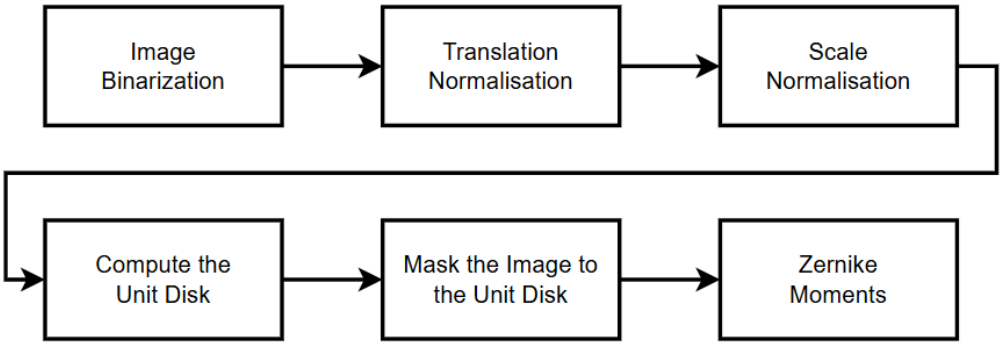


Figure 2.10: Block diagram of computing Zernike moments[43]

.

From Rao et al.[43], the table 2.3 shows the Zernike's moments of Fig 2.11, which is a binary image where the authors extracted the Zernike's moments.

n	m	Zernike Moment values
0	0	28.966
1	1	$0.49656 + 2.5592i$
2	0	-8.6967
2	2	$-5.2529 + 0.22689i$
3	1	$-0.25528 - 5.3013i$
3	3	$0.87071 - 4.3525i$
4	0	-17.969
4	2	$6.9765 - 1.2463i$
4	4	$-5.975 + 0.24356i$
5	1	$-0.51687 + 0.10042i$
5	3	$-1.745 + 7.6192i$
5	5	$0.622 - 0.60109i$

Table 2.3: Zernike Moments for Telugu Alphabet (Table 3 from Rao et al., 2013)[43]

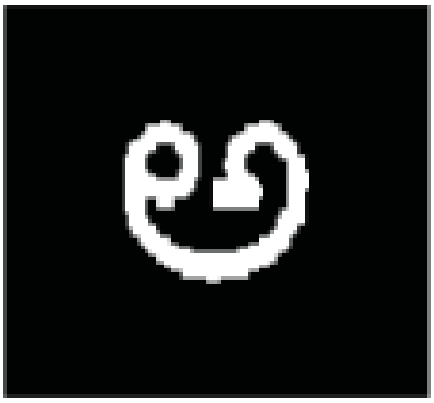


Figure 2.11: TELUGU from Rao et al., 2013 [43]

Simple Python code for Zernike's moments calculation

```
import pyfeats
import cv2
img = cv2.imread('image.png', cv2.IMREAD_GRAYSCALE)
features_z,_= pyfeats.zernikes_moments(img, radius=9)
```

This code will calculate the Zernike's moments for the image.png

2.4.3 Wavelet Packet Transform (WPT)

The *Wavelet Packet Transform* (WPT) is a generalization of the standard discrete wavelet transform (DWT) in which both low-frequency (approximation) and high-frequency (detail) components are recursively decomposed [35]. In the context of 2D image processing,

WPT allows for a more flexible and detailed multiresolution representation of images, where every subband (including horizontal, vertical, and diagonal details) can be further analyzed at finer scales.

Given an image $x(m,n) \in \mathbb{R}^{N\times N}$, the WPT is constructed by recursively applying 2D separable filter banks. At each stage of decomposition, the image (or subimage) is convolved with a pair of 1D analysis filters $h_0(n)$ (lowpass) and $h_1(n)$ (highpass) along both rows and columns as shown in Fig 2.12.

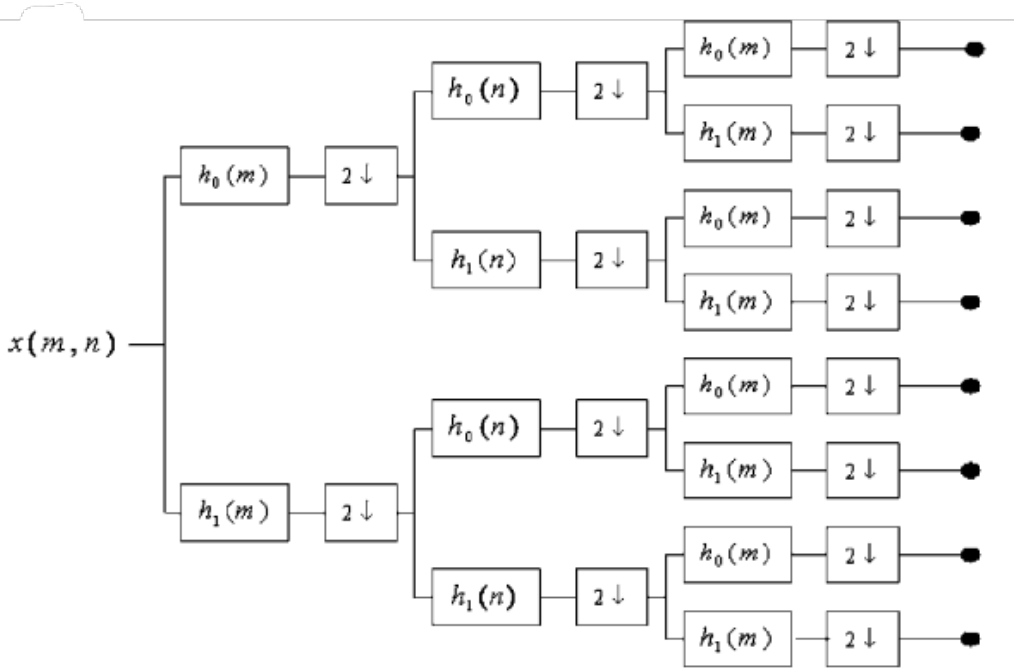


Figure 2.12: Filter bank of wavelet packet transform [30].

This results in four subbands (Fig 2.13):

- LL: low-frequency content in both dimensions (approximation),
- LH: low in rows, high in columns (horizontal details),
- HL: high in rows, low in columns (vertical details),
- HH: high-frequency content in both dimensions (diagonal details).

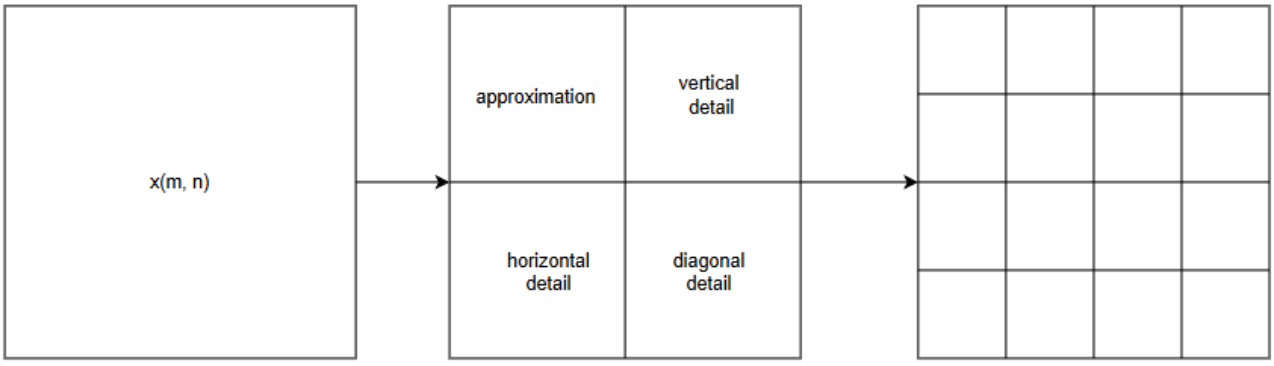


Figure 2.13: Resulting decomposition of wavelet packet transform [30].

In contrast to the standard wavelet transform, where only the LL subband is decomposed

recursively, the WPT recursively applies the same filtering and downsampling procedure to all subbands. The recursive scheme creates a full quad-tree of subbands, Fig 2.13.

Let $x_k^{(j)}(n_1, n_2)$ denote the image block at level j and node k in the wavelet packet tree. Each subband is obtained by:

$$x_{4k}^{(j+1)}(n_1, n_2) = \sum_{m_1} \sum_{m_2} h_0(m_1) h_0(m_2) \cdot x_k^{(j)}(2n_1 - m_1, 2n_2 - m_2) \tag{2.62}$$

$$x_{4k+1}^{(j+1)}(n_1, n_2) = \sum_{m_1} \sum_{m_2} h_0(m_1) h_1(m_2) \cdot x_k^{(j)}(2n_1 - m_1, 2n_2 - m_2) \tag{2.63}$$

$$x_{4k+2}^{(j+1)}(n_1, n_2) = \sum_{m_1} \sum_{m_2} h_1(m_1) h_0(m_2) \cdot x_k^{(j)}(2n_1 - m_1, 2n_2 - m_2) \tag{2.64}$$

$$x_{4k+3}^{(j+1)}(n_1, n_2) = \sum_{m_1} \sum_{m_2} h_1(m_1) h_1(m_2) \cdot x_k^{(j)}(2n_1 - m_1, 2n_2 - m_2) \tag{2.65}$$

Each of the resulting subbands corresponds to a specific spatial frequency band and orientation. This fine-grained decomposition enables enhanced analysis of textural and structural content in images.

Each subimage $x_k^{(j)}(n_1, n_2)$ in the wavelet packet tree contains a block of wavelet packet coefficients. These coefficients represent the correlation of the image with a localized wavelet packet basis function $\psi_k^{(j)}(n_1, n_2)$:

$$c_k^{(j)}(n_1, n_2) = \langle x(n_1, n_2), \psi_k^{(j)}(n_1, n_2) \rangle \tag{2.66}$$

The set of coefficients $\{c_k^{(j)}\}$ provides a tiling of the image in both spatial and frequency domains, offering a complete and adaptive representation of image content [35].

Here we have this code in Python, which is an application of the wavelet packet transform for an image named image.png

```
import pyfeats
import cv2
img = cv2.imread('image.png', cv2.IMREAD_GRAYSCALE)
features_wp, labels_wp=pyfeats.wp_features(img, mask, maxlevel=3)
```

Where:

- **img:** The input image (likely a gray-scale or single-channel image).
- **mask:** A binary mask specifying the region of interest in the image where features should be extracted. Pixels where the mask is True (or 1) are considered for feature extraction.
- **maxlevel=3:** The maximum level of decomposition for the wavelet packet transform. A higher level means more detailed decomposition but increases computational cost.

2.4.4 CNN based feature extraction

CNN is a model that's composed of neurons in the form of filters applied following layers called convolutional layers as defined in [6]

Feature extraction using Convolutional Neural Networks (CNNs) refers to the process of automatically learning hierarchical representations from input data (typically images or time-series) through successive convolutional layers [55].

These learned features capture spatial patterns, edges, textures, and abstract concepts, enabling effective classification, detection, or regression tasks without the need for manual feature engineering.

In image processing, machines perceive images differently than humans do. While we see images as visual scenes (Fig 2.14a), machines interpret them as grids of pixels, where each pixel is represented by a numerical value, as illustrated in Fig 2.14b

(a) What a humans see [6]

0	2	15	0	0	11	10	0	0	0	0	9	9	0	0	0
0	0	0	4	60	157	236	255	255	177	95	61	32	0	0	29
0	10	16	119	238	255	244	245	243	250	249	255	222	103	10	0
0	14	170	255	255	244	254	255	253	245	255	249	253	251	124	1
2	98	255	228	255	251	254	211	141	116	122	215	251	238	255	49
13	217	243	255	155	33	226	52	2	0	10	13	232	255	255	36
16	229	252	254	49	12	0	0	7	7	0	70	237	252	235	62
6	141	245	255	212	25	11	9	3	0	115	236	243	255	137	0
0	87	252	250	248	215	60	0	1	121	252	255	248	144	6	0
0	13	113	255	255	245	255	182	181	248	252	242	208	36	0	19
1	0	5	117	251	255	241	255	247	255	241	162	17	0	7	0
0	0	0	4	58	251	255	246	254	253	255	120	11	0	1	0
0	0	4	97	255	255	255	248	252	255	244	255	182	10	0	4
0	22	206	252	246	251	241	100	24	113	255	245	255	194	9	0
0	111	255	242	255	158	24	0	0	6	39	255	232	230	56	0
0	218	251	250	137	7	11	0	0	0	2	62	255	250	125	3
0	173	255	255	101	9	20	0	13	3	13	182	251	245	61	0
0	107	251	241	255	230	98	55	19	118	217	248	253	255	52	4
0	18	146	250	255	247	255	255	255	249	255	240	255	129	0	5
0	0	23	113	215	255	250	248	255	255	248	248	118	14	12	0
0	0	6	1	0	52	153	233	255	252	147	37	0	0	4	1
0	0	5	5	0	0	0	0	0	14	1	0	6	6	0	0

(b) What machines see [6]

To extract features from images, we typically perform image filtering, where convolution is used to apply filters (or kernels) to the image. The concept of convolution in the context of images is a mathematical operation that involves sliding a small matrix, called a kernel or filter, across the image to produce a transformed output as shown in Fig 2.15.

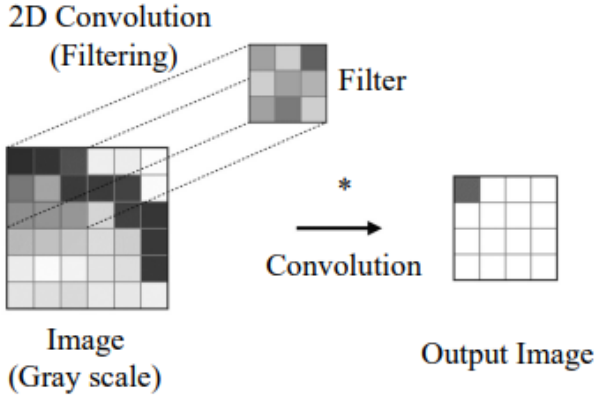


Figure 2.15: 2D convolution[6]

Below are examples of common 3×3 filters used for different tasks such as edge detection, sharpening, and smoothing [6]

1. **Impulse (Neutral) Filter**:

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{2.67}$$

2. **Sharpening Filter**:

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} \tag{2.68}$$

3. **Averaging Filter**:

$$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{2.69}$$

4. **Gaussian Filter**:

$$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \tag{2.70}$$

5. **Edge detection**

$$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix} \quad \text{Or} \quad \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad \text{Or} \quad \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \tag{2.71}$$

An example of applying a 2D convolution on a gray scale image is shown in Fig 2.16 where:

- X is the input 6×6 image,
- K is the applied 3×3 filter with a stride equal to 1,
- Y is the output feature map,
- +1 is the bias added to the convolution

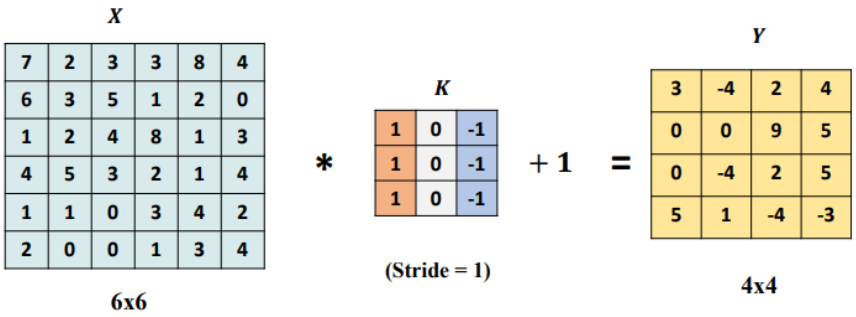


Figure 2.16: 2D convolution applied on a gray scale image[6]

And the mathematical formulation for the convolution is :

$$Y_{i,j} = \left(\sum_{a=0}^{k-1}\sum_{b=0}^{k-1} X_{i+a,j+b} \times K_{a,b}\right) + \text{bias} \tag{2.72}$$

Where $k \times k$ is the size of the filter.

After the filtering, a ReLU activation function is applied(see Fig 2.17), in order to introduce non-linearity into the network.

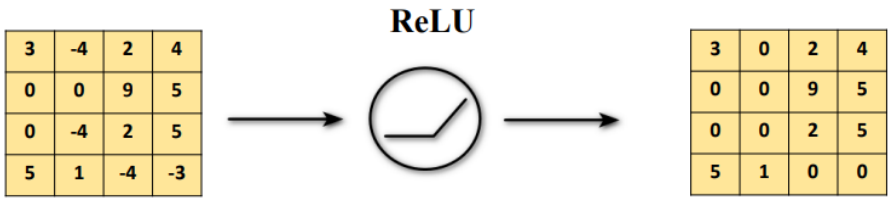


Figure 2.17: ReLU Activation function[6]

Where ReLU is defined as:

$$f(x) = \max(0, x) \tag{2.73}$$

So the output will be :

$$Y_{i,j} = ReLU\left(\left(\sum_{a=0}^{k-1}\sum_{b=0}^{k-1} X_{i+a,j+b} \times K_{a,b}\right) + \text{bias}\right) \tag{2.74}$$

Fig 2.18 illustrate the features extracted using both convolution and activation function.

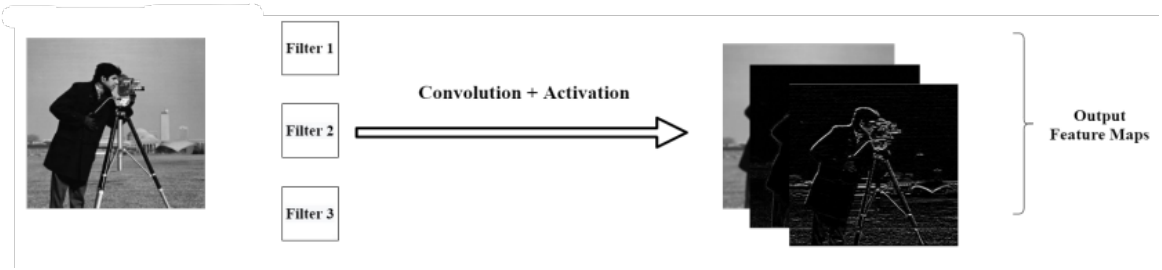


Figure 2.18: Extracted features using convolution + activation function[6]

For the multimodal approach, we utilize Multi Feature Map channels. As demonstrated in Prof. Berrani's course [6], an RGB image consists of three channels, which we treat independently with filters. After applying the filters to each channel, we fuse them and pass them through the activation function.

Fig 2.19 illustrates this concept of multimodal signal fusion. In our case, the red channel corresponds to the image representation of the EEG signal, the blue channel corresponds to the image representation of the ECG signal, and the green channel corresponds to the image representation of the PPG signal. By leveraging this architecture, we are able to perform multimodal neurophysiological signal fusion, enabling us to effectively differentiate between various levels of stress.

We can resume the architectre in Fig 2.19 by this equation:

$$Y_{i,j} = ReLU\left(\left(\sum_{c=0}^{C-1}\sum_{a=0}^{k-1}\sum_{b=0}^{k-1} X_{c,i+a,j+b} \times K_{c,a,b}\right) + \text{bias}\right) \tag{2.75}$$

Where:

- C: Number of channels
- $k \times k$: Filter size

- ReLU: Activation function

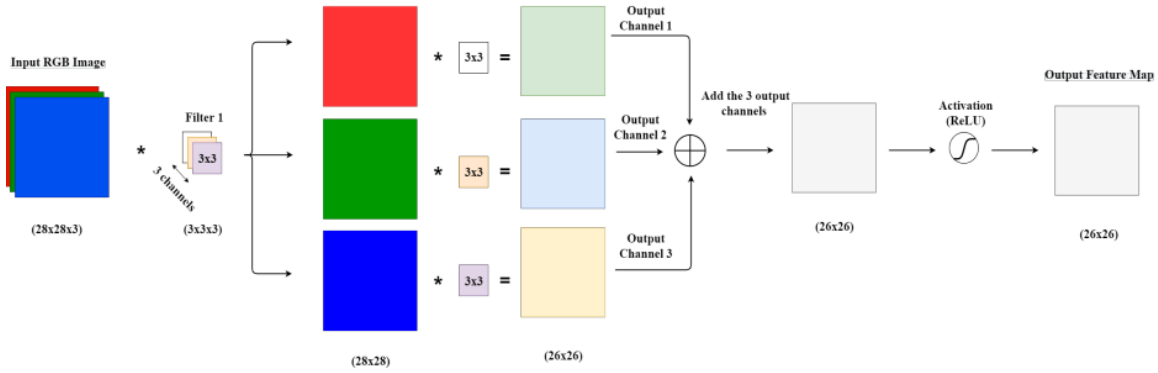


Figure 2.19: Multi Channel Input [6]

After the convolution and activation function layers, the output is passed through a **pooling layer**, also known as **downsampling**, which reduces the dimensionality of the data while preserving the most relevant features.

Fig 2.20 illustrates one of the commonly used pooling methods, namely **Max Pooling**, where a pooling window of size (2×2) is applied to the input. For each window, the output corresponds to the maximum value among all the elements within the selected region.

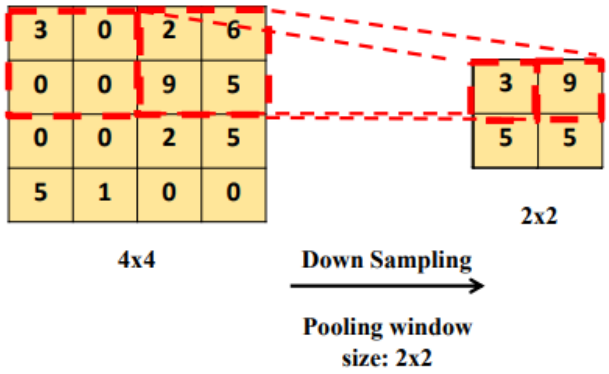


Figure 2.20: MaxPooling [6]

In this case, note that the size of the pooling window (2×2) has nothing to do with the size of the output feature map (here 2×2)

Yann LeCun et al. proposed **LeNet-5** which is a CNN model, Fig 2.21 describes the architecture of it.

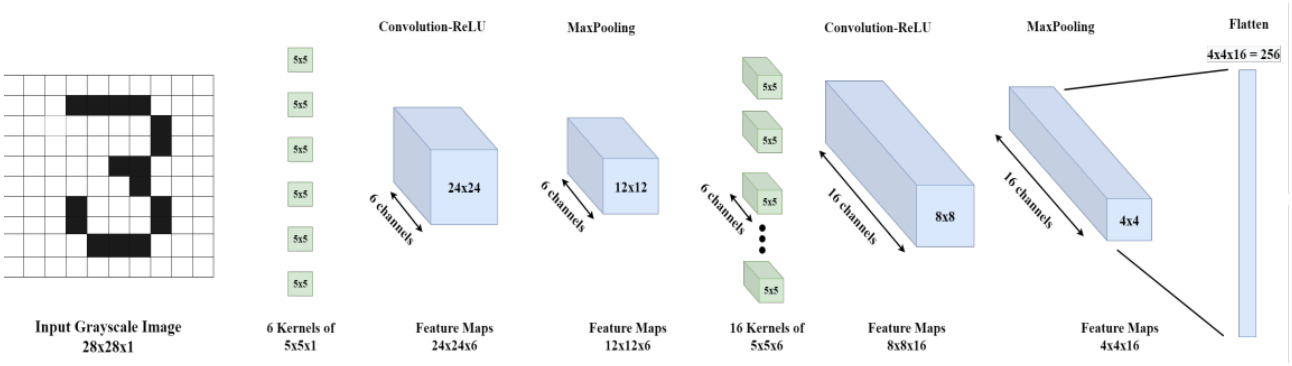


Figure 2.21: Architecture of the LeNet-5 CNN model [6]

2.5 Machine Learning Techniques for classification

From the PASS dataset[31], we selected three physiological signal modalities: EEG, ECG, and PPG. Each signal was segmented according to the experimental protocol, and each segment was assigned a corresponding stress level label derived from the subjective responses to the PASS questionnaire. As our objective is to predict stress levels based on these labeled segments, we employ supervised machine learning techniques.

In this section, we provide an overview of supervised learning, along with a review of related work that has applied these methods to physiological signal classification.

So what is **Supervised machine learning**?

Supervised machine learning refers to the process of learning a function that maps inputs to desired outputs based on a set of labeled training data. It involves supplying algorithms with input-output pairs from which the algorithm learns to generalize to new, unseen data. The primary objective is to create models that can make accurate predictions or classifications for future instances based on learned patterns [3].

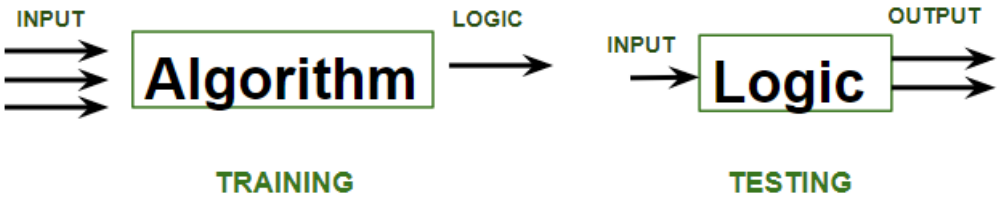


Figure 2.22: Train and test data for ML [18]

In the Fig 2.22, training phase involves feeding the algorithm labeled data, where each data point is paired with its correct output. The algorithm learns to identify patterns and relationships between the input and output data, where testing phase involves feeding the algorithm new, unseen data and evaluating its ability to predict the correct output based on the learned patterns.

