POLITECNICO DI TORINO

MASTER's Degree in Computer Engineering

MASTER's Degree Thesis

Machine Learning and data fusion of physiological signals for assessing a subject's stress level and cognitive load

Supervisors

Prof. Danilo DEMARCHI

Prof. Irene BURAIOLI

Candidate

Gabriele CANOVA

April 2024

To Curiosity,
endless engine of the universe.

Abstract

The increase in artificial intelligence integration across various industries underlines its significant impact on promoting secure working spaces, especially in preventing accidents and mitigating risks. With human-machine interaction (HMI), operators are often tasked with performing complex duties, raising mental workload and stress levels, potentially compromising their performance, and elevating the risk of accidents.
Stress refers to the psychological and physiological responses elicited by perceived demands exceeding an individual's coping capacity, while cognitive workload denotes the amount of mental effort and resources required to perform tasks effectively. Existing literature shows a correlation between these cognitive states and the physiological signals of the human body.
To investigate this relationship further, the eLions group at Politecnico di Torino conducted a study involving 61 subjects. These participants underwent "N-Back" and "Stroop" tests to induce cognitive load and stress state alteration, respectively. Their physiological data was monitored and assessed throughout the tests.
N-Back tests are conducted with visual, auditory, or combined stimulation, with three levels of difficulty, while the Stroop test had only one level. In total, four datasets were obtained. This study aimed to develop an innovative model that classifies stress and cognitive load levels based on machine learning algorithms using multimodal physiological signals obtained previously. To achieve the specified goal, an analysis of the state of the art was conducted as a starting point to find procedures suitable for the task.
The initial supervised experiments focused on binary classification, distinguishing samples into two states: rest and altered state. Across three datasets, results showed a remarkable accuracy rate of 100%. In the Audio N-Back dataset, an accuracy of 98% was achieved.
In analyzing the Stroop dataset, a classification accuracy of 75% was attained for stress classification through a combination of LDA as dimensionality reduction and then K-Nearest Neighbors as the classifier. Meanwhile, cognitive workload classification for the Visual N-Back task yielded an accuracy of 78% using K-PCA with LDA, while for the Audio N-Back task, it was 67% using PCA and LDA. Additionally, the Dual N-Back task achieved an accuracy of 81% with the blend of LDA and SVM.
In addressing the challenge of heavily imbalanced classes in the Stroop test and Dual N-Back dataset, outliers were identified and removed, leading to a transition to a 3-class classification. As a result, 80% and 85% accuracies were achieved for the Stroop test and Dual N-Back tasks. Moreover, models have been created to evaluate scenarios without one or more signals involved.
The project has resulted in the development of a comprehensive and versatile

framework able to retrieve, manipulate, and accurately classify the differentiation between a state of rest and an altered state based on an individual