According to Akinsola et al.[3], supervised classification is one of the most frequently carried out tasks in intelligent systems. The effectiveness of these algorithms depends heavily on factors such as the size of the dataset, the number of features (attributes), and the tuning of model parameters. Algorithms such as Support Vector Machine (SVM),

Naïve Bayes, Random Forest, Decision Tree, and Neural Networks are commonly used in supervised learning tasks.

1. Decision Tree

A Decision Tree is a model that splits data into subsets using decision rules based on feature values. Each internal node represents a test on a feature, each branch a result of the test, and each leaf node a class label [3, 47].

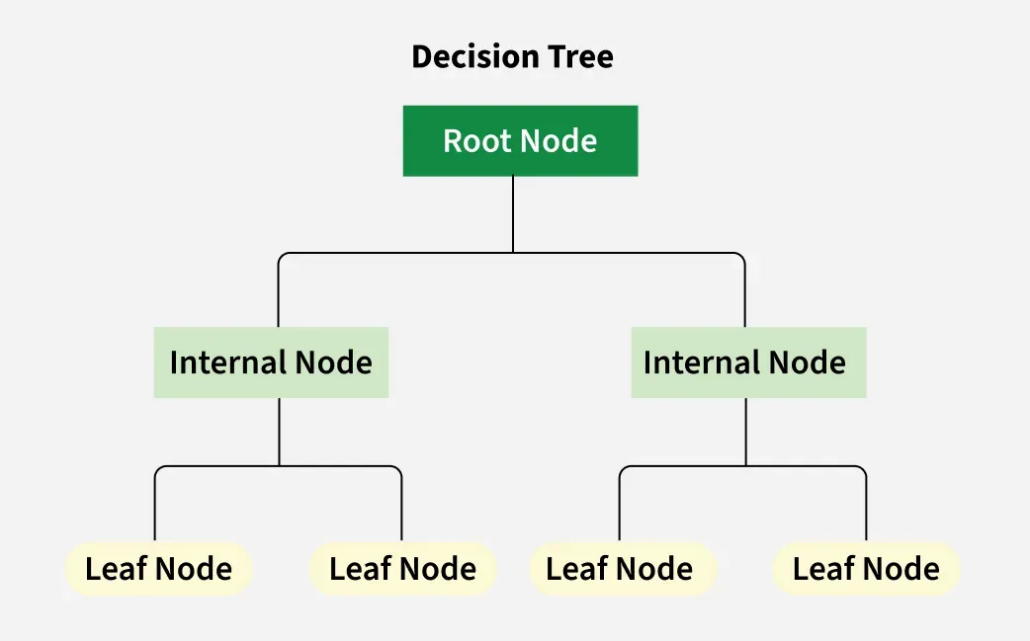


Figure 2.23: Decision Tree Structure [18]

Fig 2.23 shows the decision tree structure where its components are :

- **Root Node**: This is the initial node that represents the complete dataset from which the decision tree begins.
- **Branches**: These are the connections between nodes, indicating the path or flow from one decision to the next.
- **Internal Nodes**: These are decision points within the tree where choices are made based on specific input features.
- **Leaf Nodes**: These are the final nodes at the ends of branches, representing the ultimate outcomes or predictions.

the process in decision tree starts with a main question based on one of these features, this is called the root node.

The tree then asks a series of yes or no questions, each designed to split the data based on a specific feature and a threshold. For instance, the first question might be: “Is feature 12 less than 0.76?” Depending on the answer, the data follows one branch or another.

This process continues, with each new node asking another question about a different feature, dividing the data further and further. Eventually, we reach the end of a branch, known as a leaf node. At this point, the tree makes a final decision.

The goal of the tree is to split the data in such a way that each group becomes as pure as possible, meaning the samples in each group mostly belong to the same class. This step-by-step questioning allows the model to make accurate predictions based on the patterns it finds in the features.

2. Random Forest

Random Forest is an ensemble method that builds multiple decision trees during training and outputs the mode of the classes (classification) or mean prediction (regression) of the individual trees [3, 47].

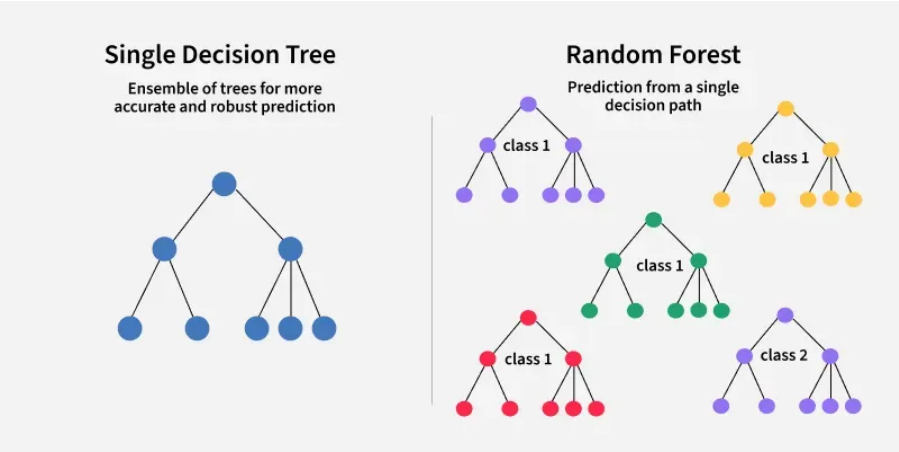


Figure 2.24: Difference between Random Forest and Decision Tree [18]

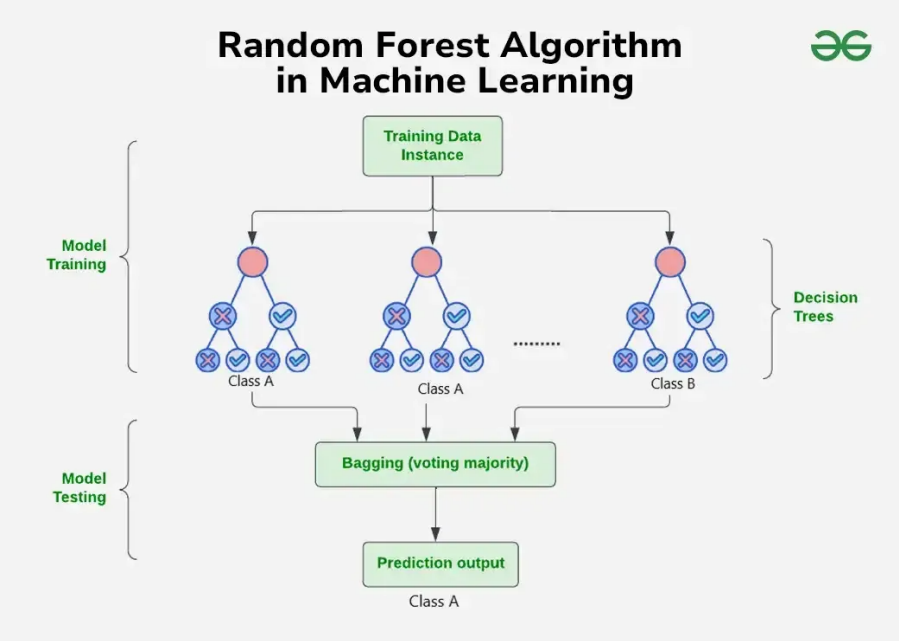


Figure 2.25: Random Forest algorithm[18]

Fig 2.24 shows the difference between a decision tree algorithm and a random forest, where the decision tree algorithm is a single learner, while the random forest algorithm is a collection of decision trees that vote together, making it more accurate and stable.

As we can see in Fig 2.25, the algorithm constructs multiple decision trees, each trained on a randomly selected subset of the data, resulting in slightly different trees. During the

construction of each tree, only a random subset of features is considered at each split, which encourages diversity among the trees and reduces overfitting. Once the trees are built, each one independently makes a prediction based on the data it has seen. For classification tasks, the final prediction is determined through majority voting, where the class chosen by most trees becomes the overall output of the model.

3. Support Vector Machines (SVMs)

SVMs find the optimal hyperplane that separates data into different classes with the maximum margin (see Fig 2.27).

They are effective in high-dimensional spaces and are versatile with different kernel functions [3, 47].

Fig 2.26, shows the hyperplanes that separates different classes, in this case they separate three classes.

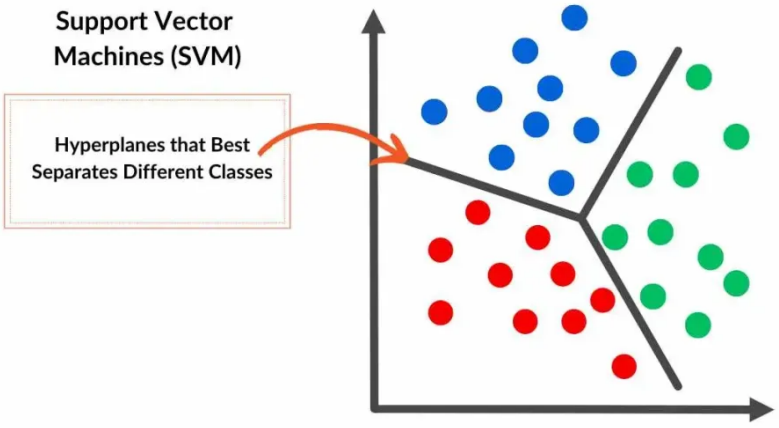


Figure 2.26: Hyperplanes that separate different classes [39]

Fig 2.27, illustrate the separation of classes with the optimal hyperplane maximising the margin.

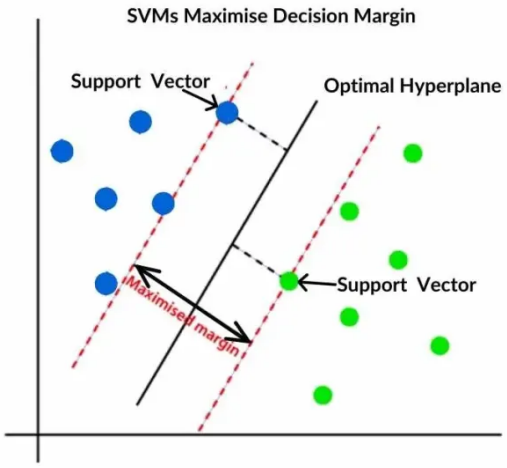


Figure 2.27: Optimal hyperplane and maximised margin [39]

4. Nearest centroid classifier

In machine learning, a nearest centroid classifier or nearest prototype classifier is a classification model that assigns to observations the label of the class of training samples whose mean (centroid) is closest to the observation. [42]

Training phase :
Given labeled training samples

$$\{(\vec{x}_1, y_1), (\vec{x}_2, y_2), (\vec{x}_3, y_3), \ldots, (\vec{x}_n, y_n)\}$$

with class labels $y_i \in \mathbf{Y}$, compute the per-class centroids

$$\vec{\mu}_\ell = \frac{1}{|C_\ell|} \sum_{i \in C_\ell} \vec{x}_i$$

where C_ℓ is the set of indices of samples belonging to class $\ell \in \mathbf{Y}$.

Prediction phase (Test) :

The class assigned to an observation $\vec{x}$ is

$$\hat{y} = \arg\min_{\ell \in \mathbf{Y}} \|\vec{\mu}_\ell - \vec{x}\|.$$

5. Neural Networks

Neural Networks consist of layers of interconnected nodes (neurons) that process data in a way inspired by the human brain. They are capable of learning complex nonlinear mappings and are powerful in handling large and complex datasets [3, 47].

Fig 2.28 shows a basic neural network architecture, with n inputs, n outputs and hidden layers between them.

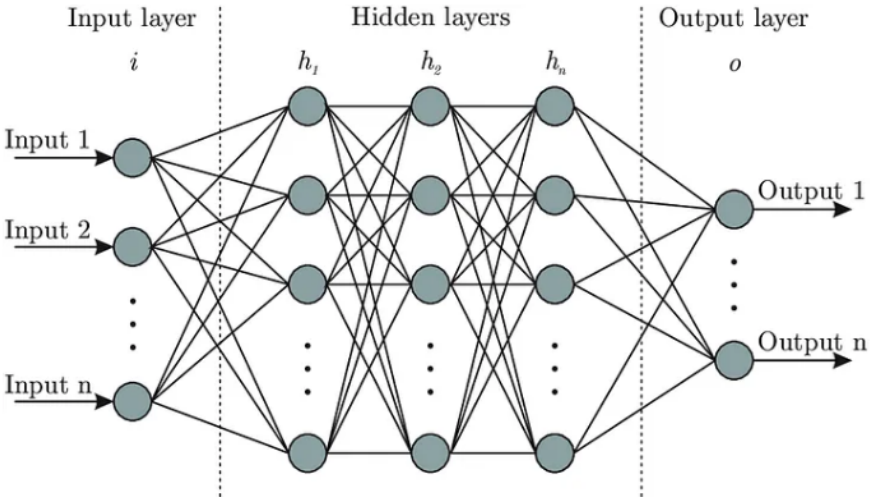


Figure 2.28: Basic neural network architecture [53]

An artificial neuron receives multiple inputs and produces an output using a linear combination followed by a non-linear activation function, Fig 2.29 illustrate the structure of a neuron. It can be mathematically described:

Inputs
Let the inputs to the neuron be:

$$x_0 = 1 \quad \text{(bias input)}, \quad x_1, x_2, \ldots, x_n \tag{2.76}$$

Weights
Each input is associated with a weight:

$$w_0 = b \quad \text{(bias weight)}, \quad w_1, w_2, \ldots, w_n \tag{2.77}$$

Linear Combination
The neuron computes a weighted sum of the inputs:

$$z = \sum_{i=0}^{n} w_i x_i \tag{2.78}$$

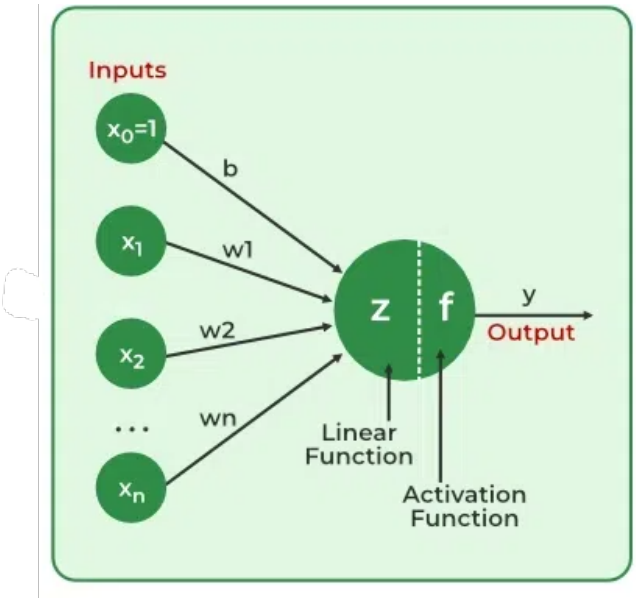


Figure 2.29: artificial neuron [19]

Activation Function
A non-linear activation function $f(\cdot)$ is applied to z:

$$y = f(z) \tag{2.79}$$

Output
The final output of the neuron is:

$$y = f\left(\sum_{i=0}^{n} w_i x_i\right) \tag{2.80}$$

Common Activation Functions
Some commonly used activation functions include:

- Sigmoid: $f(z) = \dfrac{1}{1 + e^{-z}}$
- Hyperbolic tangent: $f(z) = \tanh(z)$
- ReLU (Rectified Linear Unit): $f(z) = \max(0, z)$

2.6 Conclusion

In this chapter, we explore the full processing pipeline of neurophysiological signals (Fig 2.1), begining with noise reduction and artifact removal techniques, such as:

- **Independent Component Analysis (ICA)** and its wavelet-enhanced variant **wICA** for EEG signal purification.
- **Semi-Classical Signal Analysis (SCSA)** for ECG and PPG signals.
- Or, we will use classic filters.

Furthermore, we discussed **signal-to-image representation methods**, including:

- **Visibility Graph (VG)** and **Limited Penetrable Visibility Graph (LPVG)** for transforming time-series signals into graph-based images.
- **Gramian Angular Fields (GAF)** encode temporal correlations into 2D matrices, facilitating the application of deep learning models.

For **feature extraction**, we examine:

- SCSA-based spectral features (eigenvalues, entropy).
- Image-based descriptors (Hu's moments, Zernike moments, and wavelet packet transform).

Finally, we reviewed **supervised machine learning techniques** (e.g., nearest centroid classifier, Random Forests and Neural Networks) and their relevance to stress classification tasks.

The methodologies presented here lay the groundwork for the next chapter, where we will apply these techniques to **multimodal stress classification** for different combinations, leveraging the synergies between EEG, ECG, and PPG signals to improve detection accuracy. By combining advanced signal processing, image-based representations, and machine learning, we aim to develop a robust framework for real-time stress monitoring in dynamic environments.

Chapter 3

Multimodal Stress Classification: Implementation and Experimental Validation

3.1 Introduction

The PASS database, which includes physical activity and multimodal physiological signals, was selected for this study to support the segmentation and processing of neurophysiological data. This choice enables the design and evaluation of a robust stress classification system based on physiological signals, leveraging advanced 1D-to-2D transformation techniques, relevant feature extraction methods, and multiple machine learning classifiers.

The first step involves parsing the data from the database. This includes organizing files by participant, processing markers associated with experimental tasks, and structuring physiological data (PPG, ECG, and EEG) for further use. An initial segmentation is then performed according to the tasks (12 segments per participant), followed by filtering adapted to each type of signal, and a second segmentation specific to 2D representations.

The resulting segments are converted into Visibility Graphs (VG) and Gramian Angular Field (GAF) matrices, allowing one-dimensional signals to be represented in a two-dimensional space while preserving their temporal dynamics and structural patterns.

Two types of descriptor are extracted from these 2D representations:

- Features derived from the SCSA method;
- Descriptors based on Zernike moments, Hu moments, and wavelet packet transform features.

A multiple machine learning classification models are used to exploit these different types of features, enabling a coherent comparison of performance according to the representation-feature combination. The objective is to determine, for each type of signal (EEG, ECG, PPG), the optimal path, i.e., the most effective combination of representation and feature extraction method with a view toward a final multimodal fusion.

This part presents in detail the data parsing and structuring process, the construction of the classification pipeline, the model development, the adopted evaluation protocols, as well as the obtained results and evaluate them.

Fig 3.1 represents the project pipeline from signal acquisition to classification step.

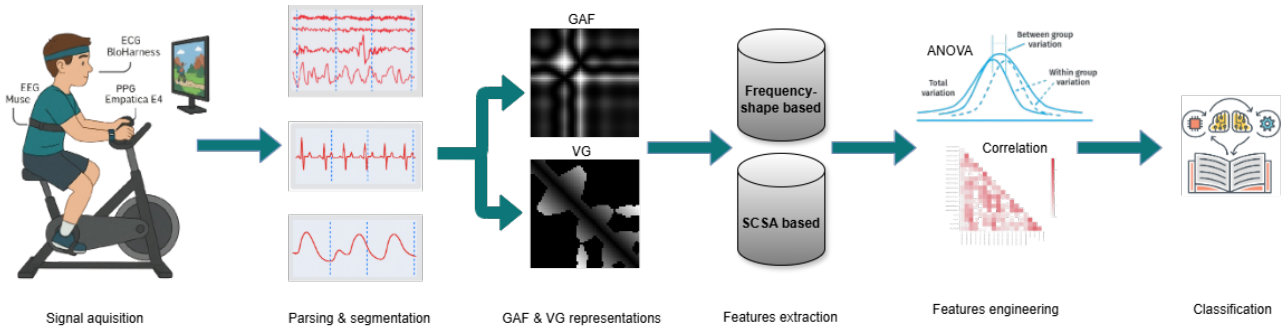


Figure 3.1: Project pipeline

3.2 Presentation of the PASS database

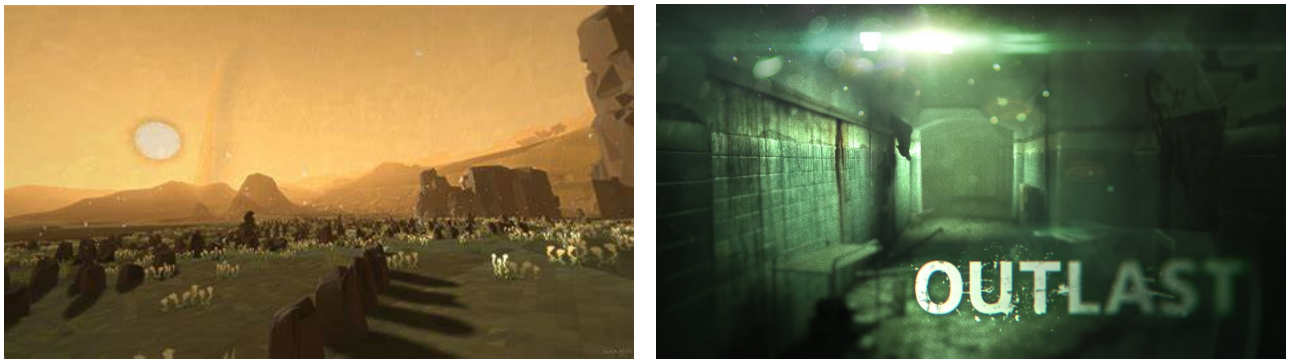
The PASS database (Physical Activity and StresS) is a multimodal dataset designed for research on stress assessment and body/brain-computer interfaces (B/BCIs) in real-world conditions. It was developed to address challenges in automated stress monitoring, particularly in mobile settings where physical activity can interfere with physiological stress markers.

3.2.1 Data Collection Modalities

For the data collection setup, the experiment involved 48 participants, each exposed to six different conditions combining two video games and three levels of physical activity (0, 18, and 24 km/h).

Figure 3.2: Experimental setup from the front (left). Experimental setup from the back (right). BioHarness 3 not shown since worn under the shirt.[41]

The stress-inducing game, Outlast (Fig 3.3b), was selected to create a high-stress condition, while TIMEframe (Fig 3.3a) was used as a non-stressful control game. Each participant played both games under all three physical activity levels, ensuring that stress responses could be analyzed across different movement intensities. This design allowed researchers to investigate the interaction between stress and physical activity on physiological signals.
Fig 3.3 shows both video games.



(a) TIMEframe video game [16] (b) OUTLAST video game [54]

Figure 3.3: video games TIMEframe and OUTLAST

Following the diagram in Fig3.4, which explains the experimental sequence of the entire experiment.

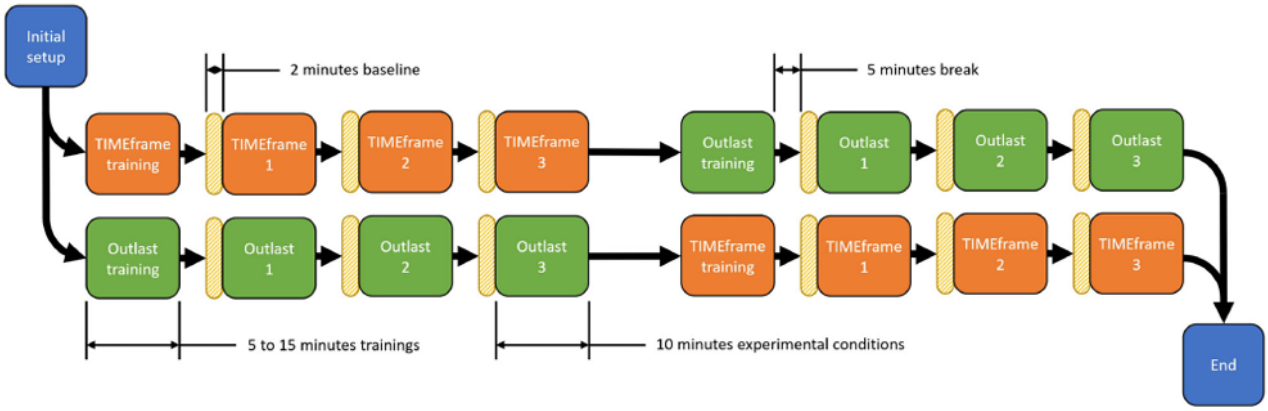


Figure 3.4: Diagram showing the experimental sequence.[41]

Using three sensors for data acquisition:

1. **The Muse Headband**(Fig 3.5), developed by Interaxon (Canada), is a compact and wireless **EEG device** designed for real-time brain activity monitoring. It captures EEG signals with a sampling rate of 220 Hz, which is sufficient for detailed brainwave analysis.

   It is composed of several essential components:

   EEG Sensors (four dry electrodes) the emplacement is as shown in Fig 3.6

   - **AF7 & AF8** (Frontal Lobe - Prefrontal Cortex).
   - **TP9 & TP10** (Temporal Lobe - Near the Ears).
   - **Reference Electrode:** Fpz (center of forehead).

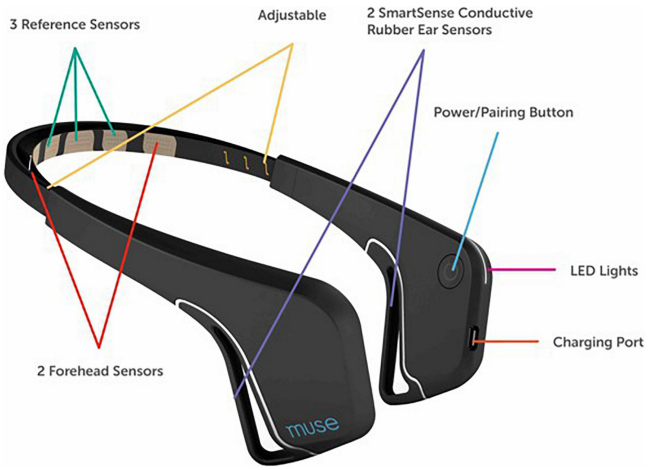


Figure 3.5: the Muse headband [50]

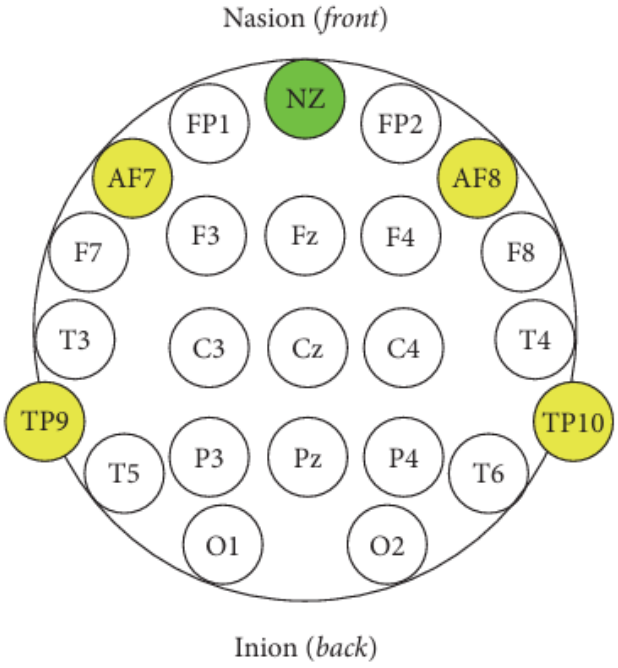


Figure 3.6: EEG sensors TP9, AF7, AF8, and TP10 of the Muse headband on the international standard EEG placement system [7]

Wireless Transmission: Bluetooth Low Energy (BLE) for real-time data transfer.

Battery: Rechargeable lithium-ion battery, lasting up to **10 hours** per charge.

Charging Port: Micro-USB.

2. **The BioHarness 3 (Zephyr/Honeywell) Fig 3.7** is a wearable physiological monitoring device designed to be worn on the chest. It integrates an embedded sensor module and a flexible strap with conductive pads for signal acquisition. The device supports wireless data transmission via Bluetooth and is compatible with various data collection software.

It measures multiple physiological parameters, including: [23]
- Heart rate: 0-240 beats per minute (BPM)
- Breath rate: 0-120 breaths per minute (BPM)
- Posture: 0-180 degrees
- Activity level: Measured in vector magnitude units
- Skin temperature: 33-41°C
- Electrocardiogram (ECG): Sampled at 250 Hz

The BioHarness 3 features onboard data storage for offline logging and real-time transmission via Bluetooth with an approximate range of 10 meters. Its rechargeable battery provides up to 24 hours of continuous operation. These specifications are illustrated in Fig 3.8 [23].

Figure 3.7: BioHarness 3 chest-strap [49]

Sensor Module Size	1.85" x 0.46" (28mm x 7mm)
Sensor Module Weight	0.635oz (18g)
Strap Fabric	Washable conductive smart fabric
Operating Temperature	-22° to 140° F (-30° to 60° C)
Humidity	0% to 95% relative humidity (non-condensing)
Battery	4.2V Li-Ion rechargeable
Run Time	12 to 28 hours
Charging Time	3 hours
IP Rating	IP-55
Wireless	Up to 2 miles (refer to RAELink3 datasheet for complete details)
GPS Accuracy (via RAELink3)	Within 5 meters

Figure 3.8: the BioHarness 3 specifications [23]

3. **The E4 wristband (Empatica)** is a wearable sensor designed like a smartwatch, equipped with flash memory and Bluetooth for data transmission to a computer. It features four sensors that capture different physiological signals: skin temperature (4 Hz), galvanic skin response (4 Hz), blood volume pulse (PPG) (64 Hz), and acceleration (32 Hz). These specifications are illustrated in Fig 3.9.

Figure 3.9: the E4 wristband [60]

For the software, the MuSAE Lab EEG Server (MuLES), a LabVIEW-based platform, was utilized for real-time recording of EEG and other physiological signals, ensuring synchronized data acquisition.
Additionally, MATLAB was employed for comprehensive data processing, including signal filtering, feature extraction, and statistical analysis. A custom-made script was specifically developed to handle experiment markers, enabling precise segmentation and analysis of physiological responses to different experimental conditions.

Subjective measures[41]: PASS database also contains subjective data (questionnaires), which were collected from a NASA-TLX and BORG tests, rating from 0 to 21,

- **NASA-TLX (Task Load Index)**: NASA-TLX was used to assess different aspects of workload. The following six dimensions were measured:
  Mental demand, Physical demand, Temporal demand, Performance, Effort, Frustration.
- **In addition**, two extra stress-related questions were included:
  - **How stressful was the task?**
  - **How scary was the task?**
- **BORG Scale (Perceived Physical Effort)**: The BORG Scale was used to measure the perceived physical effort of participants.
  Ratings were collected after each condition and after breaks.

3.2.2 Types of Recorded Signals & Preprocessing of Raw Data

The PASS database is characterized by being multimodal; it contains several types of data or neurophysiological signals, starting from the EEG, ECG, PPG, EDA , etc. In this part, we will explain each type of data separately.

3.2.2.1 EEG signal

EEG measures electrical activity in the brain using electrodes placed on the scalp as shown in Fig 3.10. The Muse headband used in this project records EEG signals at a sampling rate of 220 Hz, capturing brainwave activity across four main electrodes: TP9, AF7, AF8, and TP10, with a reference electrode at Fpz.

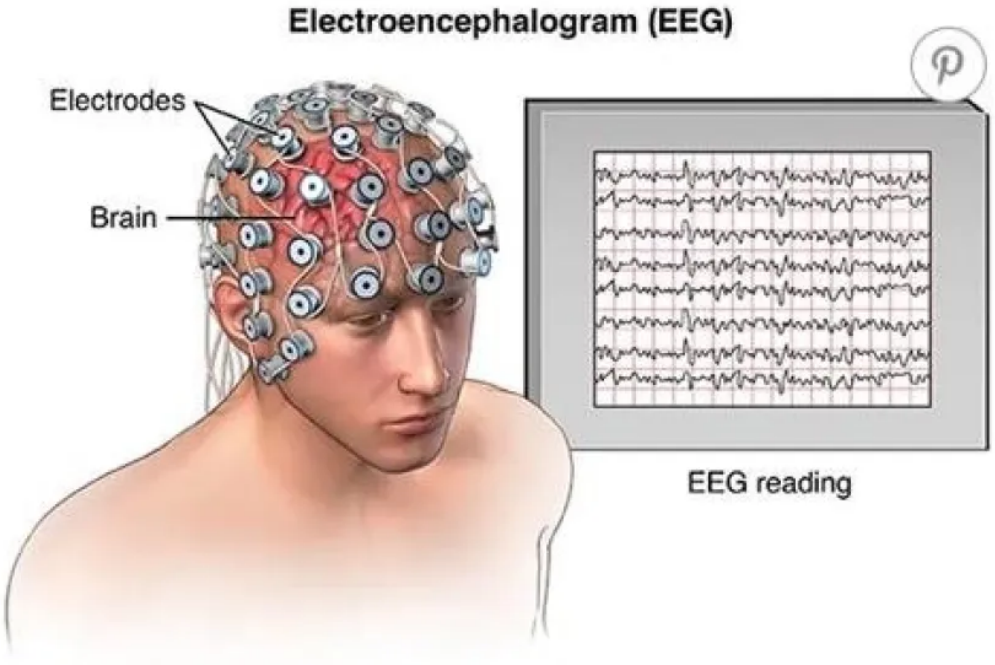


Figure 3.10: Electroencephalogram (EEG) [8]

The EEG signals are transmitted wirelessly via Bluetooth Low Energy (BLE) to a recording system.

EEG Frequency Bands Measured: [1]
The Muse headband records frequency bands including delta, theta, alpha, beta, and gamma.

- **Delta (0.5 – 4 Hz):** the slowest EEG waves, normally detected during the deep and unconscious sleep
- **Theta (4 – 8 Hz):** observed during some states of sleep and quiet focus.
- **Alpha (8 – 14 Hz):** originates during periods of relaxation with eyes closed but still awake.
- **Beta (14 – 30 Hz):** originates during normal consciousness and active concentration.
- **Gamma (over 30 Hz):** are known to have stronger electrical signals in response to visual stimulation.

Fig 3.11 shows the five frequency bands of EEG signal

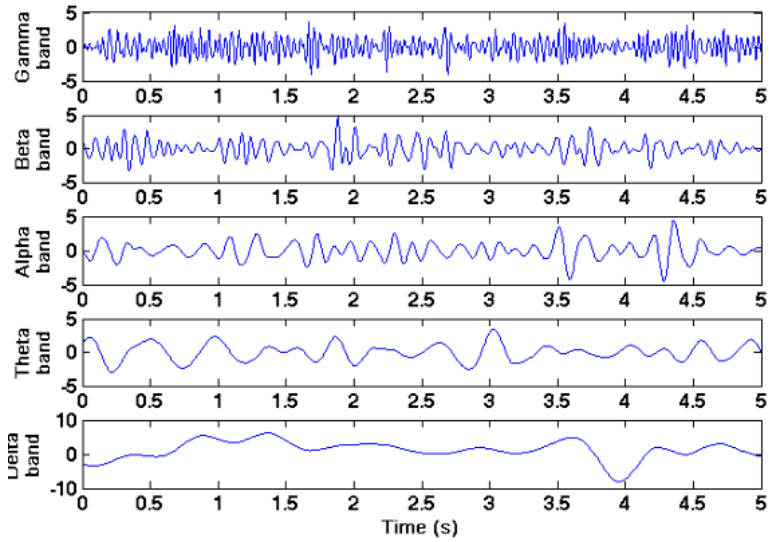


Figure 3.11: The five frequency bands of EEG signal [1]

Noise removal techniques for EEG signals include Independent Component Analysis (ICA) and wavelet decomposition to minimize movement artifacts.

The features extracted were relative to stress:
Absolute power in alpha, theta, and gamma bands, **Prefrontal asymmetry** (AF7–AF8) linked to stress levels, **coherence** in TP9–TP10 for stress evaluation, **amplitude modulation** features such as beta-delta and gamma-delta interactions.

After feature extraction, they were used to train and test machine learning models for stress classification.

3.2.2.2 ECG signal

ECG records the electrical activity of the heart, providing critical insights into heart rate variability (HRV), a key biomarker of stress.

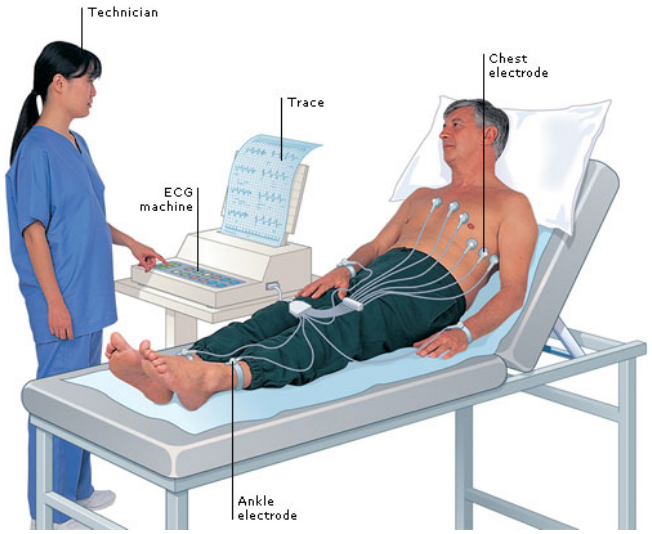
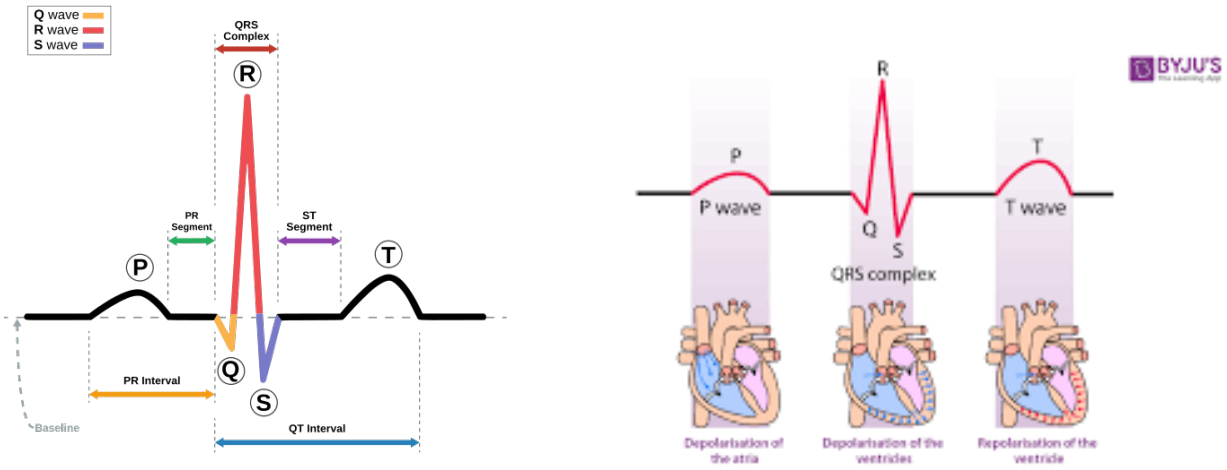


Figure 3.12: ECG Procedure Image [38]

As shown in Fig 3.12, the technician is measuring the electrical activity of the heart for a person using an ECG machine that provides a trace like the theoretical trace in Fig 3.13a.

A typical ECG waveform includes a **P-wave**, **QRS complex**, and **T-wave**, representing the heart's electrical activity during depolarization and repolarization(Fig 3.13b), as illustrated in Figure 3.13a. The **P-wave** reflects atrial depolarization, the **QRS complex** (largest amplitude) corresponds to ventricular depolarization, and the **T-wave** represents ventricular repolarization. Up to **12 leads** can be used to capture ECG signals from different angles. A normal **sinus rhythm** ensures coordinated heart function, while disruptions in this pattern indicate **arrhythmias**, caused by abnormal electrical impulses. ECG is the primary tool for detecting and diagnosing various arrhythmias cite Farag2023.



(a) ECG of a heart in normal sinus rhythm [13]

(b) ECG of a heart in normal sinus rhythm depolarization and repolarization[9]

Figure 3.13: ECG of a heart

In the PASS database, the BioHarness 3 chest strap collects ECG data at a sampling rate of 250 Hz.

After data acquisition, the Pan-Tompkins algorithm (a widely used method for detecting R-peaks in ECG signals) was applied to extract the interbeat interval (IBI) time series.

From ECG data, we extract key features associated with stress, including **heart rate** (HR), **interbeat intervals** (IBI), and the **standard deviation of normal-to-normal intervals** (SDNN) for heart rate variability (HRV) analysis. Additionally, we compute the **low-frequency (LF) to high-frequency (HF) power ratio**, a critical marker for autonomic nervous system activity and stress quantification.

3.2.2.3 Electrodermal activity (EDA)

Electrodermal activity (EDA; sometimes known as galvanic skin response, or GSR) refers to the variation in the electrical conductance of the skin due to sweat secretion. It reflects sweat gland activity, which is controlled by the autonomic nervous system, making it a valuable indicator of stress.

The EDA data was captured using the E4 wristband at 4 Hz using skin electrodes, by applying a low, undetectable, and constant voltage to the skin and then measuring how the skin conductance varies

Applying Filtering using a low-pass Chebyshev filter (1 Hz, 8th order) to remove noise, the data was normalized by computing the average of baselines.

The features extracted from EDA data in this case was the **EDA Level** which measures the overall conductance of the skin due to sweat gland activity, **EDA Responses** which counts the number of short-term electrodermal responses and **the relative Low-Frequency (LF) Power** which represents power in the 0.045–0.15 Hz band, linked to stress response.

3.2.2.4 Breathing rate

Breathing patterns are directly influenced by stress and can be analyzed for physiological monitoring, recorded by the BioHarness 3 at 25 Hz using chest expansion sensors.

Downsampled from 18 Hz to 6 Hz and filtered using a low-pass Chebyshev filter (2 Hz, 8th order) to remove noise.

The features extracted were **breathing Rate**, which is the number of breaths per minute,**breathing Variability**, which quantifies irregularities in breathing using sample entropy, and **the sigh Rate**, which identifies deep breaths exceeding one standard deviation of normal breathing amplitude.

3.2.2.5 Skin temperature

Skin temperature is the temperature measured at the surface of the skin, which can fluctuate due to environmental conditions, physical activity, and autonomic nervous system responses (such as vasoconstriction under stress).

It was recorded using an E4 wristband at a sampling rate of 4 Hz.

Unlike other signals, such as ECG or EDA, no specific filtering or noise reduction was applied to the raw skin temperature data. Since stress affects temperature, the average temperature level was computed, and a delta temperature (difference between the initial and final temperature of a condition) was determined.

The features extracted here are **the temperature**, which is the absolute value of skin temperature, and **the temperature Delta**, which represents the change in temperature over time, often used to assess stress responses.

3.2.2.6 Blood volume pulse (BVP)

PPG measures blood volume changes using optical sensors and is commonly used to assess heart rate and vascular responses.

In this dataset, the optical sensor in the E4 wristband was used to record PPG signals at a sampling rate of 64 Hz.

Blood Volume Pulse (BVP), derived from PPG, serves as an indicator of autonomic nervous system activity.

The feature extracted here was the **minimum BVP**, which corresponds to diastolic pressure, which may be affected by stress. and the **maximum BVP**, which corresponds to systolic pressure, which varies with autonomic nervous system activity.

Finally, the PASS database is publicly available at the MUSAELAB website.[31]

3.3 From raw data to segmented and labelled data

Data preparation constitutes a key step in our work. Its purpose is to parse, structure, segment, and label the neurophysiological signals in a rigorous manner in order to make them exploitable for subsequent processing and classification phases.

3.3.1 Parsing and Participant Selection

Parsing was performed using MATLAB, automating the extraction of signals and markers from the `.mat` files. A hierarchical structure per participant was then established, facilitating processing by signal and task.

The PASS database [31] contains neurophysiological recordings (PPG, ECG, EEG) from 48 participants. However, only a portion of them completed all six intended experimental tasks correctly. Specifically, 23 participants completed all tasks with well-structured metadata. These participants were selected for the main training and testing phase of the machine learning models.

For the remaining 25 participants, the metadata presented irregularities, including missing files, incorrect markers, or incomplete durations. Despite these limitations, we were able to parse, restructure, and process their data in a controlled manner in order to use them for an external validation phase. This strategy allows for evaluating the generalization capability of the models on partially heterogeneous data.

3.3.2 Segmentation Based on Task Markers

Each signal contains 24 markers (12 start and 12 end), corresponding to 6 active tasks interleaved with 6 rest periods (Practice(0)). These conditions are summarized in the following table:

Experimental condition	**Condition start code**	**Condition end code**
Practice (0)	10	20
Timeframe (1)	11	21
Timeframe (2)	12	22
Timeframe (3)	13	23
Outlast (4)	14	24
Outlast (5)	15	25
Outlast (6)	16	26

Table 3.1: Experimental conditions with corresponding start and end codes.

The active phases last on average 10 minutes, while the intermediate rest periods (Practice(0)) last on average 2 minutes. This temporal segmentation enables us to divide each complete recording into 12 distinct segments per participant.

3.3.3 Stress Label Assignment

Stress labels are derived from the `PASS_Subjective_Database.csv` file, which contains subjective responses of participants regarding their perceived stress during each experimental task. These responses are expressed on a scale from 1 to 21.

For the active phases, stress values are directly assigned to the segments without modification.

For the rest phases (Practice(0)), a label of 0 is assigned, indicating the absence of stress during these recovery periods.

The tasks associated with the Timeframe and Outlast games yielded differentiated stress values. Timeframe tasks generally generated low to moderate stress levels, while Outlast tasks, designed to induce higher stress, produced higher stress values.

3.4 The stress classification pipeline

In this section, we detail the stress classification pipeline used for the analysis of neurophysiological signals. This pipeline is structured in several steps, ranging from data pre-processing to machine learning model validation.

3.4.1 Data pre-processing

Before any classification step, the signals must be pre-processed to ensure their quality and compliance. This step includes:

- **Second segmentation :** For each type of signal, we chose the sampling frequency value as the window size for the segments: PPG used a window size of 64 samples, EEG used 220 samples, and ECG used 250 samples.
- **Noise filtering:** A basic filtering is applied (neurokit2 library for Python) to remove artifacts and high-frequency noise. For this, filtering methods suited to each type of signal (PPG and ECG) we used.

PPG Clean Function

```
neurokit2.ppg_clean(ppg_signal, sampling_rate=1000,
heart_rate=None, method="elgendi")
```

This function cleans raw PPG signals using the Elgendi method, which applies a bandpass Butterworth filter (typically 0.5–8 Hz) to remove motion artifacts, baseline drift, and high-frequency noise. When a heart rate is provided, the filter can adapt by centering around it to improve peak detection and pulse analysis.

ECG Clean Function

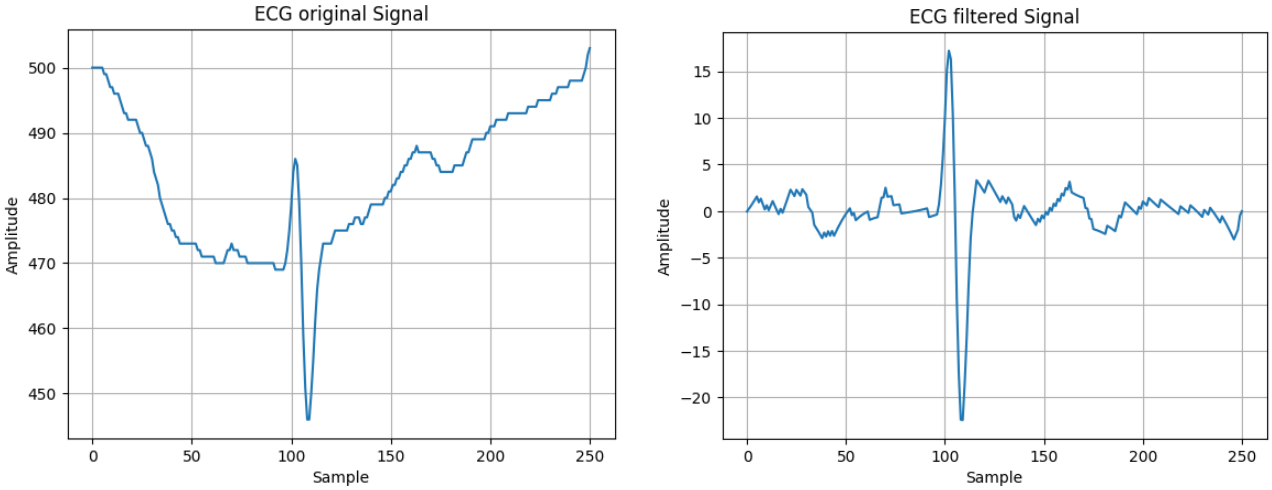
```
neurokit2.ecg_clean(ecg_signal, 250, method="vg")
```

This function cleans ECG signals using the VG (Visibility Graph) method by applying a 4.0 Hz high-pass Butterworth filter (order 2), which effectively removes low-frequency baseline wander. It enhances R-peak detection, making it especially suitable for heart rate variability (HRV) analysis and other ECG-based measurements.

The selection of the 'vg' method was intentional and based on the observation that frequency components below 4 Hz do not contribute significantly to the features relevant in our study.

- **Normalization:** Each signal is normalized to make the data comparable across different participants and trials. Normalization is performed on the values of each signal by adjusting the mean and standard deviation.

Taking an example of an ECG signal (250 samples), illustrated in Fig 3.14a, we filtered it using the `neurokit2` library. The result is shown in Fig 3.14b, where we can observe that the baseline wander and low-frequency noise have been significantly reduced, making the key features of the ECG (such as the QRS complex) more prominent and easier to analyze.



(a) ECG original signal

(b) ECG filtered signal

Figure 3.14: Filtering ECG signal

3.4.2 Signal to Image Representation Using VG and GAF

Once the signals are preprocessed, they are transformed into 2D representations that enhance the structure of the information and allow for the extraction of more relevant features. Two main techniques are used: Visibility Graphs (VG) and Gramian Angular Fields (GAF) image representations.

3.4.2.1 Visibility Graphs (VG) image representation

The visibility graph method transforms a one-dimensional signal into a graph, where each point of the signal becomes a node, and the edges are established between nodes according to a temporal visibility rule.

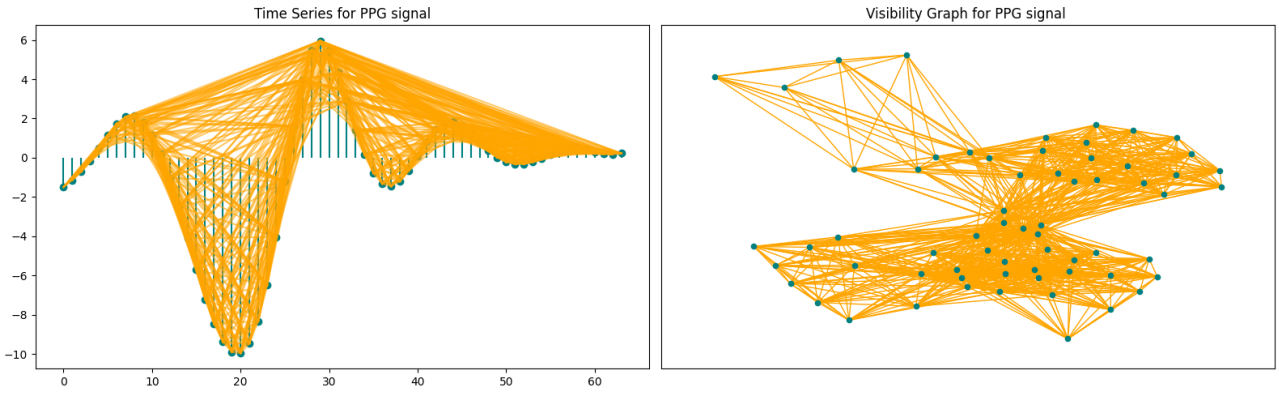


Figure 3.15: Visibility graph of a random selected PPG window

Fig 3.15 shows, on the left, the PPG (photoplethysmographic) signal represented as a time series, with the edges of the visibility graph overlaid in orange. Each point of the signal is treated as a node, and an edge is drawn between two nodes if the line connecting them does not intersect any intermediate point of the signal (the principle of the Natural Visibility Graph (NVG)).
On the right, the associated visibility graph is shown: each node corresponds to a point in the signal, and the connections represent the visibility relationships between them. This representation transforms a time series into a graph structure, enabling topological analysis of the signal that is useful for feature extraction.

From this graph, we generate a weighted adjacency matrix (Fig 3.16). Unlike a classical binary matrix, we use a weighting called '`abs_h_distance`', which represents the absolute horizontal distance between two connected nodes. Thus, each cell (i, j) of the matrix takes the value $|j - i|$ if the nodes i and j are connected. This weighting explicitly encodes the temporal reach of each connection in the original signal, capturing both the topological structure and temporal dynamics of the signal.

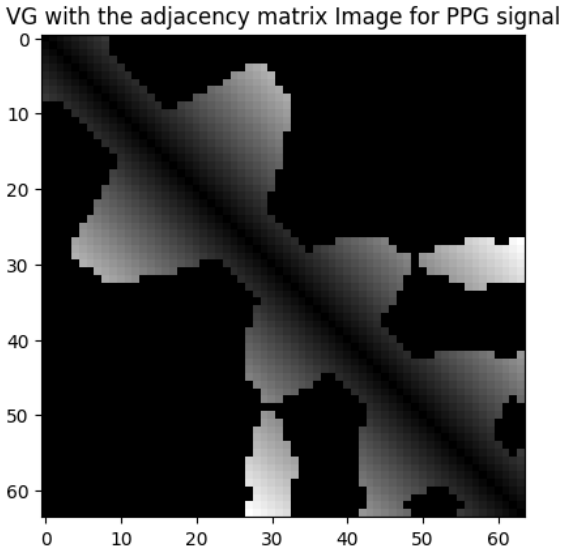


Figure 3.16: Visibility graph image representation of a random selected PPG window

3.4.2.2 Gramian Angular Field (GAF)

In parallel, the signals are also transformed into images using the GAF method (Fig 3.17), which relies on an angular projection of the time series. Each point of the signal

x_i is converted into a trigonometric angle ($arccos(x_i)$), and a matrix is then computed from the cosine summation of these angles, generating a symmetric image that captures temporal correlations.

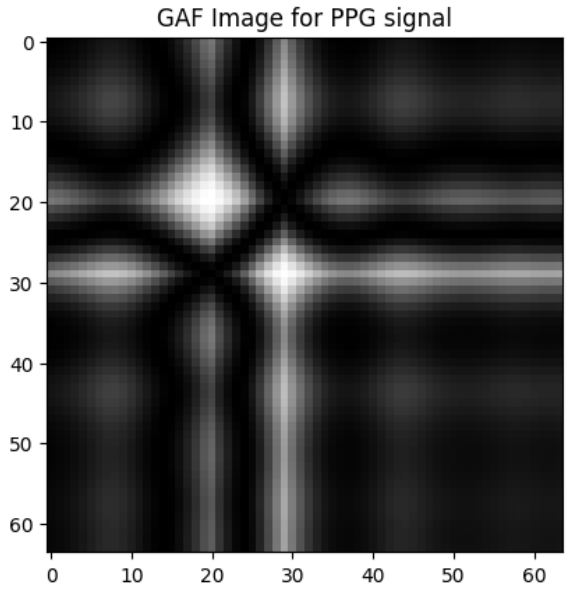


Figure 3.17: Gramian Angular Field of a random selected PPG window

Note that the images generated from both methods were resized to reduce processing time in subsequent steps for ECG and EEG images, originally sized at 250×250 and 220×220 respectively, were downscaled to 100×100 and 64×64. The images derived from the PPG signals, initially at 64×64, were kept at their original size.

These image representations (VG and GAF) are used in the extraction of relevant features for classification afterward.

3.4.3 Feature extraction methods

To compare different feature extraction strategies, two types of approaches have been applied to our transformed representations (VG and GAF). Each method aims to capture specific properties of the signals from their 2D representations.

3.4.3.1 2D SCSA Applied for images

To apply the SCSA method on our VG and GAF representations, we used an indirect 2D approach, which involves applying 1D SCSA on each row and column of the 2D matrix. Specifically, each row (or column) is treated as an independent 1D signal on which the SCSA decomposition is applied. The features obtained for all the rows and columns are then aggregated (e.g., via averaging or concatenation) to construct a final feature vector for each representation.

This method leverages the 2D structure of the image while relying on the robustness of 1D SCSA, providing a balance between spatial expressiveness and computational efficiency.

Given the symmetry of visibility graph (VG) and gramian angular field (GAF) image representations, we limit our computations to the columns only (or rows, as they are equivalent). For each column, we apply the one-dimensional Semi-Classical Signal Analysis (1D

SCSA), which yields the negative eigenvalues κ_{ij} and their associated eigenfunctions ψ_{ij}, where i indexes eigenvalues/eigenfunctions, j indexes columns (signals position) of the image (Fig 3.18 shows how the eigenvalues and eigenfunctions are organized).

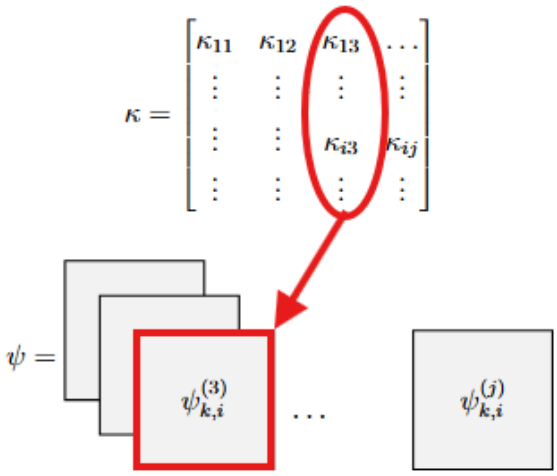


Figure 3.18: Eigenvalues κ and their associated eigenfunctions ψ

Features from Eigenvalues κ : From the eigenvalue matrix $\kappa \in \mathbb{R}^{N\times M}$ (where N is the number of eigenvalues per column and M is the number of the image's columns), we compute the following statistical and spectral features:

Mean:	$\mathbb{E}_j\left[\mathbb{E}_i[\kappa_{ij}]\right]$
Standard Deviation:	$\mathbb{E}_j\left[\text{std}_i(\kappa_{ij})\right]$
First Invariant:	$\mathbb{E}_j\left[4h\sum_i \kappa_{ij}\right]$
Second Invariant:	$\mathbb{E}_j\left[\frac{16h}{3}\sum_i \kappa_{ij}^3\right]$
Third Invariant:	$\mathbb{E}_j\left[\frac{256h}{7}\sum_i \kappa_{ij}^7\right]$
Number of Negative Eigenvalues:	$\mathbb{E}_j[N_y]$
First Eigenvalue Squared:	$(\mathbb{E}_j[\kappa_{1j}])^2$
First Eigenvalue Ratio:	$\frac{\mathbb{E}_j[\kappa_{1j}]}{h}$
First Eigenvalue Median:	$\frac{\text{median}_j(\kappa_{1j})}{h}$
Skewness:	$\mathbb{E}_j[\text{Skew}_i(\kappa_{ij})]$
Kurtosis:	$\mathbb{E}_j[\text{Kurt}_i(\kappa_{ij})]$
Entropy:	$\mathbb{E}_j\left[\text{Entropy}_i(\kappa)\right]$
First-Order Difference:	$\mathbb{E}_j[\Delta_i(\kappa)]$

Features from Eigenfunctions ψ: From the 3D array $\psi \in \mathbb{R}^{N\times L\times M}$, we compute:

$$
\begin{aligned}
\text{Mean:} &\quad \mathbb{E}_j\left[\mathbb{E}_i[\mathbb{E}_k[\psi_{ijk}]]\right] \\
\text{StandardDeviation:} &\quad \mathbb{E}_j\left[\mathbb{E}_i[\mathrm{std}_k(\psi_{ijk})]\right] \\
\text{First Invariant:} &\quad \mathbb{E}_j\left[\mathbb{E}_i\left[4h\sum_k \psi_{ijk}\right]\right] \\
\text{Second Invariant:} &\quad \mathbb{E}_j\left[\mathbb{E}_i\left[\frac{16h}{3}\sum_k \psi_{ijk}^3\right]\right] \\
\text{Third Invariant:} &\quad \mathbb{E}_j\left[\mathbb{E}_i\left[\frac{256h}{7}\sum_k \psi_{ijk}^7\right]\right] \\
\text{First Eigenfunction Squared:} &\quad \left(\mathbb{E}_{j,k}[\psi_{0jk}]\right)^2 \\
\text{First Eigenfunction Ratio:} &\quad \frac{\mathbb{E}_{j,k}[\psi_{0jk}]}{h} \\
\text{First Eigenfunction Median:} &\quad \frac{\mathrm{median}_{j,k}(\psi_{0jk})}{h} \\
\text{Skewness:} &\quad \mathbb{E}_j\left[\mathbb{E}_i\left[\mathrm{Skew}_k(\psi_{ijk})\right]\right] \\
\text{Kurtosis:} &\quad \mathbb{E}_j\left[\mathbb{E}_i\left[\mathrm{Kurt}_k(\psi_{ijk})\right]\right] \\
\text{Entropy:} &\quad \mathbb{E}_j\left[\mathbb{E}_i\left[\mathrm{Entropy}_k(\psi)\right]\right] \\
\text{First-Order Difference:} &\quad \mathbb{E}_j\left[\mathbb{E}_i\left[\Delta_k(\psi)\right]\right]
\end{aligned}
$$

All computed features from eigenvalues and eigenfunctions are concatenated into a final feature vector that compactly characterizes the structural and spectral content of the visibility graph image (or Gramian angular field image) using the SCSA framework.

3.4.3.2 Zernike Moments + Hu Moments + Wavelet Packet Transform

The 2D VG and GAF representations are also analyzed using a combination of classical image descriptors, let $f(x, y)$ denote the 2D VG or GAF representation of a 1D time series signal, where (x, y) indexes the image pixels. The following features are extracted from this image:

- **Zernike Moments:** Capture global shape features using a basis of orthogonal polynomials on the unit disk. First, map (x, y) to polar coordinates (ρ, θ) within the unit circle:

$$Z_{nm} = \frac{n+1}{\pi}\sum_x\sum_y f(x, y)V_{nm}^*(\rho, \theta)$$

  where $V_{nm}(\rho, \theta) = R_{nm}(\rho)e^{jm\theta}$ and R_{nm} is the radial polynomial.

- **Hu Moments:** Describe shape using central moments, invariant to translation, scale, and rotation:

$$m_{pq} = \sum_x\sum_y x^p y^q f(x, y), \quad \mu_{pq} = \sum_x\sum_y (x-\bar{x})^p (y-\bar{y})^q f(x, y)$$

  where $\bar{x} = \frac{m_{10}}{m_{00}}$, $\bar{y} = \frac{m_{01}}{m_{00}}$. From these, 7 Hu invariant moments ϕ_1 to ϕ_7 are computed, such as:

$$\phi_1 = \eta_{20} + \eta_{02}, \quad \phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2$$

  with η_{pq} being normalized central moments.

- **Wavelet Packet Features:** Enable hierarchical frequency analysis. The VG or GAF image is recursively decomposed into frequency subbands using wavelet packet decomposition. At each level j, the image is split into subbands and features (mean and standard deviation) are computed from the coefficients:

$$\text{WP\_coif\_mean} = \frac{1}{N}\sum_{i=1}^{N} c_i \ , \quad \text{WP\_coif\_std} = \frac{1}{N}\sqrt{\sum_{i=1}^{N}(c_i - \frac{1}{N}\sum_{i=1}^{N} c_i)^2} \tag{3.1}$$

  where c_i are the wavelet coefficients.

This combination yields a rich and complementary feature vector, capturing both geometric structure and textural details, the list is in Table 3.2.

Table 3.2: Zernike Moments + Hu Moments + Wavelet Packet Transform features list

Feature Category	**Feature Names**
Zernike Moments (radius=9)	Zernikes_Moments_radius_9_0, Zernikes_Moments_radius_9_1, Zernikes_Moments_radius_9_2, Zernikes_Moments_radius_9_3, Zernikes_Moments_radius_9_4, Zernikes_Moments_radius_9_5, Zernikes_Moments_radius_9_6, Zernikes_Moments_radius_9_7, Zernikes_Moments_radius_9_8, Zernikes_Moments_radius_9_9, Zernikes_Moments_radius_9_10, Zernikes_Moments_radius_9_11, Zernikes_Moments_radius_9_12, Zernikes_Moments_radius_9_13, Zernikes_Moments_radius_9_14, Zernikes_Moments_radius_9_15, Zernikes_Moments_radius_9_16, Zernikes_Moments_radius_9_17, Zernikes_Moments_radius_9_18, Zernikes_Moments_radius_9_19, Zernikes_Moments_radius_9_20, Zernikes_Moments_radius_9_21, Zernikes_Moments_radius_9_22, Zernikes_Moments_radius_9_23, Zernikes_Moments_radius_9_24
Hu Moments	Hu_Moment_0, Hu_Moment_1, Hu_Moment_2, Hu_Moment_3, Hu_Moment_4, Hu_Moment_5, Hu_Moment_6
Wavelet Packet (coif1) Features	WP_coif1_aah_mean, WP_coif1_aah_std, WP_coif1_aav_mean, WP_coif1_aav_std, WP_coif1_aad_mean, WP_coif1_aad_std, WP_coif1_aha_mean, WP_coif1_aha_std, WP_coif1_ahh_mean, WP_coif1_ahh_std, WP_coif1_ahv_mean, WP_coif1_ahv_std, WP_coif1_ahd_mean, WP_coif1_ahd_std, WP_coif1_ava_mean, WP_coif1_ava_std, WP_coif1_avh_mean, WP_coif1_avh_std, WP_coif1_avv_mean, WP_coif1_avv_std, WP_coif1_avd_mean, WP_coif1_avd_std, WP_coif1_ada_mean, WP_coif1_ada_std, WP_coif1_adh_mean, WP_coif1_adh_std, WP_coif1_adv_mean, WP_coif1_adv_std, WP_coif1_add_mean, WP_coif1_add_std, WP_coif1_haa_mean, WP_coif1_haa_std, WP_coif1_hah_mean,

	WP_coif1_hah_std, WP_coif1_hav_mean, WP_coif1_hav_std, WP_coif1_had_mean, WP_coif1_had_std, WP_coif1_hha_mean, WP_coif1_hha_std, WP_coif1_hhh_mean, WP_coif1_hhh_std, WP_coif1_hhv_mean, WP_coif1_hhv_std, WP_coif1_hhd_mean, WP_coif1_hhd_std, WP_coif1_hva_mean, WP_coif1_hva_std, WP_coif1_hvh_mean, WP_coif1_hvh_std, WP_coif1_hvv_mean, WP_coif1_hvv_std, WP_coif1_hvd_mean, WP_coif1_hvd_std, WP_coif1_hda_mean, WP_coif1_hda_std, WP_coif1_hdh_mean, WP_coif1_hdh_std, WP_coif1_hdv_mean, WP_coif1_hdv_std, WP_coif1_hdd_mean, WP_coif1_hdd_std, WP_coif1_vaa_mean, WP_coif1_vaa_std, WP_coif1_vah_mean, WP_coif1_vah_std, WP_coif1_vav_mean, WP_coif1_vav_std, WP_coif1_vad_mean, WP_coif1_vad_std, WP_coif1_vha_mean, WP_coif1_vha_std, WP_coif1_vhh_mean, WP_coif1_vhh_std, WP_coif1_vhv_mean, WP_coif1_vhv_std, WP_coif1_vhd_mean, WP_coif1_vhd_std, WP_coif1_vva_mean, WP_coif1_vva_std, WP_coif1_vvh_mean, WP_coif1_vvh_std, WP_coif1_vvv_mean, WP_coif1_vvv_std, WP_coif1_vvd_mean, WP_coif1_vvd_std, WP_coif1_vda_mean, WP_coif1_vda_std, WP_coif1_vdh_mean, WP_coif1_vdh_std, WP_coif1_vdv_mean, WP_coif1_vdv_std, WP_coif1_vdd_mean, WP_coif1_vdd_std, WP_coif1_daa_mean, WP_coif1_daa_std, WP_coif1_dah_mean, WP_coif1_dah_std, WP_coif1_dav_mean, WP_coif1_dav_std, WP_coif1_dad_mean, WP_coif1_dad_std, WP_coif1_dha_mean, WP_coif1_dha_std, WP_coif1_dhh_mean, WP_coif1_dhh_std, WP_coif1_dhv_mean, WP_coif1_dhv_std, WP_coif1_dhd_mean, WP_coif1_dhd_std, WP_coif1_dva_mean, WP_coif1_dva_std, WP_coif1_dvh_mean, WP_coif1_dvh_std, WP_coif1_dvv_mean, WP_coif1_dvv_std, WP_coif1_dvd_mean, WP_coif1_dvd_std, WP_coif1_dda_mean, WP_coif1_dda_std, WP_coif1_ddh_mean, WP_coif1_ddh_std, WP_coif1_ddv_mean, WP_coif1_ddv_std, WP_coif1_ddd_mean, WP_coif1_ddd_std

3.4.4 Features engineering

At this stage, each signal (PPG, ECG, and the four EEG channels) is processed through two distinct 2D representations: the Visibility Graph (VG) and the Gramian Angular Field (GAF). For each representation, two types of feature vectors are extracted:

- Features from the Semi-Classical Signal Analysis (SCSA) method (29 type of features).
- Concatenated image descriptors from Zernike moments, Hu moments, and Wavelet-

packet transform (159 type of features).

This results in four distinct feature paths per signal:

- **VG_SCSA**: Features from SCSA applied on the VG image.
- **VG_Moments**: Concatenated moment-based features from the VG image.
- **GAF_SCSA**: Features from SCSA applied on the GAF image.
- **GAF_Moments**: Concatenated moment-based features from the GAF image.

Each feature path is independently used to train a classifier. However, prior to classification, a feature selection process is performed to retain only the most relevant features. This process includes:

- **Computing the correlation** between features to eliminate redundancy.
- **Applying the ANOVA test** to assess the statistical significance of each feature.

These steps ensure that only the most informative and discriminative features are used for model training.

3.4.4.1 Correlation Between Features

Correlation between features refers to the statistical relationship or dependency between two variables. In the context of feature selection, it is important to identify and remove highly correlated (redundant) features, as they provide overlapping information and can negatively affect the performance and generalization of the classifier. By computing a correlation matrix (e.g in Fig 3.19 of a correlation matrix), features that exhibit strong linear relationships (e.g., Spearman correlation coefficient close to 1 or -1) can be detected and filtered out, ensuring a more compact and informative feature set.

In our case we choose to use Spearman correlation, with threshold equal to 0.75 (remove one of the features that have a correlation coefficient higher than 0.75), after that, we keep only the uncorrelated features in order to remove redundancy and as a result, we will obtain a reduced number of features.

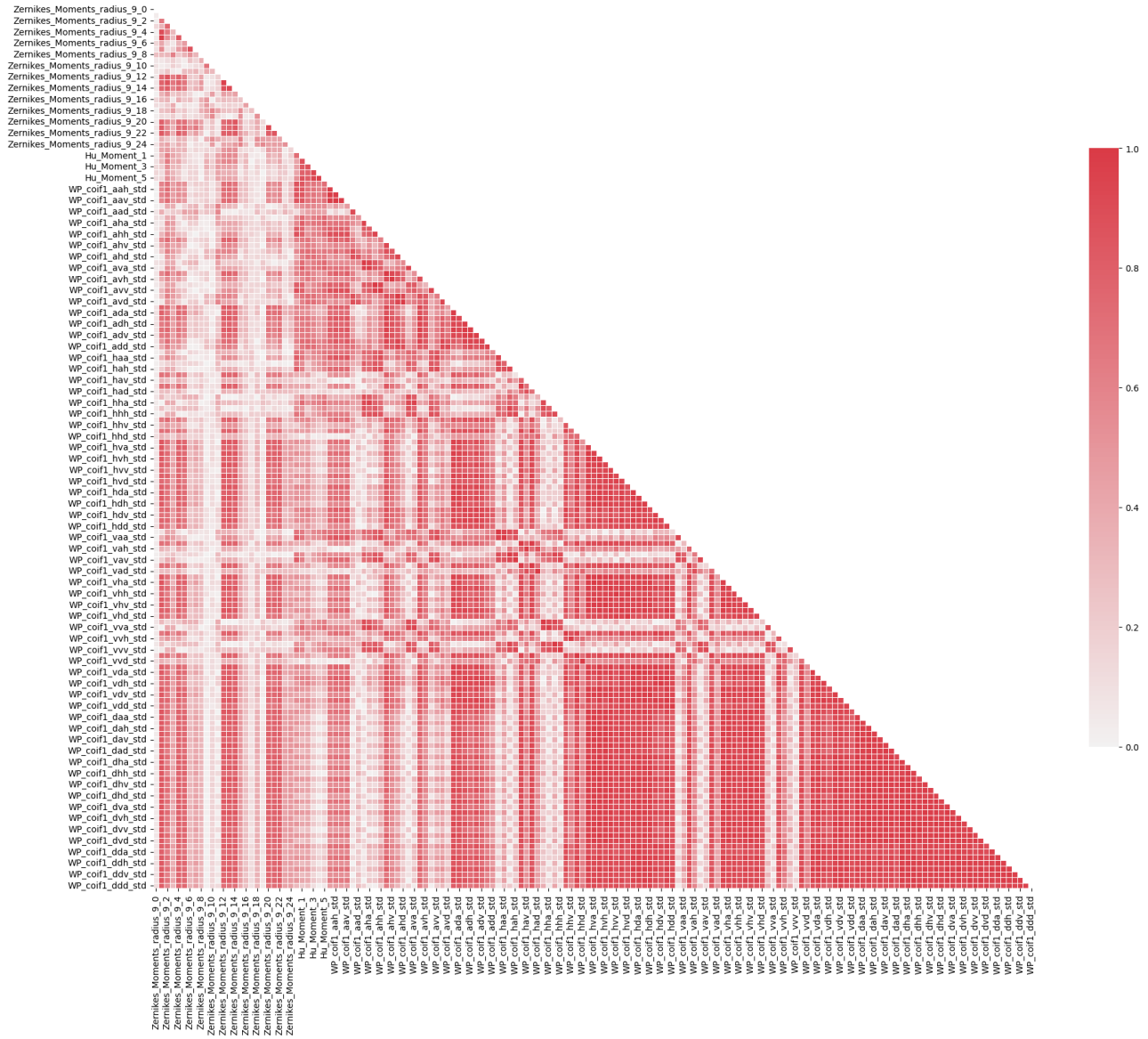


Figure 3.19: Correlation matrix of the Zernike + Hu + wavelet features from VG image for the ECG signal

Fig 3.20 shows the correlation matrix after removing correlated features for **Zernike + Hu + wavelet features from VG image of the ECG signal**, we notice that the correlation matrix get smaller and some features are not existing, due to the first step of features engineering.

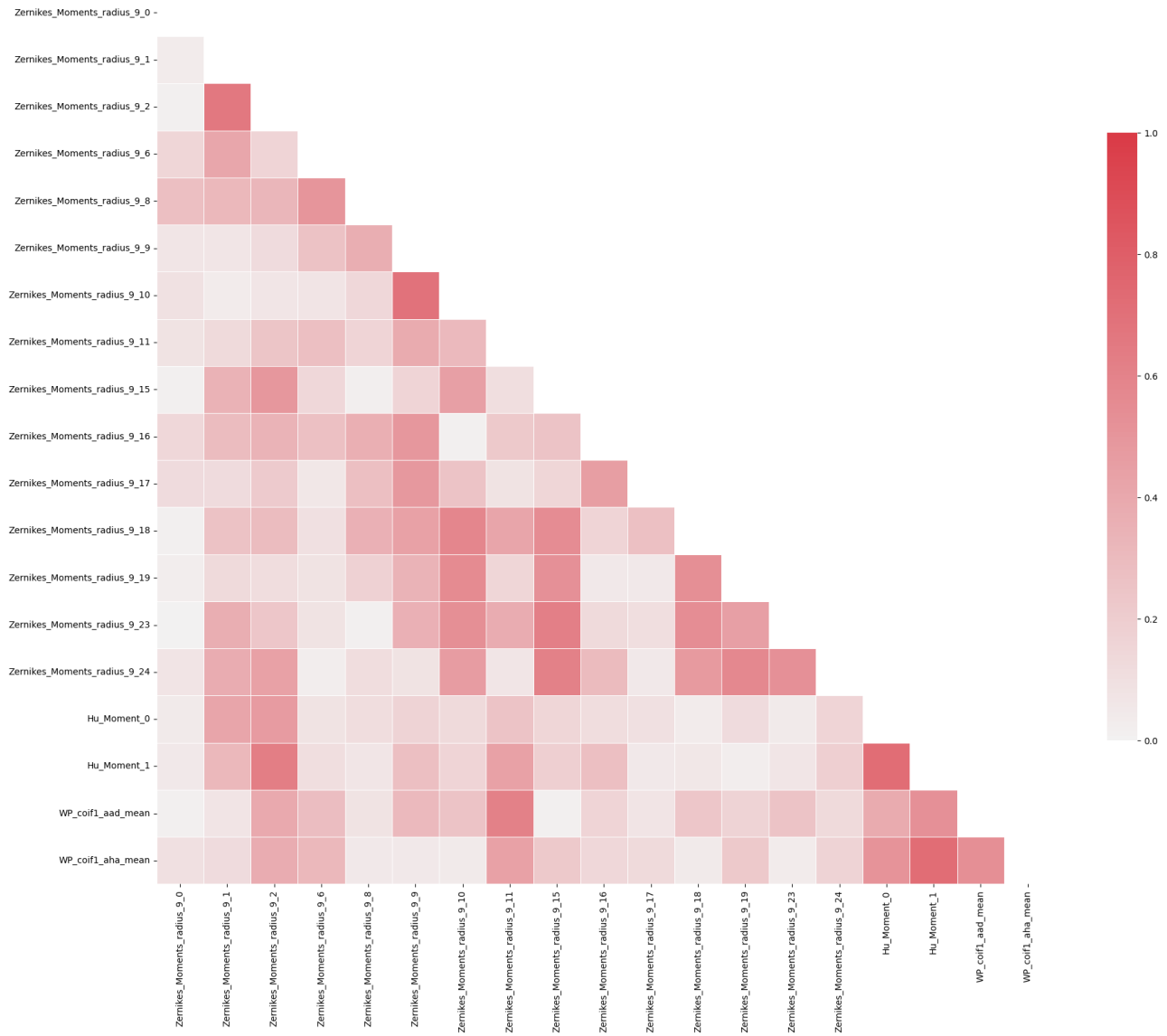


Figure 3.20: Correlation matrix of the Zernike + Hu + wavelet features from VG image for the ECG signal after removing the correlated features

It is important to note that we performed this step for all paths.

3.4.4.2 ANOVA Test

The ANOVA (Analysis of Variance) test is a statistical method used to determine whether there are significant differences between the means of three or more independent groups. In the context of feature selection, the ANOVA test evaluates each feature by measuring how well it separates the classes. A higher ANOVA score indicates that the feature contributes significantly to distinguish between different classes, making it a strong candidate for inclusion in the classification model.
Let X_j denote the j^{th} feature across N total samples, and suppose the dataset is partitioned into C classes. Let n_c be the number of samples in class c. Define:

- $\bar{X}_j$: the overall (global) mean of feature X_j over all samples.
- $\bar{X}_{jc}$: the mean of feature X_j within class c.
- X_{ji}: the value of feature X_j for sample i.

To assess how well feature X_j discriminates between the classes, we compute its **ANOVA F-score**, defined as the ratio of *between-class variance* to *within-class variance*:

$$F_j = \frac{\text{Between-Class Variance}}{\text{Within-Class Variance}} \tag{3.2}$$

$$F_j = \frac{\frac{1}{C-1}\sum_{c=1}^{C} n_c(\bar{X}_{jc} - \bar{X}_j)^2}{\frac{1}{N-C}\sum_{c=1}^{C}\sum_{i\in c}(X_{ji} - \bar{X}_{jc})^2} \tag{3.3}$$

Alternatively, this can be written as:

$$F_j = \frac{MS_B}{MS_W} \tag{3.4}$$

where:

- MS_B is the *between-class mean square* (variation of class means from the global mean),
- MS_W is the *within-class mean square* (variation of samples within each class).

Interpretation:

- A higher F_j indicates that the feature X_j exhibits significantly different means across the classes, implying strong discriminative power.
- A lower F_j suggests that the feature is not helpful for class separation.

Following the initial feature selection step using a correlation matrix, we removed redundancy by eliminating one of each pair of features with a correlation coefficient greater than or equal to 0.75. This ensured that the remaining features were sufficiently uncorrelated, minimizing multicollinearity and preserving interpretability.

In the second step, we applied an **ANOVA (Analysis of Variance)** test to the reduced feature set. This test assesses the statistical significance of each feature in differentiating between the predefined stress classes. From the set of uncorrelated features, we selected the top k features, specifically $k = 10$ before fusion, based on their F-scores, identifying those most relevant for distinguishing between stress levels (As shown in Table 3.3, only the top three features out of the ten most significant ones are presented for illustration).

This two-step approach balances the removal of redundancy (via correlation analysis) with the retention of statistically informative features (via ANOVA), thereby optimizing the quality of the feature set used for model training.

Table 3.3: The three first significant Features after ANOVA test

Feature	**F-value**
Zernikes_Moments_radius_9_15	4.030560
Zernikes_Moments_radius_9_18	3.985633
Zernikes_Moments_radius_9_2	3.155824

The procedure works as follows:

1. Compute the F-statistic F_j for each feature X_j using the ANOVA formulation above.
2. Rank all features based on their F-scores.
3. Select the top k features with the highest scores.

Table 3.4: Comparison of Random Forest and Extra Trees Classifiers

File	SAC	SAA	RF Accuracy	ET Accuracy
ECG Features				
VG_ECG_P1	(138, 24)	(138, 10)	0.4524	0.4524
VG_ECG_SCSA	(138, 2)	(138, 2)	0.5238	0.5238
GAF_ECG_P1	(138, 10)	(138, 10)	0.5238	0.5238
GAF_ECG_SCSA	(138, 2)	(138, 2)	0.5238	0.5476
PPG Features				
VG_PPG_P1	(138, 10)	(138, 10)	0.5000	0.5000
VG_PPG_SCSA	(138, 2)	(138, 2)	0.5714	0.4524
GAF_PPG_P1	(138, 9)	(138, 9)	0.5000	0.5000
GAF_PPG_SCSA	(138, 4)	(138, 4)	0.5714	0.5000
EEG Features				
VG_EEG_C1_P1	(138, 7)	(138, 7)	0.4762	0.5476
VG_EEG_C1_SCSA	(138, 3)	(138, 3)	0.5000	0.4762
GAF_EEG_C1_P1	(138, 7)	(138, 7)	0.5476	0.5238
GAF_EEG_C1_SCSA	(138, 4)	(138, 4)	0.5476	0.4762
VG_EEG_C2_P1	(138, 5)	(138, 5)	0.4762	0.5238
VG_EEG_C2_SCSA	(138, 5)	(138, 5)	0.5238	0.4524
GAF_EEG_C2_P1	(138, 8)	(138, 8)	0.5238	0.4524
GAF_EEG_C2_SCSA	(138, 3)	(138, 3)	0.4762	0.5238
VG_EEG_C3_P1	(138, 22)	(138, 10)	0.5000	0.5476
VG_EEG_C3_SCSA	(138, 4)	(138, 4)	0.5000	0.4048
GAF_EEG_C3_P1	(138, 7)	(138, 7)	0.4762	0.4762
GAF_EEG_C3_SCSA	(138, 3)	(138, 3)	0.3810	0.4524
VG_EEG_C4_P1	(138, 7)	(138, 7)	0.5238	0.5000
VG_EEG_C4_SCSA	(138, 3)	(138, 3)	0.4524	0.5000
GAF_EEG_C4_P1	(138, 7)	(138, 7)	0.5238	0.5714
GAF_EEG_C4_SCSA	(138, 4)	(138, 4)	0.5238	0.4524

The outcomes of all feature extraction paths for each neurophysiological signal are illustrated through a representative example in Table 3.4. This table does not aim to cover all conducted experiments, but rather provides an illustrative instance of the detailed information on the shape of the resulting datasets, the number of samples, and the dimensionality of the feature vectors, both after correlation-based selection (denoted as

SAC) and after the subsequent ANOVA-based selection (denoted as SAA).

Each feature set in this example was then used to train two different classifiers: **Random Forest** and **Extra Trees**. The corresponding classification accuracies obtained with both models are also reported in Table 3.4, enabling a comparative assessment of the illustrated feature extraction strategies.

3.4.5 Binary classification

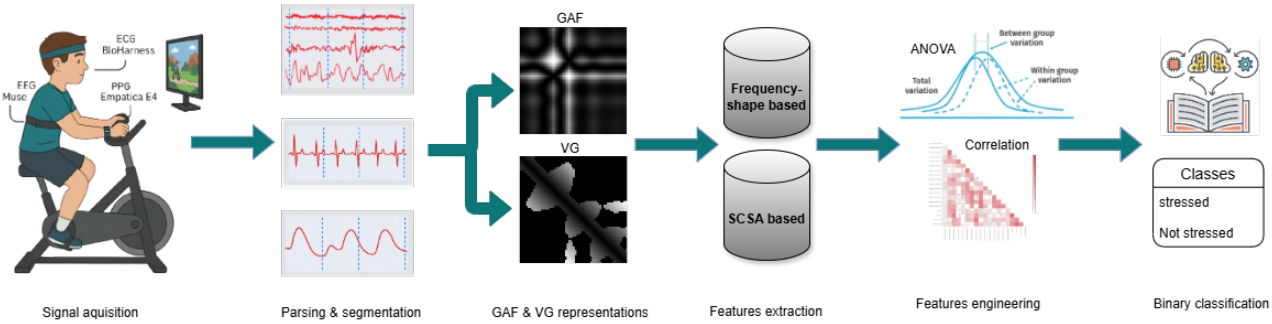


Figure 3.21: Binary classification pipeline

In our case, as shown in Fig 3.21, we started with a binary classification (stress/non-stress), the histogram of the original labels is shown in Fig 3.22

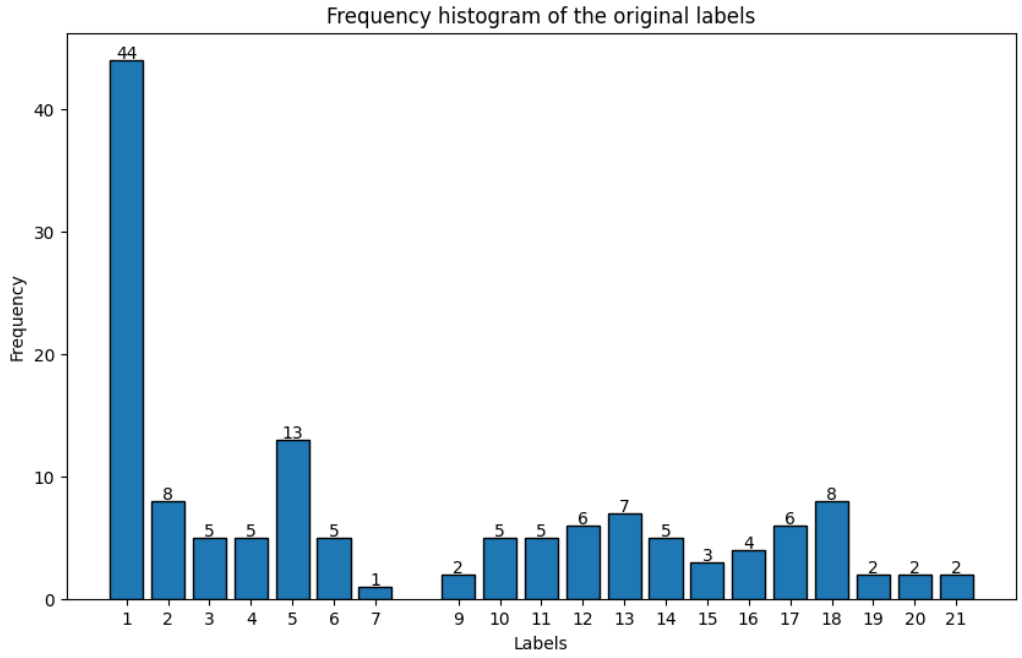


Figure 3.22: Frequency histogram of the original labels

3.4.5.1 Unbalanced classes (original split)

Afterward, the stress levels are grouped into two distinct classes:

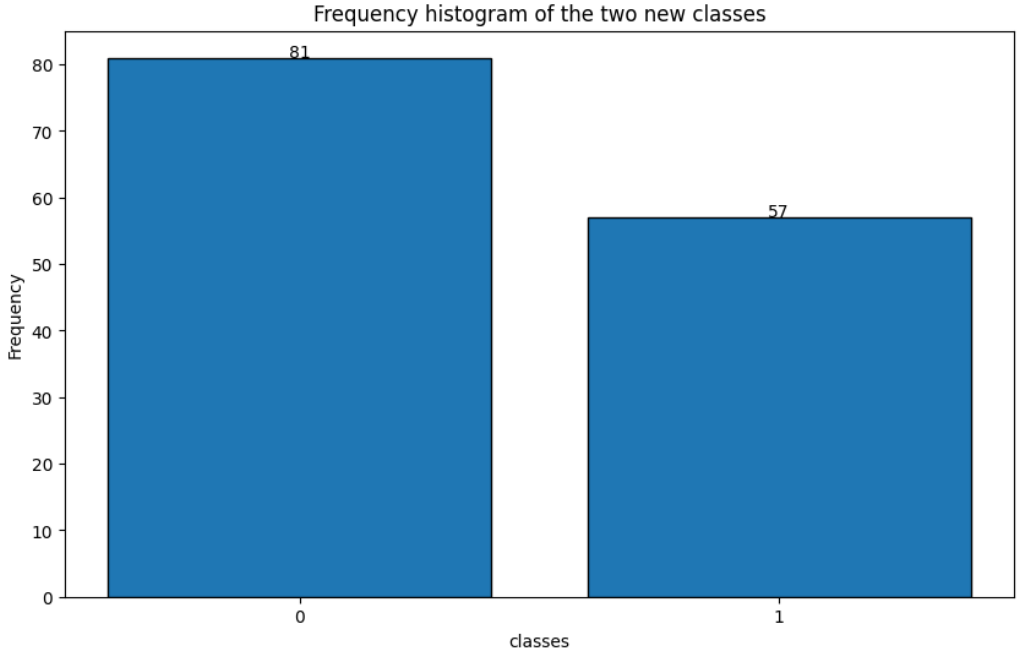


Figure 3.23: Frequency histogram of the two new classes

This regrouping is based on the participants' self-reported stress ratings and the nature of the tasks. As shown clearly in Fig 3.24, there is a noticeable threshold at a stress rating of 7 (for the case of two classes).

The first three conditions (TIMEFRAME (no bike), TIMEFRAME (14 km/h), and TIMEFRAME (28 km/h)) all yielded stress ratings below this threshold, indicating that the video game was not perceived as stressful (as it should be). Therefore, these instances were assigned to **Class 0** (non-stress).

In contrast, the remaining conditions (OUTLAST (no bike), OUTLAST (14 km/h), and OUTLAST (28 km/h)) resulted in stress ratings consistently above 7, reflecting a clear stress response. These were accordingly grouped into **Class 1** (stress).

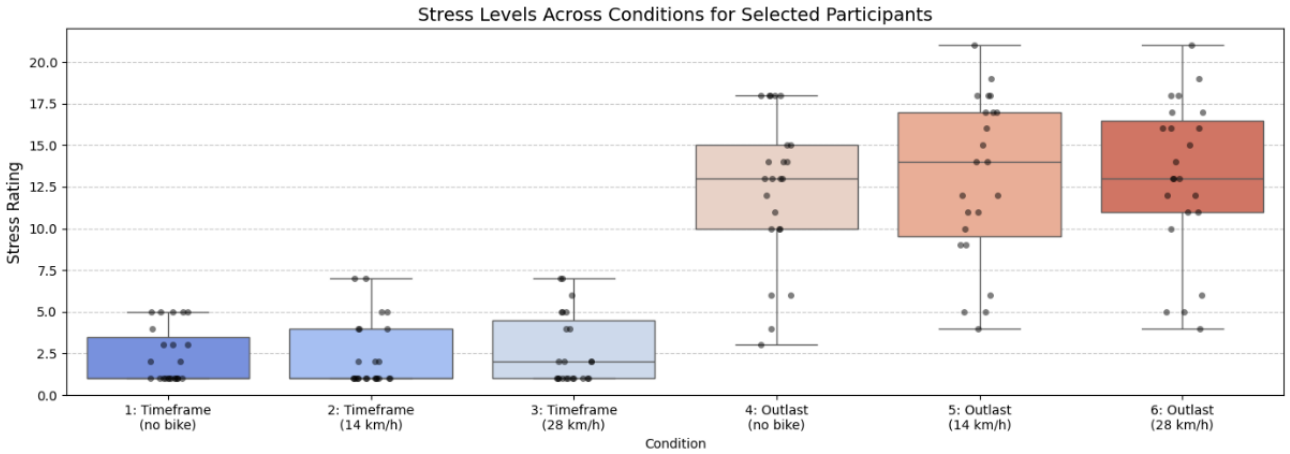


Figure 3.24: Stress levels across conditions for selected participants

After training the dataset using **LazyClassifier**, and getting the best model in each path, the table 3.5 summarizes the results for each path,

Table 3.5: Best accuracy for paths

Path	Model	Accuracy	Balanced Accuracy	F1 Score
ECG Features				
VG_ECG_P1	SGDClassifier	0.62	0.62	0.62
VG_ECG_SCSA	DecisionTreeClassifier	0.60	0.57	0.55
GAF_ECG_P1	BernoulliNB	0.62	0.58	0.55
GAF_ECG_SCSA	Perceptron	0.64	0.65	0.64
PPG Features				
VG_PPG_P1	LabelPropagation	0.57	0.56	0.56
VG_PPG_SCSA	PassiveAggressiveClassifier	0.64	0.62	0.63
GAF_PPG_P1	AdaBoostClassifier	0.67	0.65	0.65
GAF_PPG_SCSA	BaggingClassifier	0.74	0.73	0.73
EEG Features				
VG_EEG_C1_P1	NuSVC	0.69	0.67	0.68
VG_EEG_C1_SCSA	Perceptron	0.60	0.58	0.58
GAF_EEG_C1_P1	LabelPropagation	0.67	0.65	0.65
GAF_EEG_C1_SCSA	RandomForestClassifier	0.55	0.52	0.50
VG_EEG_C2_P1	DecisionTreeClassifier	0.60	0.58	0.58
VG_EEG_C2_SCSA	ExtraTreeClassifier	0.60	0.58	0.59
GAF_EEG_C2_P1	XGBClassifier	0.62	0.61	0.62
GAF_EEG_C2_SCSA	ExtraTreesClassifier	0.60	0.57	0.55
VG_EEG_C3_P1	NuSVC	0.60	0.58	0.59
VG_EEG_C3_SCSA	LGBMClassifier	0.60	0.57	0.57
GAF_EEG_C3_P1	DecisionTreeClassifier	0.67	0.66	0.66
GAF_EEG_C3_SCSA	BernoulliNB	0.57	0.53	0.47
VG_EEG_C4_P1	BernoulliNB	0.69	0.66	0.66
VG_EEG_C4_SCSA	NuSVC	0.64	0.63	0.64
GAF_EEG_C4_P1	AdaBoostClassifier	0.64	0.62	0.63
GAF_EEG_C4_SCSA	XGBClassifier	0.62	0.62	0.62

Note that the rows highlighted in yellow represent the feature extraction paths that yielded the highest accuracies. To further improve classification performance, we adopted a multimodal approach by combining features from all these high-performing paths. We then applied feature engineering techniques, including correlation analysis and ANOVA feature selection, to reduce dimensionality and keep only the relevant features.

For the multimodal classification step, the **XGBoost Classifier** was selected, achieving an accuracy of **73%**, which equals the highest accuracy observed across all individual feature extraction paths.

This choice was motivated by using the LazyClassifier to compare various models on our

best-performing feature sets, where **XGBoost classifier** consistently delivered the most promising results.

The confusion matrix below, Fig 3.25, illustrates the performance of the classifier on a binary classification task

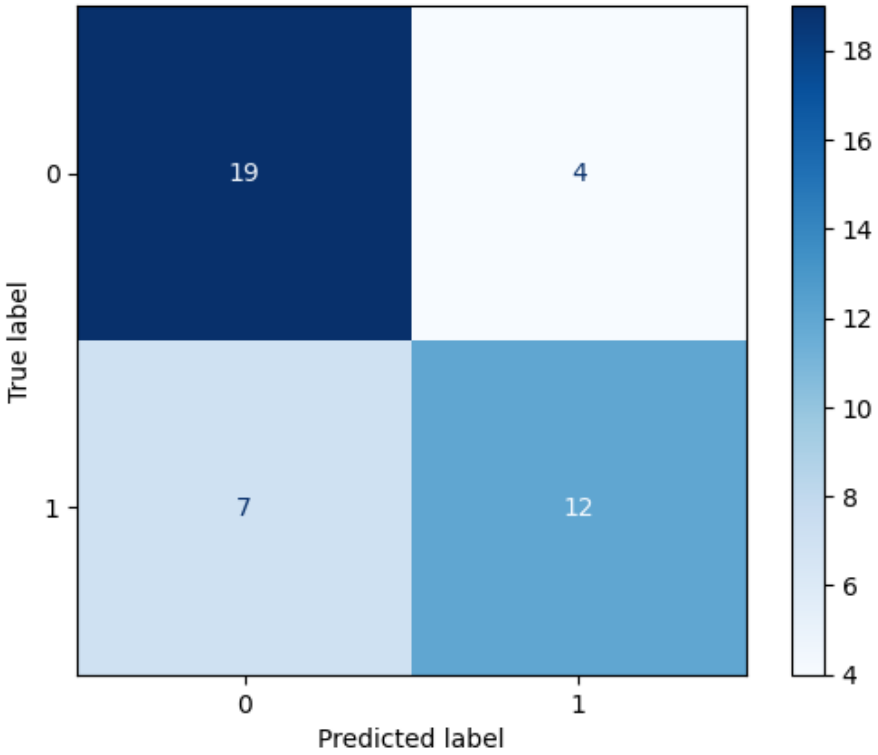


Figure 3.25: Confusion Matrix of the binary classification without downsampling

From this matrix, we can compute the following metrics:

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN} \tag{3.5}$$

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.6}$$

- For **Class 0**:

$$\text{Precision} = \frac{19}{19 + 7} \approx 0.731, \quad \text{Recall} = \frac{19}{19 + 4} \approx 0.826, \quad \text{F1-score} \approx 0.775 \tag{3.7}$$

- For **Class 1**:

$$\text{Precision} = \frac{12}{12 + 4} = 0.750, \quad \text{Recall} = \frac{12}{12 + 7} \approx 0.632, \quad \text{F1-score} \approx 0.685 \tag{3.8}$$

These results show that the classifier performs slightly better on **class 0**, with higher recall and F1-score compared to **class 1**. However, the recall for class 1 is relatively low, indicating that several positive instances were incorrectly classified. The model's poor ability to detect class 1 may be **due to the imbalance** in our dataset, as class 0 contains significantly more samples.

3.4.5.2 Balanced classes using downsampling

To address the issue observed when using the original dataset as it was, we considered making the dataset **more balanced** by applying **downsampling**. As a result, we obtained this new histogram (Fig 3.26) showing the new distribution of classes 0 and 1 after downsampling.

Remark:
We consider the new distribution to be balanced, even though the number of samples in each class is not exactly the same in Fig 3.26. The difference is minimal, with 60 samples in class 0 and 57 in class 1.

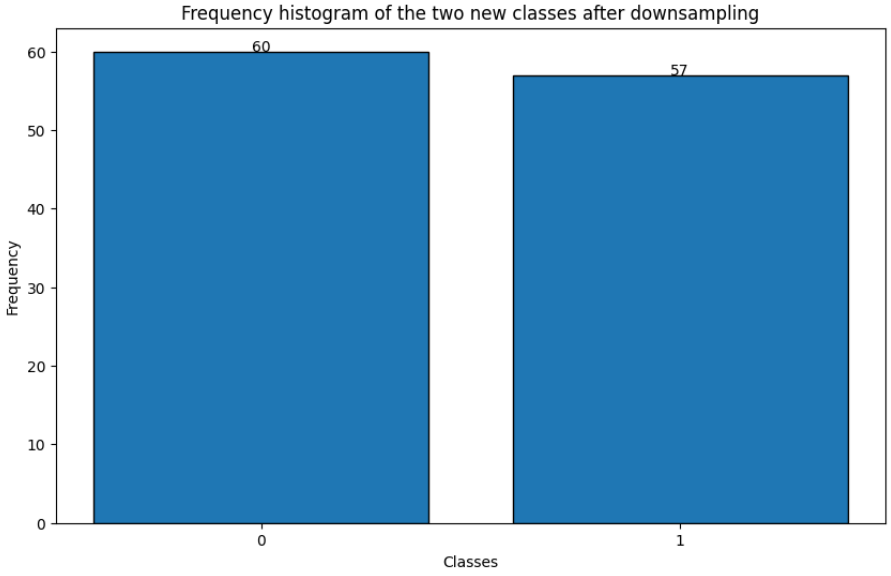


Figure 3.26: Frequency histogram of the two new classes after downsampling

Table 3.6: Best accuracy for paths after downsampling

Path	Model	Accuracy	Balanced Accuracy	F1 Score
ECG Features				
VG_ECG_P1	LogisticRegression	0.64	0.64	0.64
VG_ECG_SCSA	Perceptron	0.67	0.63	0.62
GAF_ECG_P1	KNeighborsClassifier	0.61	0.61	0.61
GAF_ECG_SCSA	SGDClassifier	0.61	0.61	0.61
PPG Features				
VG_PPG_P1	KNeighborsClassifier	0.58	0.61	0.57
VG_PPG_SCSA	SGDClassifier	0.61	0.64	0.58
GAF_PPG_P1	QDA	0.61	0.57	0.54
GAF_PPG_SCSA	CalibratedClassifierCV	0.69	0.70	0.70
EEG Features				
VG_EEG_C1_P1	LinearSVC	0.61	0.61	0.61
VG_EEG_C1_SCSA	AdaBoostClassifier	0.64	0.64	0.64

GAF_EEG_C1_P1	RidgeClassifier	0.72	0.72	0.72
GAF_EEG_C1_SCSA	ExtraTreesClassifier	0.69	0.70	0.70
VG_EEG_C2_P1	ExtraTreeClassifier	0.69	0.71	0.69
VG_EEG_C2_SCSA	ExtraTreeClassifier	0.67	0.66	0.66
GAF_EEG_C2_P1	CalibratedClassifierCV	0.61	0.62	0.61
GAF_EEG_C2_SCSA	GaussianNB	0.75	0.73	0.74
VG_EEG_C3_P1	QDA	0.72	0.73	0.72
VG_EEG_C3_SCSA	DecisionTreeClassifier	0.67	0.67	0.67
GAF_EEG_C3_P1	SVC	0.67	0.66	0.66
GAF_EEG_C3_SCSA	QDA	0.69	0.68	0.69
VG_EEG_C4_P1	NuSVC	0.69	0.71	0.69
VG_EEG_C4_SCSA	NearestCentroid	0.67	0.66	0.66
GAF_EEG_C4_P1	LabelPropagation	0.72	0.72	0.72
GAF_EEG_C4_SCSA	KNeighborsClassifier	0.64	0.66	0.63

Once again, it is important to highlight that the rows marked in yellow correspond to the feature extraction paths that yielded the highest classification accuracies. Notably, the average accuracy across the top-performing paths before downsampling was **67.5%**. After applying downsampling, the average increased to **70.66%**, reflecting a clear improvement. This enhancement can be attributed to the balancing of the dataset, which helped mitigate class imbalance and improve model performance.

After that, we adopted a multimodal strategy by combining the features of all these top-performing paths. We then applied feature engineering techniques, including correlation analysis and ANOVA feature selection, to reduce dimensionality and retain only the most relevant features.
For the classification stage, we selected the **Random Forest Classifier**, which achieved an accuracy of **78%**, exceeding the highest score obtained among all individual feature extraction paths (with 5%).
The confusion matrix shown in Fig 3.27 illustrates the classifier's performance on the binary classification task.

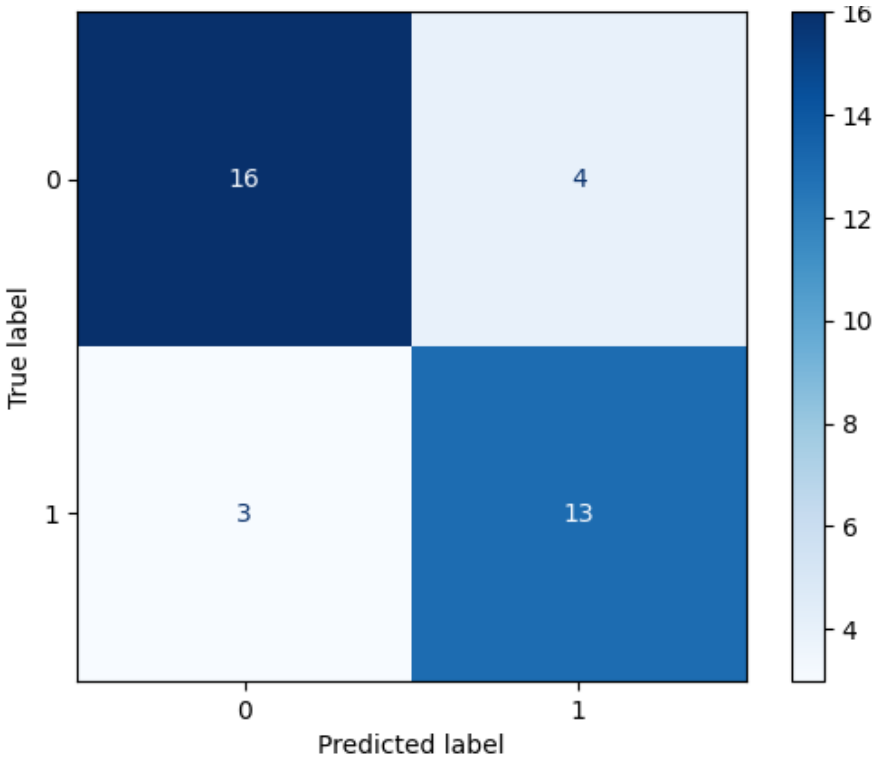


Figure 3.27: Confusion Matrix of the binary classification with downsampling

From this matrix, we compute the following metrics:

- For **Class 0**:

$$\text{Precision} = \frac{16}{16+3} \approx 0.842, \quad \text{Recall} = \frac{16}{16+4} \approx 0.800, \quad \text{F1-score} \approx 0.820 \tag{3.9}$$

- For **Class 1**:

$$\text{Precision} = \frac{13}{13+4} \approx 0.765, \quad \text{Recall} = \frac{13}{13+3} \approx 0.812, \quad \text{F1-score} \approx 0.788 \tag{3.10}$$

These results demonstrate a significant improvement in the classifier's ability to detect both classes compared to the model trained without downsampling. In particular, the recall for class 1 increased from **0.632** to **0.812**, showing that downsampling successfully addressed the data imbalance and helped the model better detect the minority class.

Permutation test:

Fig 3.28 illustrates the results of the permutation test conducted on the test set. The histogram represents the distribution of classification accuracies obtained when the labels were randomly shuffled 500 times, simulating the scenario in which there is no real relationship between features and labels. The red dashed line indicates the actual accuracy score achieved by the model on the correctly labeled data.

The model obtained an accuracy of 72%, while the majority of permutation scores are concentrated around lower values. The computed p-value of 0.0319 indicates that fewer than 4% of the random permutations reached or exceeded the actual model accuracy, providing evidence that the model performs significantly better than chance.

Although the red line is not extremely far from the distribution of random scores, it still lies on the tail of the histogram. This suggests that while the model does learn meaningful patterns, the signal in the data might be relatively weak or the classes are not perfectly separable (subjective questionnaire). Nevertheless, the result remains statistically significant ($p < 0.05$), justifying confidence in the model's predictive power.

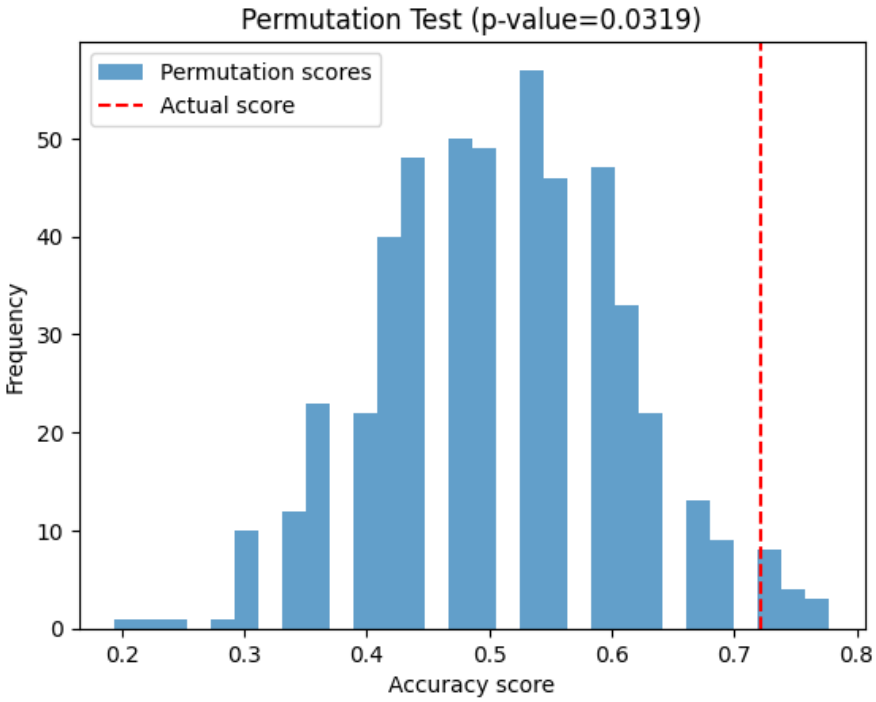


Figure 3.28: Permutation Test

3.4.5.3 Balanced classes using data augmentation in feature space

To improve the performance (which is closely tied to the number of training samples), we will apply data augmentation to the minority class instead of downsampling the majority class. Specifically, we will use the SMOTE technique to generate synthetic samples for class 1, while keeping class 0 unchanged.

SMOTE (Synthetic Minority Oversampling Technique) is a widely used statistical method for addressing class imbalance in classification problems. It works by generating synthetic instances for the minority classes, rather than simply duplicating existing samples. Importantly, SMOTE does not modify or reduce the number of instances in the majority class.

The algorithm operates in the feature space by selecting each minority instance and generating new, synthetic samples based on its nearest neighbors. For a given minority sample x_i, the algorithm identifies its k-nearest neighbors $\{x_{\text{nn}_1}, \ldots, x_{\text{nn}_k}\}$ from the same class. Then, new samples are created by linear interpolation between x_i and a randomly selected neighbor x_{nn}, using the formula:

$$x_{\text{new}} = x_i + \lambda(x_{\text{nn}} - x_i), \quad \lambda \sim \mathcal{U}(0,1), \tag{3.11}$$

where $\mathcal{U}(0,1)$ denotes the continuous uniform distribution over $[0,1]$. This process allows SMOTE to synthesize new points that lie along the line segments connecting a minority sample and its neighbors, thereby producing more diverse and generalized samples.

SMOTE Algorithm

Let $\mathcal{D} = \{(x_i, y_i)\}$ be the original dataset, where $x_i \in \mathbb{R}^d$ and $y_i \in \{\text{class labels}\}$.

Let T be the desired number of samples per class (the target for balancing).

1. **Count samples per class** For each class c, define:
$$n_c = |\{i : y_i = c\}| . \tag{3.12}$$
In our case $n_c = \{81, 57\}$

Build the set of classes to oversample: (T = 81)
$$\text{sampling\_strategy} = \{c \mid n_c < T\} = \{class1 = 57\}. \tag{3.13}$$

2. **Determine k-nearest-neighbors** Let:
$$n_{\min} = \min_{c:n_c<T} n_c, \tag{3.14}$$
$$k = \max\left(1,\ \min(5,\ n_{\min} - 1)\right). \tag{3.15}$$

3. **Generate synthetic samples for each class c with $n_c < T$** Let:
$$G_c = T - n_c = \{81 - 57\} \quad \text{(number of new samples to generate).} \tag{3.16}$$
Let the original feature vectors for class c be:
$$\mathcal{S}_c = \{x_i \mid y_i = c\} \subset \mathbb{R}^d. \tag{3.17}$$

For each $x_i \in \mathcal{S}_c$ (cycled as needed), repeat until G_c new points are generated:

   a. Find the k nearest neighbors of x_i within $\mathcal{S}_c \setminus \{x_i\}$, call this set $\text{NN}_k(x_i)$.

   b. Randomly select one neighbor $x_{\text{nn}} \in \text{NN}_k(x_i)$.

   c. Generate new point:
$$x_{\text{new}} = x_i + \lambda(x_{\text{nn}} - x_i), \ \text{ where } \ \lambda \sim \mathcal{U}(0, 1) \tag{3.18}$$

   d. Assign label $y_{\text{new}} = c$ to x_{new}.

4. **Final dataset** The final, balanced dataset is:
$$\mathcal{D}_{\text{balanced}} = \{(x_i, y_i)\} \cup \{(x_{\text{new}}, c)\}, \tag{3.19}$$
where for each targeted class c, we now have:
$$n_c + G_c = T \quad \text{samples.} \tag{3.20}$$

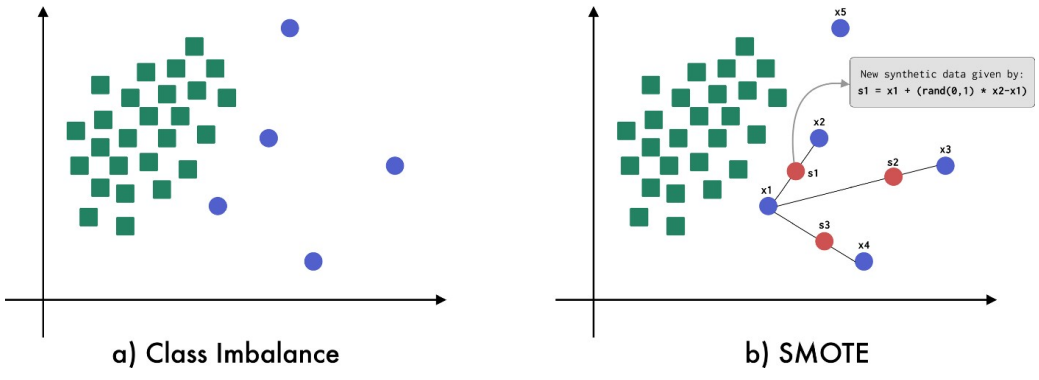


Figure 3.29: SMOTE algorithm

Fig 3.29 illustrates how SMOTE algorithm creates new samples from the original ones.

After applying data augmentation, each class in the feature dataset contains 81 samples. We then proceed with the feature engineering steps, including correlation analysis and ANOVA testing, to select the most relevant features. This refinement of the dataset leads to a noticeable improvement in the model's training accuracy. The data augmentation is applied to the selected best paths (after concatenation of their features).

Based on several trials, we find that **Random Forest Classifier** gives the best performance with test accuracy equal to 88%, and Fig 3.30, shows the confusion matrix for this case.

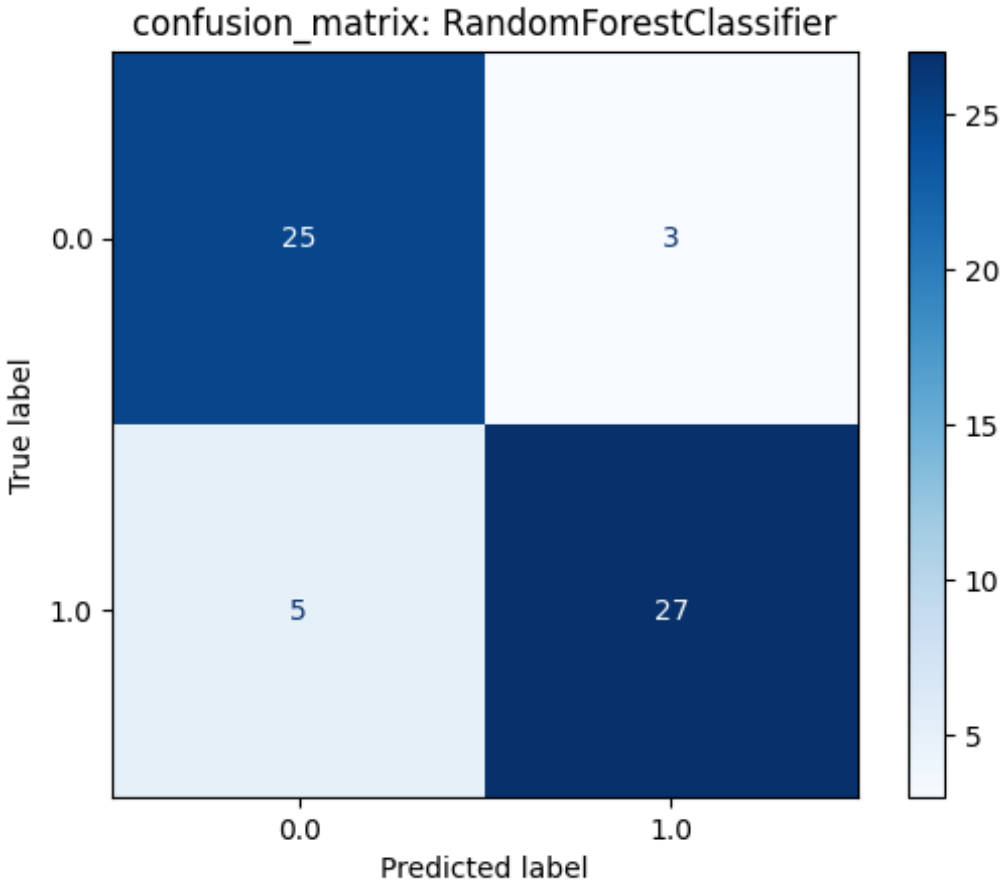


Figure 3.30: Confusion matrix for 2 classes after data augmentation

From this matrix Fig 3.30, we compute the following metrics:

- For **Class 0**:

$$\text{Precision} = \frac{25}{25+5} \approx 0.833, \quad \text{Recall} = \frac{25}{25+3} \approx 0.893, \quad \text{F1-score} \approx 0.862 \quad (3.21)$$

- For **Class 1**:

$$\text{Precision} = \frac{27}{27+3} = 0.900, \quad \text{Recall} = \frac{27}{27+5} \approx 0.843, \quad \text{F1-score} \approx 0.871 \quad (3.22)$$

From these metrics, we can clearly observe a consistent enhancement in all evaluation metrics (precision, recall, and F1-score) when using the augmented dataset compared to the unbalanced and downsampled versions.

This improvement is primarily due to the use of SMOTE-based data augmentation, which addresses the class imbalance not by reducing the number of samples in the majority class (as in downsampling), but by synthetically generating new, plausible samples for the minority class. This technique helps the model to better learn the decision boundaries for the underrepresented class without losing valuable information from the majority class.

As a result, the classifier becomes more balanced in its predictions, leading to better generalization and fairness across classes, especially noticeable in the significant gain in Class 1's F1-score and in the test accuracy increase from 73% (unbalanced) and 78% (downsampling) to 88% (data augmentation).

Permutation test:

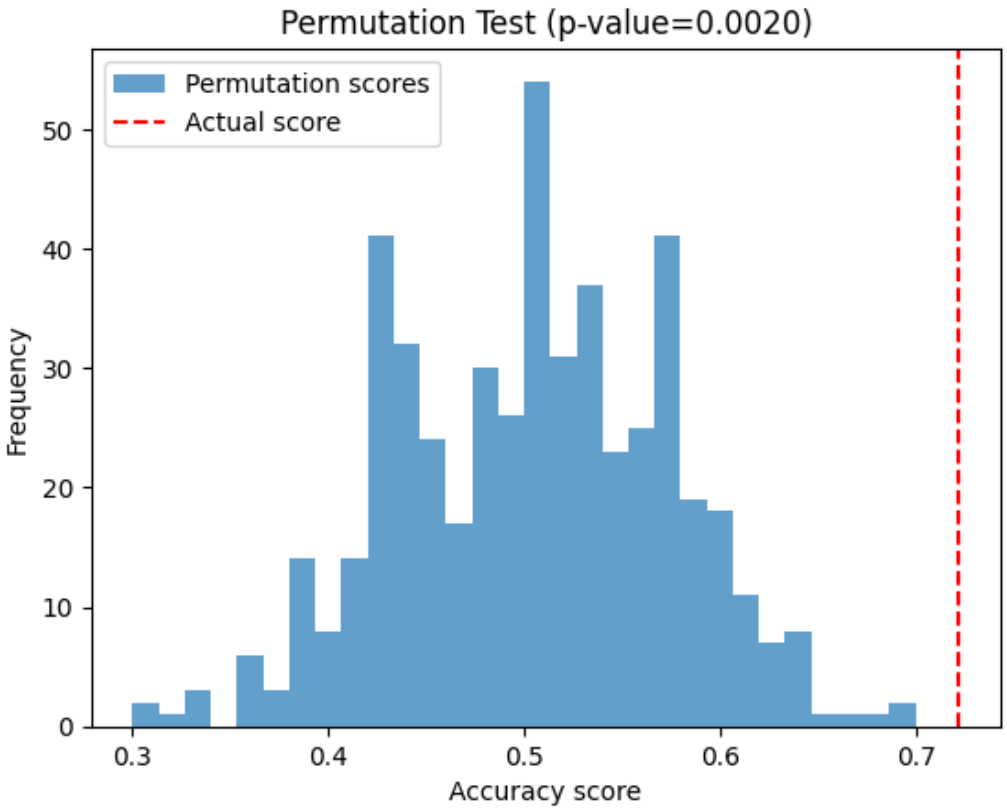


Figure 3.31: permutation test for the data augmentation with 2 classes

The permutation test results in Fig 3.31 show that the red line (representing the model's actual performance) lies far from the distribution of scores obtained under the null hypothesis. With a p-value of only 0.2%, this indicates that the likelihood of achieving such performance by random chance is extremely low. Therefore, we can confidently conclude that the model's predictions are statistically significant and not due to randomness.

Neural network model

We now proceed with training a neural network using the architecture detailed in Table 3.7, which provides a comprehensive summary of the model's structure.

Layer (type)	Output Shape	Param #
Dense (dense)	(None, 64)	45,312
BatchNormalization (batch_normalization)	(None, 64)	256
Dropout (dropout)	(None, 64)	0
Dense (dense_1)	(None, 32)	2,080
BatchNormalization (batch_normalization_1)	(None, 32)	128
Dropout (dropout_1)	(None, 32)	0
Dense (dense_2)	(None, 16)	528
BatchNormalization (batch_normalization_2)	(None, 16)	64
Dropout (dropout_2)	(None, 16)	0
Dense (dense_3)	(None, 1)	17

Table 3.7: Model architecture summary

The training process demonstrates stable convergence (Fig 3.32), with the model showing neither underfitting nor overfitting, as evidenced by a validation accuracy of approximately $\approx 87\%$

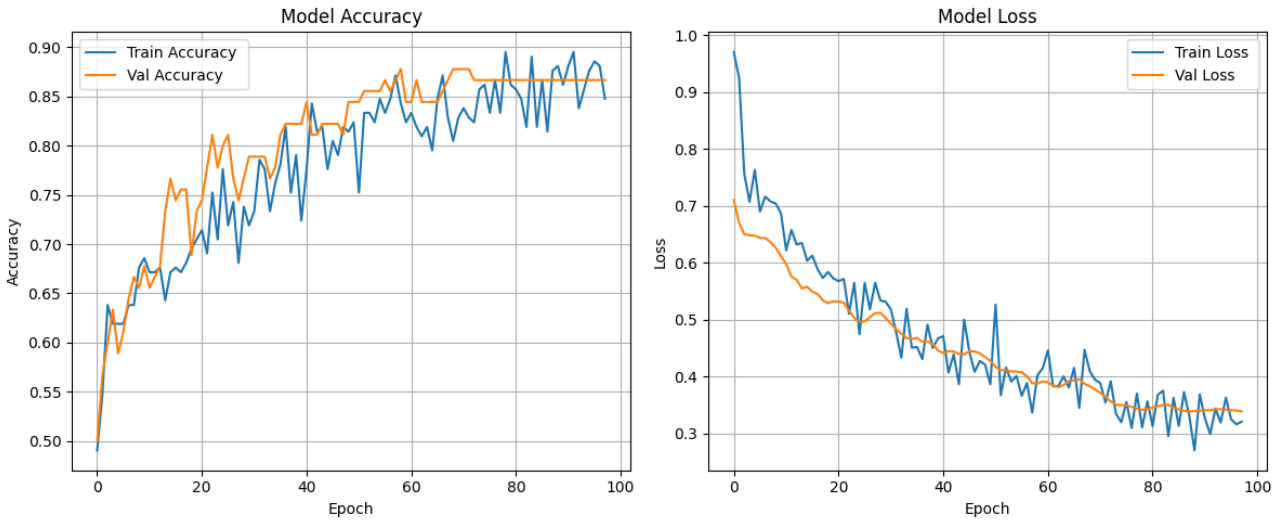


Figure 3.32: Training and validation accuracy and loss curves for the neural network.

3.4.5.4 Conclusion

In this study, we explored binary classification for stress detection using neurophysiological signals (ECG, PPG, and EEG). Initially, the dataset exhibited class imbalance, leading to biased model performance favoring the majority class (non-stress) as shown in the case of **unbalanced classes** where we achieved **73% accuracy** with XGBoost but with poor recall for the minority class (stress). Addressing this, we evaluated two approaches:

- **Downsampling**: Address the majority class to have balanced classes with improved performance, yielding **78% accuracy** with Random Forest and better recall for both classes.
- **SMOTE-Based data augmentation**: Synthetic oversampling of the minority

class further enhanced results, achieving **88% accuracy** with Random Forest and balanced precision-recall metrics.

Statistical validation (permutation tests) confirmed the significance of these results.

A neural network trained on augmented data also demonstrated strong performance (≈87% validation accuracy).

3.4.6 Three-class classification

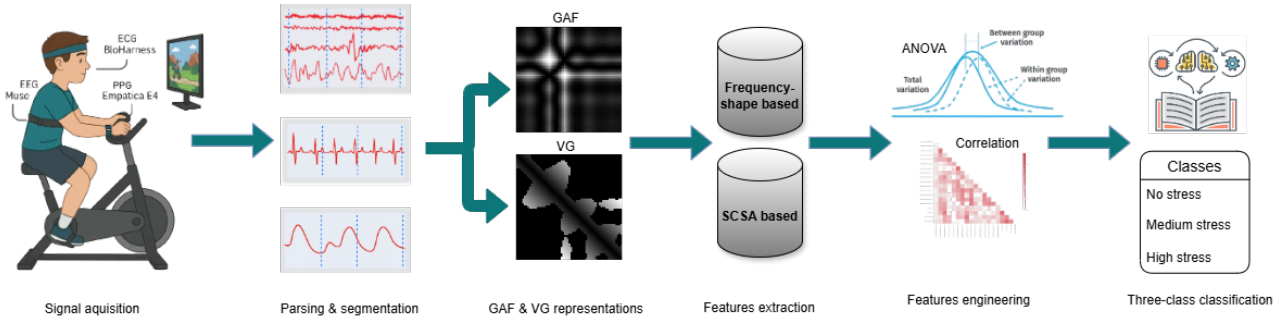


Figure 3.33: Three-class classification pipeline

Fig 3.33 illustrates the complete pipeline for the three-class classification task, which is similar to the previous one, except for a modification in the final step, the classification stage.

Figure 3.34 shows the relationship between physical effort (BORG scale) and self-reported stress levels under different conditions. Two thresholds (levels 7 and 14) divide the data into three stress classes: low, medium, and high.

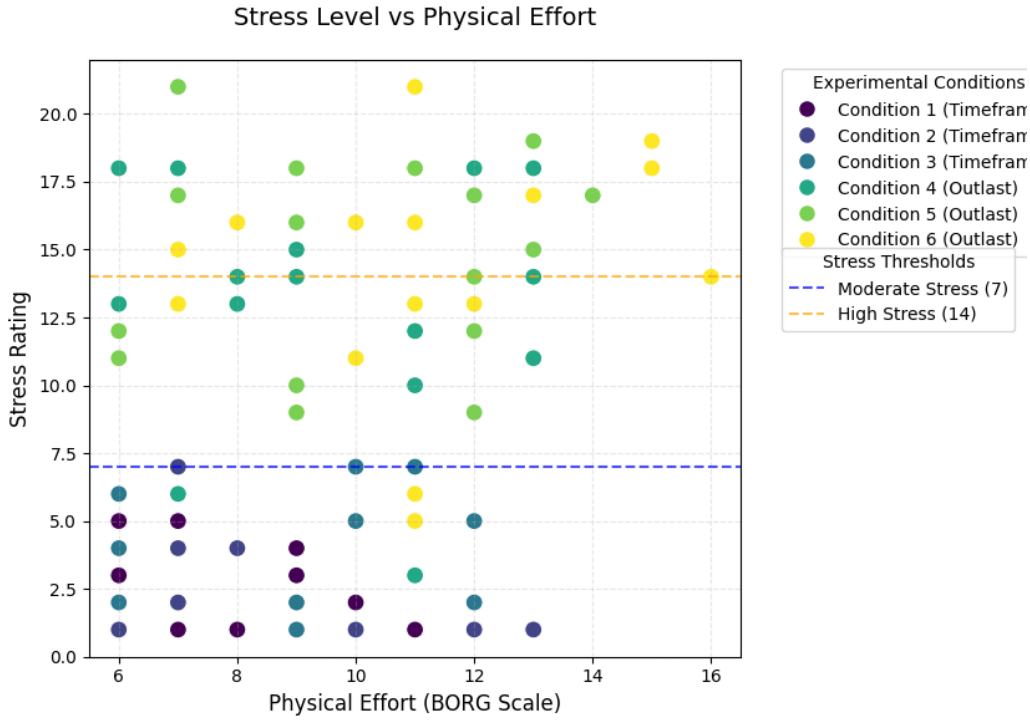


Figure 3.34: Stress level vs physical effort

These thresholds allow for a clear segmentation of the dataset and reflect the experimental

design:

- **Class 0 (Low Stress)**: Mainly from *Timeframe* conditions 1 to 3, where physical effort did not induce stress.
- **Class 1 (Medium Stress)**: Primarily from *Outlast Condition 1*, with moderate stress due to cognitive demand.
- **Class 2 (High Stress)**: From *Outlast Conditions 2 and 3*, marked by intense psychological stress.

The visualization confirms that physical effort alone does not explain stress; rather, psychological context plays a key role. This justifies the use of the selected thresholds, which capture the intended experimental distinctions.

3.4.6.1 Unbalanced classes (original split)

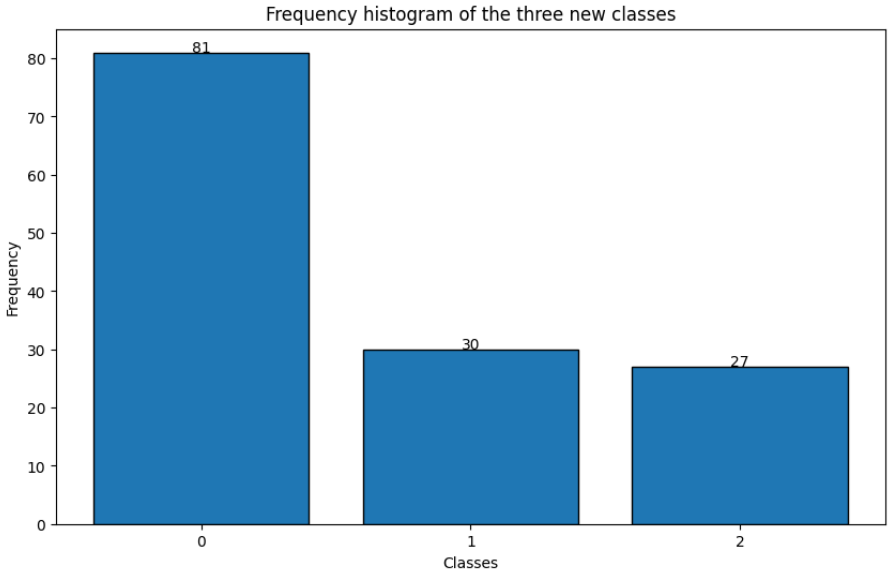


Figure 3.35: Unbalanced classes (original split)

After regrouping, each individual path was trained independently of the others. Prior to training, all paths were evaluated using correlation and ANOVA tests. Table 3.8 presents the results obtained using the **LazyClassifier**. The yellow-highlighted rows indicate the best-performing paths for each signal type.

Table 3.8: All models' performance for each feature path for the three classes distribution

Path	Model	Accuracy	Balanced Acc	F1 Score
ECG Features				
VG_ECG_P1	CalibratedCV	0.55	0.33	0.39
VG_ECG_SCSA	LGBM	0.57	0.45	0.47
GAF_ECG_P1	SGD	0.57	0.36	0.44
GAF_ECG_SCSA	Bagging	0.62	0.52	0.55
PPG Features				

Continued on next page

Table 3.8 – continued from previous page

Path	Model	Accuracy	Balanced Acc	F1 Score
VG_PPG_P1	Dummy	0.55	0.33	0.39
VG_PPG_SCSA	RandomForest	0.55	0.45	0.48
GAF_PPG_P1	PassiveAggressive	0.57	0.39	0.47
GAF_PPG_SCSA	AdaBoost	0.62	0.52	0.59
EEG Features				
VG_EEG_C1_P1	BernoulliNB	0.55	0.33	0.39
VG_EEG_C1_SCSA	BernoulliNB	0.55	0.33	0.39
GAF_EEG_C1_P1	PassiveAggressive	0.62	0.44	0.55
GAF_EEG_C1_SCSA	Bagging	0.60	0.47	0.52
VG_EEG_C2_P1	CalibratedCV	0.55	0.33	0.39
VG_EEG_C2_SCSA	Bagging	0.55	0.45	0.48
GAF_EEG_C2_P1	RidgeCV	0.55	0.33	0.39
GAF_EEG_C2_SCSA	QDA	0.57	0.36	0.44
VG_EEG_C3_P1	SVC	0.55	0.33	0.39
VG_EEG_C3_SCSA	Dummy	0.55	0.33	0.39
GAF_EEG_C3_P1	XGBoost	0.57	0.45	0.48
GAF_EEG_C3_SCSA	CalibratedCV	0.55	0.33	0.39
VG_EEG_C4_P1	GaussianNB	0.60	0.44	0.51
VG_EEG_C4_SCSA	QDA	0.55	0.37	0.48
GAF_EEG_C4_P1	AdaBoost	0.60	0.44	0.54
GAF_EEG_C4_SCSA	LDA	0.57	0.36	0.44

We selected all the best-performing paths and concatenated their features using a multimodal approach. After applying correlation and ANOVA tests, we evaluated the combined feature set using the LGBM Classifier, which gave the best performance in this scenario. The balanced accuracy after multimodality was **59%**, which is greater than the performance from the best individual path, indicating that performance gain was increased relatively to the unimodal performance. This is due to multimodality.
However, this accuracy is so low, and this lack of improvement is likely due to the severe class imbalance.

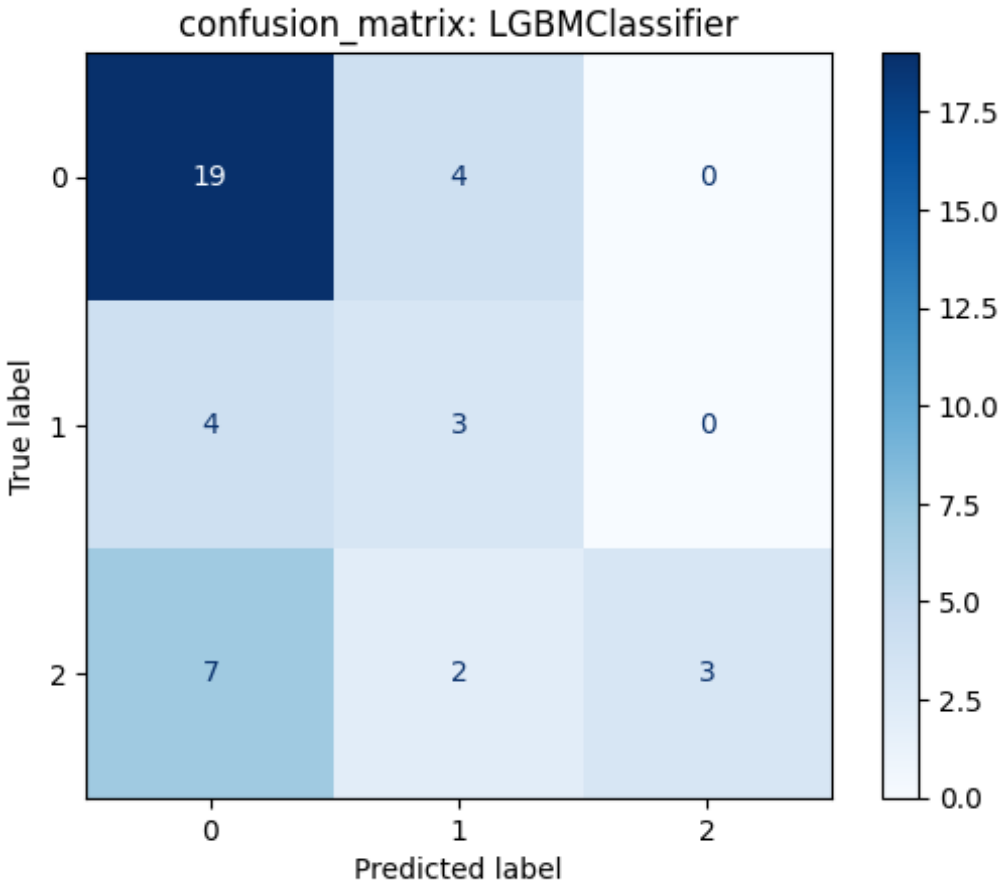


Figure 3.36: Confusion matrix of unbalanced three classes

From the confusion matrix in Fig 3.36 we calculated these metrics:

- For **Class 0**:

$$\text{Precision} = \frac{19}{19+4+7} \approx 0.633, \quad \text{Recall} = \frac{19}{19+4} \approx 0.826, \quad \text{F1-score} \approx 0.716 \tag{3.23}$$

- For **Class 1**:

$$\text{Precision} = \frac{3}{3+4+2} \approx 0.333, \quad \text{Recall} = \frac{3}{3+4} \approx 0.429, \quad \text{F1-score} \approx 0.375 \tag{3.24}$$

- For **Class 2**:

$$\text{Precision} = \frac{3}{3+0+0} = 1.000, \quad \text{Recall} = \frac{3}{3+7+2} = 0.250, \quad \text{F1-score} \approx 0.400 \tag{3.25}$$

From the classification results, we observe that:

- **Class 0** is relatively well predicted with a high recall of approximately 82.6% and a moderate precision of 63.3%. This indicates that most actual instances of Class 0 are correctly identified, although some instances from other classes are mistakenly classified as Class 0.

- **Class 1** suffers from both low precision (33.3%) and low recall (42.9%). This suggests that the model has difficulty distinguishing this class both in terms of correct

identification and avoiding misclassification. This class may require more representative features or balanced samples.

- **Class 2** achieves perfect precision (100%) but has a very low recall (25%). This means that while all predicted Class 2 instances are correct, the model fails to detect most of the true Class 2 samples, leading to many false negatives. This behavior is typical in imbalanced classification, where the model becomes overly conservative in predicting minority classes.

Overall, these metrics highlight a performance imbalance across classes. While Class 0 is handled reasonably well, Classes 1 and especially 2 require further attention. This raises the question: **If we apply data augmentation in the feature space of the concatenated data from the best-performing paths, can we achieve a similar performance gain as observed in the binary classification case?**

3.4.6.2 Balanced classes using data augmentation in feature space

By applying SMOTE for data augmentation (to the concatenated data from the best-performing paths of the original split), we increased the number of samples in the minority classes (Class 1 and 2) to match the majority class (Class 0), resulting in 81 samples per class. Using the **Extra Trees Classifier**, which demonstrated the best performance in this case, we achieved an improved test accuracy $\approx$ **84%**.
As illustrated in Fig 3.37, the confusion matrix exhibits a clearly defined diagonal with strong blue intensity, indicating that the model performs significantly better in this balanced scenario compared to the previous unbalanced case.

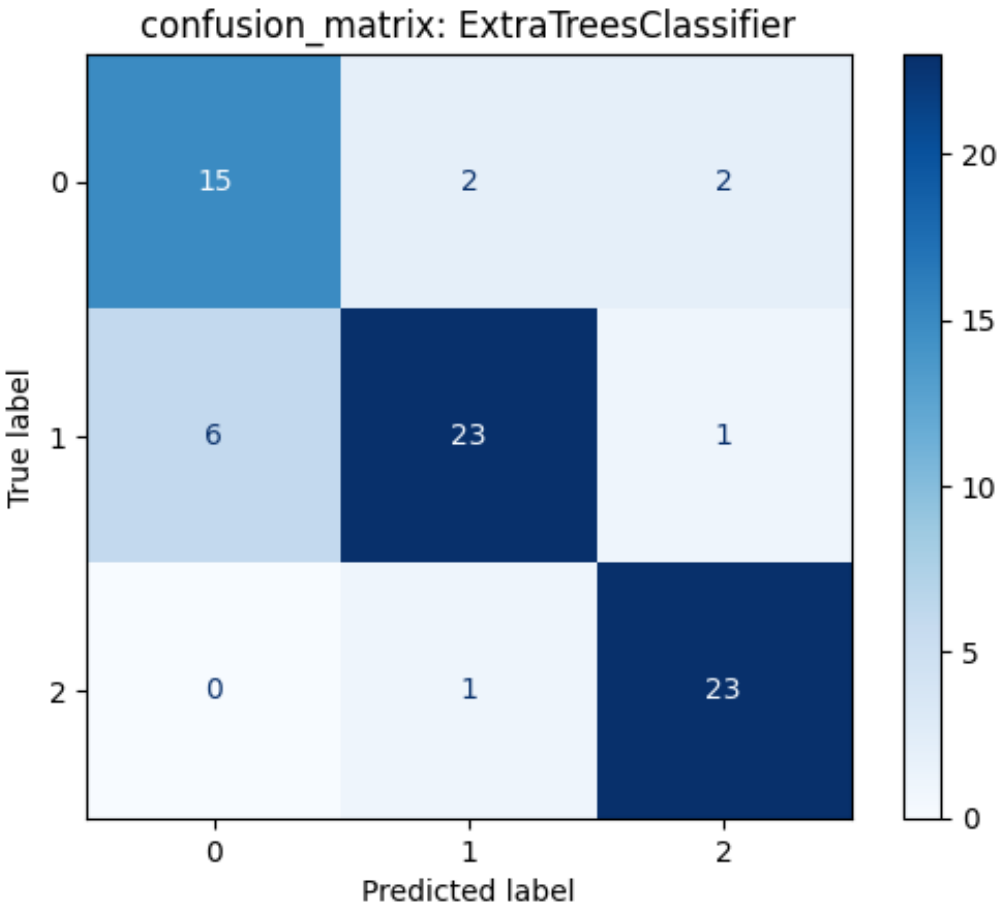


Figure 3.37: Confusion matrix of balanced three classes using data augmentation

Based on the confusion matrix shown in Fig 3.37, we computed the classification metrics:

- For **Class 0**:

$$\text{Precision} = \frac{15}{15+6} \approx 0.714, \quad \text{Recall} = \frac{15}{15+4} \approx 0.789, \quad \text{F1-score} \approx 0.750 \tag{3.26}$$

- For **Class 1**:

$$\text{Precision} = \frac{23}{23+3} \approx 0.885, \quad \text{Recall} = \frac{23}{23+7} \approx 0.767, \quad \text{F1-score} \approx 0.821 \tag{3.27}$$

- For **Class 2**:

$$\text{Precision} = \frac{23}{23+3} \approx 0.885, \quad \text{Recall} = \frac{23}{23+1} \approx 0.958, \quad \text{F1-score} \approx 0.920 \tag{3.28}$$

Permutation test:

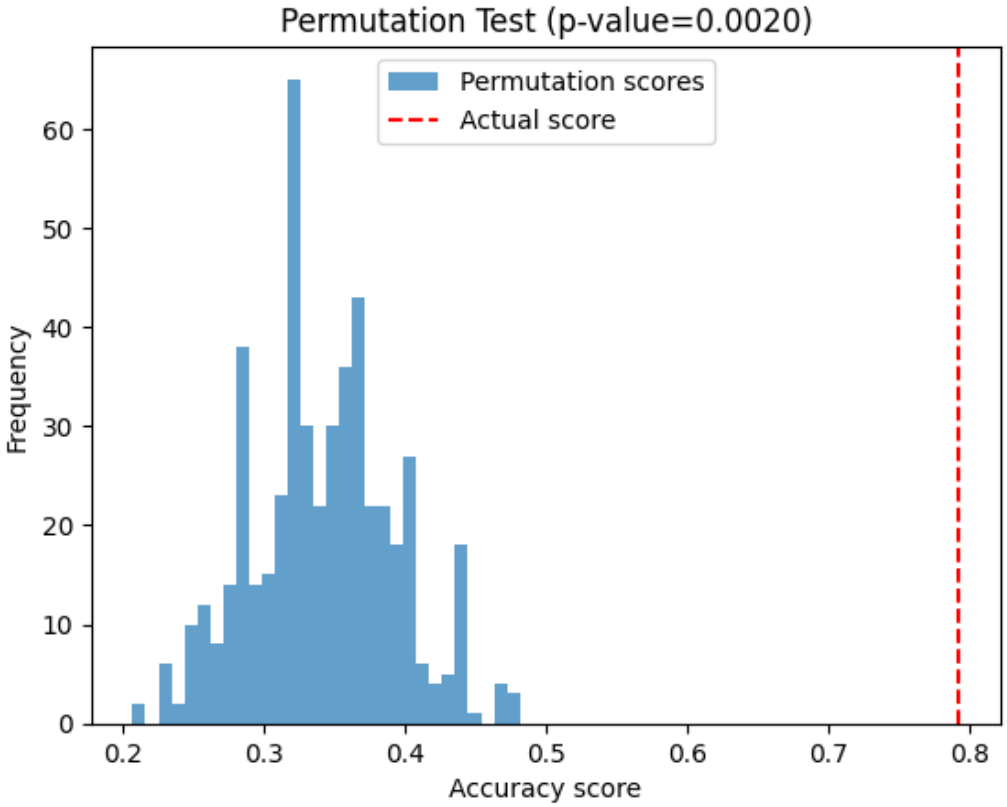


Figure 3.38

Figure 3.38 presents the results of the permutation test. The histogram shows the distribution of accuracy scores obtained under the null hypothesis, where class labels are randomly shuffled. The red dashed line represents the actual model accuracy (approximately 79%), which lies far in the tail of the distribution. The associated p-value is 0.002, meaning that only 0.2% of the randomized models achieved a performance equal to or greater than the original model. This very low p-value strongly suggests that the model's performance is statistically significant and not due to random chance.

neural network model The same neural network architecture used for the binary classification task was employed here; however, in this case, the final dense layer uses a softmax activation function with three output units corresponding to the three classes. As illustrated in Fig 3.39, both the training and validation accuracy curves show excellent performance, converging towards approximately 90%, which demonstrates the model's strong generalization capability in this multi-class scenario.

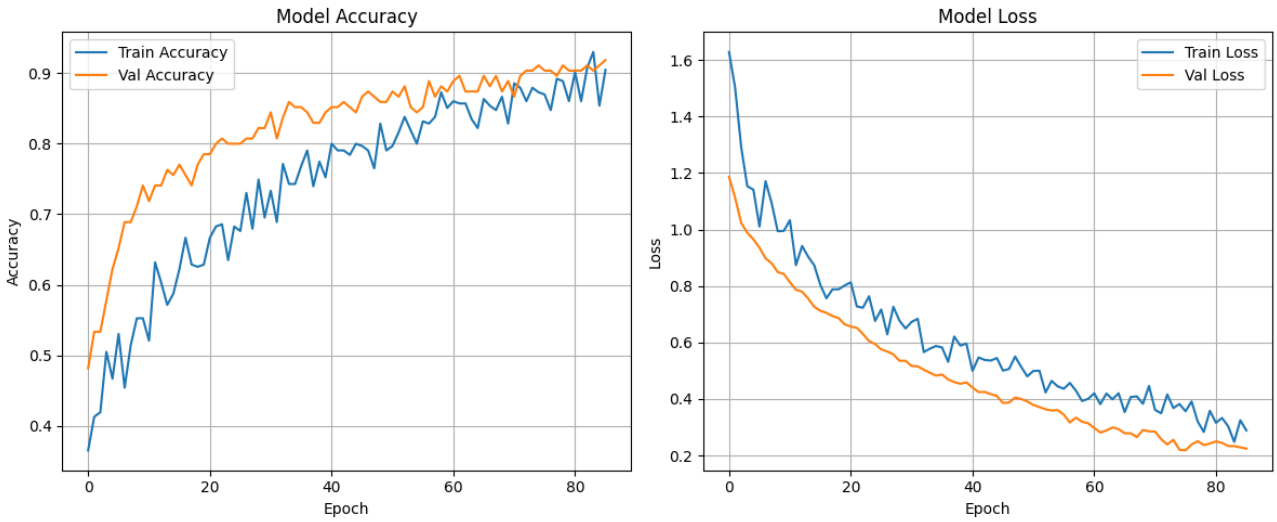


Figure 3.39: Accuracy and loss curves for 3 augmented classes

3.4.6.3 Conclusion

Based on our results, **applying data augmentation in the feature space of the concatenated data from the best-performing paths does lead to a significant performance improvement**, closely mirroring the gain observed in the binary classification scenario. This confirms the effectiveness of feature-space augmentation even in the more challenging multi-class context. However, while augmentation improves performance, it remains a surrogate for real-world variability. **Therefore, collecting and processing more real data is essential** to further enhance the model's generalization and robustness under realistic conditions.

3.5 Results and discussion

Table 3.9: Summary of results for two classes

Signal	best path in the signal	Accuracy
ECG only	GAF with SCSA	65%
PPG only	GAF with SCSA	73%
EEG channel 1 only	VG with P1	67%
EEG channel 2 only	GAF with P1	61%
EEG channel 3 only	GAF with P1	66%
EEG channel 4 only	VG with P1	66%
Multimodality (unbalanced)	Fusion	73%
Multimodality (balanced with downsampling)	Fusion	78%
Multimodality (balanced with SMOTE)	Fusion	88%

Table 3.10: Summary of results for three classes

Signal	**best path in the signal**	**Accuracy**
ECG only	GAF with SCSA	52%
PPG only	GAF with SCSA	52%
EEG channel 1 only	GAF with SCSA	47%
EEG channel 2 only	VG with SCSA	45%
EEG channel 3 only	GAF with P1	45%
EEG channel 4 only	VG with P1	44%
Multimodality (unbalanced)	fusion	59%
Multimodality (balanced with SMOTE)	fusion	84%

From Table 3.9, it is evident that the multimodal approach significantly improves classification performance, especially when applied to a balanced dataset. In the case of downsampling, the accuracy increases from 73% to 78%, marking a 5% gain. When data augmentation is applied using SMOTE, the accuracy further rises to 88%, highlighting the effectiveness of combining multiple signals. Similarly, in Table 3.10, even with an unbalanced dataset, the use of multimodality improves the accuracy from 52% (best single modality) to 59%. With data augmentation, the performance reaches 84%. These results demonstrate that leveraging the complementary information from ECG, EEG, and PPG signals through a multimodal fusion strategy leads to more robust and accurate stress classification outcomes.

Table 3.11: Evaluation metrics for each class in different dataset configurations for two class distributions

Dataset Type	**Class**	**Precision**	**Recall**	**F1-Score**	**Test accuracy**
Unbalanced	Class 0	0.731	0.826	0.775	73% (XGBoost)
	Class 1	0.750	0.632	0.685	
Balanced (Downsampling)	Class 0	0.842	0.800	0.820	78% (RF)
	Class 1	0.765	0.812	0.788	
Balanced (Augmentation)	Class 0	0.833	0.893	0.862	88% (RF)
	Class 1	0.900	0.843	0.871	87% (NN)

Table 3.12: Evaluation metrics for each class in different dataset configurations for three class distributions

Dataset Type	**Class**	**Precision**	**Recall**	**F1-Score**	**Test accuracy**
Unbalanced	Class 0	0.633	0.826	0.716	
	Class 1	0.333	0.429	0.375	59% (LGBM)
	Class 2	1.000	0.250	0.400	
Balanced (Augmentation)	Class 0	0.714	0.789	0.750	84% (ET)
	Class 1	0.885	0.767	0.821	
	Class 2	0.885	0.958	0.920	90% (NN)

From Tables 3.11 and 3.12, it is evident that class imbalance significantly deteriorated the model's performance. In both two-class and three-class settings, unbalanced datasets led to lower precision, recall, and F1-scores, particularly for minority classes. Additionally, the limited amount of data negatively impacted the classifier's ability to generalize, especially in the unbalanced configurations.

Conversely, the use of data balancing strategies, such as downsampling and augmentation, substantially improved the performance across all classes. This improvement is reflected in both individual metrics (precision, recall, and F1-score) and overall test accuracy, with the best results obtained using data augmentation combined with either Random Forest or Neural Network classifiers. These results highlight the importance of addressing class imbalance and data scarcity to build robust and fair models.

The performance achieved by all models was not due to chance. This was validated using permutation tests, which consistently yielded p-values below 0.05 across all experiments, confirming the statistical significance of the results. Furthermore, the training curves of accuracy and loss from the neural network models demonstrated stable and consistent learning behavior, with no signs of underfitting or overfitting. This reinforces the reliability and generalization capability of the proposed models.

3.6 Conclusion

This chapter presents the key contributions and findings of our project. The core of our work demonstrated that a **multimodal approach** is effective for neurophysiological signal processing and classification. By combining multiple signal sources, we observed clear performance improvements, even in complex classification scenarios.

Our proposed pipeline begins with **signal filtering** using the `NeuroKit2` library's methods, ensuring cleaner and more reliable data. This preprocessed signal is then transformed into **image representations** through techniques such as **VG** and **GAF**, capturing both spatial and temporal features that are not evident in raw time series data.

Following this, we applied **feature extraction** methods including **SCSA** and **P1**, which effectively highlighted informative patterns from the signal-derived images. The **feature engineering** stage employed statistical techniques such as **Spearman correlation** and the **ANOVA test** to select the most relevant and discriminative features, significantly reducing dimensionality while retaining meaningful information.

Next, we incorporated **multimodality**, which proved to be a key factor in improving classification accuracy. The fusion of data from different signal sources enhanced the model's ability to generalize across diverse physiological conditions and stress levels.

Model training followed these stages, leveraging the engineered features and benefiting from the structured pipeline. The results demonstrated that this end-to-end approach is highly effective in both **binary** and **three-class** classification tasks.

A major challenge faced was the issue of **dataset imbalance**, which led to biased learning favoring the majority class and a higher misclassification rate for underrepresented classes. To address this, we applied several strategies:

- **Downsampling** the majority class,
- **Data augmentation** using the SMOTE algorithm.

These techniques helped mitigate the imbalance to some extent, though our results highlight the need for more diverse and representative datasets. We recommend expanding datasets with a larger number of participants and improving metadata quality.

In summary, the pipeline, starting from signal filtering, through signal-to-image representation, feature extraction and engineering, multimodal integration, and concluding with model training, proved to be a robust and effective framework for stress level classification based on neurophysiological signals.

General Conclusion and Perspectives

Cognitive stress has emerged as a critical modern challenge, impairing performance in high-stakes domains while increasing risks of cardiovascular and neurological disorders. Traditional assessment methods like subjective surveys or single-modality biosensors are limited due to recall bias, temporal delays, and inability to capture stress complexity. Wearable neurotechnology now enables breakthrough multimodal monitoring through synchronized EEG, ECG, and PPG measurements. Our research demonstrates that this integrated approach overcomes unimodal limitations by providing comprehensive, real-time stress quantification, a vital capability for adaptive interventions in workplaces, athletics, and healthcare. The fusion of complementary biosignals creates a robust physiological stress fingerprint, transforming our ability to detect and manage cognitive load dynamically.

At the heart of our contribution is a novel processing pipeline that transforms raw neurophysiological signals into actionable stress classifications. By employing advanced techniques from neurokit2 library, we've achieved significant improvements in signal quality. The innovative conversion of 1D time-series data into 2D representations through Visibility Graphs and Gramian Angular Fields has allowed us to capture both temporal dynamics and structural patterns that conventional methods might miss. This approach, combined with careful feature extraction using both spectral methods and spatial descriptors and features engineering using correlation analysis and ANOVA test, has yielded a system capable of distinguishing not just between stress and non-stress states, but between varying degrees (3) of stress intensity.

Our results speak to the effectiveness of this multimodal fusion. The framework achieved 88%, 87% accuracy in binary stress classification and 84%, 90% accuracy in the more challenging three-class differentiation (low, medium, and high stress), outperforming unimodal approaches by significant margins. Particularly noteworthy is our handling of class imbalance through SMOTE augmentation, which addressed a common pitfall in stress detection systems and improved minority-class recall by $\approx 30\%$. These advances are not merely academic, they represent concrete progress toward practical, real-world stress monitoring solutions.

What sets this work apart is its pioneering combination of three distinct signal modalities for fine-grained stress classification, coupled with innovative signal-to-image transformation techniques. To our knowledge, this is the first system to successfully classify three stress levels by fusing EEG, ECG, and PPG data while leveraging the unique strengths of SCSA-based feature extraction. The richness of the PASS dataset, with its inclusion of both cognitive tasks and physical activity, has been instrumental in developing and validating these methods under conditions that approximate real-world variability.

Looking ahead, this framework opens numerous exciting possibilities. Future work may include:

- Replacing handcrafted features with deep learning techniques, particularly convolutional neural networks (CNNs), which can be optimized for processing the GAF and VG representations developed in this work.
- Developing software tools based on these models to facilitate broader usage and evaluation, particularly in portable and mobile environments.
- Implementing the system on FPGA hardware, which could significantly accelerate computationally intensive tasks such as signal-to-image transformation, feature extraction, and real-time classification, thereby enabling deployment in wearable devices.
- Applying the framework in various domains, including athletic training optimization, workplace wellness monitoring, and clinical stress disorder detection.
- Expanding dataset collection efforts by ensuring balanced, diverse, and representative data from different demographic groups. This will help reduce bias and improve the generalizability and fairness of the models.
- Addressing ethical considerations, including data privacy and security concerns, which are critical as physiological monitoring systems transition toward real-world use.
- Improving accessibility and inclusivity, ensuring that developed tools are effective and usable across a wide range of populations and contexts.

Bibliography

[1] Mohammed Abo-Zahhad, Sabah Mohamed Ahmed, and Sherif Nagib Abbas Seha. A new eeg acquisition protocol for biometric identification using eye blinking signals. *International Journal of Intelligent Systems Technologies and Applications*, 2015(06):48–54, 2015.

[2] Agency for Healthcare Research and Quality. Nasa task load index (nasa-tlx). https://digital.ahrq.gov/health-it-tools-and-resources/evaluation-resources/workflow-assessment-health-it-toolkit/all-workflow-tools/nasa-task-load-index, 2024.

[3] JET Akinsola, FY Osisanwo, O Awodele, JO Hinmikaiye, O Olakanmi, and J Akinjobi. Supervised machine learning algorithms: Classification and comparison. *International Journal of Computer Trends and Technology (IJCTT)*, 2017.

[4] F. Al-Shargie, M. Kiguchi, N. Badruddin, S. C. Dass, A. F. Hani, and T. B. Tang. Mental stress assessment using simultaneous measurement of eeg and fnirs. *Biomedical Optics Express*, 7(10):3882–3898, 2016.

[5] American Psychological Association. State-trait anxiety inventory (stai).

[6] Prof. Sid-Ahmed BERRANI. Convolutional neural network (cnn). Ecole Nationale Polytechnique, 2023.

[7] Jordan J. Bird, Diego R. Faria, Luis J. Manso, Anikó Ekárt, and Christopher D. Buckingham. A deep evolutionary approach to bioinspired classifier optimisation for brain-machine interaction. *Complexity*, 2019:Article ID 4316548, 14 pages, 2019.

[8] Bright Brain Centre. Electroencephalogram (eeg) brainwaves, 2025.

[9] BYJU'S. Electrocardiogram (ecg), 2024.

[10] Raymundo Cassani, Hubert Banville, and Tiago H. Falk. MuLES: An Open Source EEG Acquisition and Streaming Server for Quick and Simple Prototyping and Recording. In *IUI'15 Companion*, pages 9–12, Atlanta, GA, USA, 2015. ACM.

[11] Nazareth P. Castellanos and Valeri A. Makarov. Recovering eeg brain signals: Artifact suppression with wavelet enhanced independent component analysis. *Journal of Neuroscience Methods*, 158(2):300–312, 2006.

[12] Precious Medical Centre. Digital brain screen - cognitive assessment tool.

[13] Mohammed M. Farag. A tiny matched filter-based cnn for inter-patient ecg classification and arrhythmia detection at the edge. *Sensors*, 23(3):1365, 2023.

[14] Susan Folkman and Richard S. Lazarus. *Manual for the Ways of Coping Question-*

naire. Consulting Psychologist Press, Palo Alto, CA, 1988.

[15] Random Seed Games. Timeframe press kit. https://randomseedgames.com/press/timeframe, 2025.

[16] GameSnort. Timeframe - screenshots.

[17] GeeksforGeeks. Ml - independent component analysis.

[18] GeeksforGeeks. Supervised machine learning.

[19] GeeksforGeeks. Neural networks: A beginner's guide. https://www.geeksforgeeks.org/neural-networks-a-beginners-guide/, 2021.

[20] GeeksforGeeks. Lazy predict library in python for machine learning. https://www.geeksforgeeks.org/lazy-predict-library-in-python-for-machine-learning/, 2023.

[21] Rajdeep Ghosh, Nabamita Deb, Kaushik Sengupta, et al. Sam 40: Dataset of 40 subject eeg recordings to monitor the induced-stress while performing stroop color-word test, arithmetic task, and mirror image recognition task. *Data in Brief*, 40:107772, 2022.

[22] Thomas S. Holmes. Life situations and disease. *Journal of Academy of Psychosomatic Medicine*, 19:747, 1978.

[23] Honeywell. Bioharness 3 datasheet, 2024.

[24] Zhihu Huang and Jinsong Leng. Analysis of hu's moment invariants on image scaling and rotation. In *2010 2nd International Conference on Computer Engineering and Technology (ICCET)*, volume 7, pages 476–480. IEEE, 2010.

[25] Aapo Hyvärinen and Erkki Oja. An introduction to independent component analysis: Infomax and fastica algorithms. *ResearchGate*, 2010.

[26] Queensland Brain Institute. Limbic system - brain anatomy.

[27] Seung-Min Jung, Gi-Ran Park, and Seong-Whan Lee. Effect of eeg frequency bands on deep learning-based stress level classification. In *2020 International Conference on Electronics, Information, and Communication (ICEIC)*, pages 1–6. IEEE, Jan 2020.

[28] J. Kim, S. Lee, H. Park, and Y. Choi. Stress recognition using face images and facial landmarks. In *2020 International Conference on Electronics, Information, and Communication (ICEIC)*, pages 1–4. Institute of Electrical and Electronics Engineers Inc., 2020.

[29] M. Kisserli and S. Laleg-Kirati. Semi-classical signal analysis for denoising and classifying images. *Signal Processing*, 190:109876, 2021.

[30] Kyung-Woo Ko, Oh-Seol Kwon, Chang-Hwan Son, and Yeong-Ho Ha. Color embedding and recovery based on wavelet packet transform. *Journal of Imaging Science and Technology*, 52(1):010501–010501–10, 2008.

[31] MuSAE Lab. Pass-raw dataset, 2024.

[32] Taous-Meriem Laleg, Emmanuelle Crépeau, and Michel Sorine. Semi-classical signal analysis. *arXiv preprint arXiv:1007.0938*, 2010.

[33] Taous-Meriem Laleg-Kirati, J. Zhang, E. Achten, and H. Serrai. Spectral data denoising using semi-classical signal analysis: application to localized mrs. *NMR in Biomedicine*, 29(6):685–696, 2016.

[34] Yoshihiro Maruyama, Aimi Kawano, Shizuko Okamoto, Tomoko Ando, Yoshinobu Ishitobi, Yoshihiro Tanaka, Ayako Inoue, Junko Imanaga, Masayuki Kanehisa, Haruka Higuma, Taiga Ninomiya, Jusen Tsuru, Hiroaki Hanada, and Jotaro Akiyoshi. Differences in salivary alpha-amylase and cortisol responsiveness following exposure to electrical stimulation versus the trier social stress tests. *PLOS ONE*, 7(7):e39375, 2012.

[35] Pierre Moulin. Multiscale image decompositions and wavelets. In *Guide to Image and Video Compression: Standards and Techniques*, chapter 6, pages 123–142. Springer, 2008.

[36] Natural Evolution. L'échelle de borg : Une évaluation de l'effort perçu. https://natural-evolution.fr/echelle-de-borg/, 2024.

[37] Prabhakar K. Nayak and Niranjan U. Cholayya. Independent component analysis of electroencephalogram. *ResearchGate*, 2006.

[38] Online Biology Notes. Ecg procedure image, 2018.

[39] Neri Van Otten. Support vector machines (svm) made simple & how to tutorial.

[40] Ioanna V. Papathanasiou, Konstantinos Tsaras, Anna Neroliatsiou, and Aikaterini Roupa. Stress: Concepts, theoretical models and nursing interventions. *American Journal of Nursing Science*, 4(2-1):45–50, 2015.

[41] Mark Parent, Isabela Albuquerque, Abhishek Tiwari, Raymundo Cassani, Jean-François Gagnon, Daniel Lafond, Sébastien Tremblay, and Tiago H. Falk. PASS: A Multimodal Database of Physical Activity and Stress for Mobile Passive Body/Brain-Computer Interface Research. *Frontiers in Neuroscience*, 14:542934, 2020.

[42] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay. sklearn.neighbors.nearestcentroid. https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.NearestCentroid.html.

[43] P. Bhaskara Rao, D. Vara Prasad, and Ch. Pavan Kumar. Feature extraction using zernike moments. *International Journal of Latest Trends in Engineering and Technology (IJLTET)*, 2(2):228–233, 2013.

[44] Amit Ray. The perceived stress scale (pss) score assessment method for stress reduction: An overview. *Compassionate AI Research*, 3(9):55–61, 2024.

[45] Kerry Ressler. Protect your brain from stress, 2021.

[46] Maria Sara Nour Sadoun, Juan Manuel Vargas, Mohamed Mouad Boularas, Arnaud Boutin, François Cottin, and Taous-Meriem Laleg-Kirati. Cognitive stress detection during physical activity using simultaneous, mobile eeg and ecg signals. In *IFAC Conference*, 2024.

[47] scikit-learn developers. Supervised learning — scikit-learn 1.4.2 documentation.

[48] Hans Selye. *The Stress of Life*. McGraw-Hill, New York, 1956.

[49] SupplyCore. Zephyr bioharness 3.0 side strap - performance monitoring, 2025.

[50] TechnPlay. Hypnos masque test, 2025.

[51] Duygu Sinanc Terzi. Gramian angular field transformation-based intrusion detection. *Computer Science*, 23(4):571–585, 2022.

[52] The MathWorks, Inc. MATLAB - mathworks. https://www.mathworks.com/products/matlab.html, 2024.

[53] Amirsina Torfi. Designing your neural networks. https://medium.com/data-science/designing-your-neural-networks-a5e4617027ed, 2017.

[54] Wallup.net. Video games - outlast.

[55] Thomas Wiatowski and Helmut Bölcskei. A mathematical theory of deep convolutional neural networks for feature extraction. *IEEE Transactions on Information Theory*, 64(3):1845–1866, 2018.

[56] Wikipedia contributors. Neurokit, 2024.

[57] Wikipedia contributors. Python (programming language), 2024.

[58] Wikipedia contributors. Visual studio code, 2024.

[59] Wikipedia contributors. Outlast — Wikipedia, the free encyclopedia, 2025.

[60] Wonderful Engineering. This medical smartwatch is capable of predicting seizures, 2025.

Appendix 1:
Tools used in the PASS dataset collection

1. Timeframe

TIMEframe was originally created for the Ludum Dare 27 game jam competition by Random Seed Games, inspired by games like Proteus and Dear Esther. After a very positive reception an updated version was created for commercial release to expand on the themes of the prototype.In **TIMEframe** you will discover a world in slow motion. Explore the remnants of a mysterious civilization on the brink of destruction. A melancholy soundtrack will accompany you on your journey, highlighting each location you visit with its own unique theme...culminating in a world-changing event. [15].

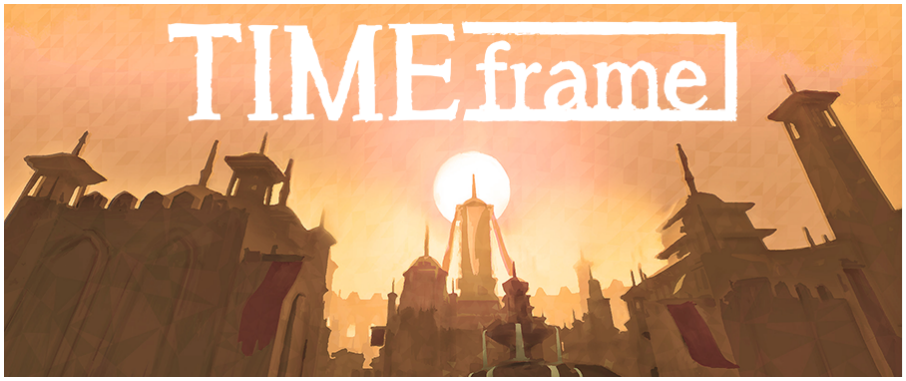


Figure 0.40: TIMEFRAME [15]

2. OUTLAST

Figure 0.41: OUTLAST [59]

Outlast (0.41) is a 2013 first-person psychological survival horror video game developed and published by Red Barrels. It revolves around freelance investigative journalist Miles Upshur, who decides to investigate Mount Massive Asylum, a remote psychiatric hospital, located deep in the mountains of Lake County, Colorado.

In **Outlast**, players control Miles Upshur, a journalist investigating the sinister Mount Massive Asylum in Colorado. The game is a first-person survival horror experience emphasizing stealth over combat. Players can move, hide, and use the environment to evade dangerous patients, as there are no weapons or health bars. Darkness dominates the asylum, requiring a camcorder's night vision to see, which consumes limited batteries scattered around the facility. Jump scares, audio cues, and collected documents enhance the game's tension and reveal deeper story elements [59].

3. MuSAE Lab EEG Server (MuLES):

The MuSAE Lab EEG Server (MuLES) is an open-source software tool developed by the Multimedia/Multimodal Signal Analysis and Enhancement Laboratory (MuSAE Lab) at INRS in Montreal. MuLES is designed to simplify the acquisition, recording, and real-time streaming of electroencephalography (EEG) data from various consumer-grade portable EEG devices. It provides a standardized interface that allows users to work with different EEG headsets without needing to delve into each device's specific software development kits (SDKs) or application programming interfaces (APIs) [10].

4. NASA-TLX & BORG tests

NASA-TLX

A subjective assessment tool for evaluating **mental workload (MWL)** during task performance. Measures workload across six dimensions:

- **Mental Demand** : Cognitive effort (e.g., thinking, calculating).
- **Physical Demand** : Intensity of physical activity required.
- **Temporal Demand** : Time pressure experienced.

- **Effort** : Perceived exertion to maintain performance.
- **Performance** : Self-rated success in task completion.
- **Frustration Level** : Emotional state (e.g., insecurity vs. contentment).

In NASA-TLX, Participants rate each dimension on a scale from **1 (low)** to **20 (high)** post-trial, uses **paired comparisons** (15 pairwise combinations) to identify dominant workload factors, reducing inter-rater and task variability.

The main purpose of this test is to quantify overall workload by weighting individual scores based on pairwise preferences, ensuring a tailored and reliable assessment.[2]

BORG test

Developed by Swedish psychophysiologist Gunnar Borg in the 1960s, the Borg Scale is a simple numerical tool for quantifying an individual's perceived exertion during exercise.

While stress assessment is crucial in sports training, rehabilitation, and physiotherapy, **BORG test** enables adjustment of workout intensity based on individual objectives and capabilities, helps avoid both overload and underload situations, reducing injury risks, allows tracking of improvements over time and training adaptation and provides tangible feedback to help maintain motivation and achieve fitness goals.[36]

Appendix 2:
Materials, Tools, and Libraries

1. MATLAB

MATLAB (Fig 0.42) is a programming and numeric computing platform used by millions of engineers and scientists to analyze data, develop algorithms, and create models [52]. Used for processing the collected data, including a custom-made script to handle experiment markers and analyze physiological signals.

Figure 0.42: MATLAB

2. VScode

Visual Studio Code, commonly referred to as VS Code, is an integrated development environment developed by Microsoft for Windows, Linux, macOS and web browsers. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded version control with Git. Users can change the theme, keyboard shortcuts and preferences, as well as install extensions that add functionality.
Visual Studio Code is proprietary software released under the "Microsoft Software License", but based on the MIT licensed program named "Visual Studio Code – Open Source" (also known as "Code – OSS"), also created by Microsoft and available through GitHub.
In the 2024 Stack Overflow Developer Survey, out of 58,121 responses, 73.6% of respondents reported using Visual Studio Code, more than twice the percentage of respondents who reported using its nearest alternative, Visual Studio. [58]

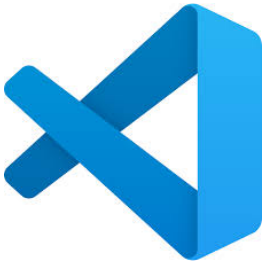


Figure 0.43: VSCode

3. Python

Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation.
Python is dynamically type-checked and garbage-collected. It supports multiple programming paradigms, including structured (particularly procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.
Guido van Rossum began working on Python in the late 1980s as a successor to the ABC programming language, and he first released it in 1991 as Python 0.9.0. Python 2.0 was released in 2000. Python 3.0, released in 2008, was a major revision not completely backward-compatible with earlier versions. Python 2.7.18, released in 2020, was the last release of Python 2.
Python consistently ranks as one of the most popular programming languages, and it has gained widespread use in the machine learning community.[57]

Figure 0.44: Python

4. Scikit-learn

scikit-learn (formerly scikits.learn and also known as sklearn) is a free and open-source machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support-vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy. Scikit-learn is a NumFOCUS fiscally sponsored project.

Figure 0.45: Scikit-learn

5. Neurokit2

NeuroKit ("nk") is an open source toolbox for physiological signal processing. The most recent version, NeuroKit2, is written in Python and is available from the PyPI package repository. NeuroKit2 includes tools to work with cardiac activity from electrocardiography (ECG) and photoplethysmography (PPG), electrodermal activity (EDA), respiratory (RSP), electromyography (EMG), and electrooculography (EOG) signals.

It enables the computation of Heart Rate Variability (HRV) and Respiratory Variability (RRV) metrics.

It also implements a variety of different algorithms to detect R-peaks and other QRS waves, including an efficient in-house R-peak detector.

For neurophysiological signals such as EEG, it supports microstates and frequency band analysis.

It also includes a comprehensive set of functions used for fractal physiology, allowing the computation of various measures of complexity (including entropy and fractal dimensions).[56]

6. Lazy Classifier (Lazy Predict)

Lazy Predict is the one tool you need for your predictive modeling projects. It is a simple and efficient tool that makes your predictive modelling projects easier and faster. Lazy Predict is a Python library that provides a simple and efficient way to make predictions. It is easy to use and easy to install. Lazy Predict is open source and is released under the MIT license. Lazy Predict is a great tool for predictive modelling projects. It is simple to use and easy to install. It is open source and released under the MIT license[20].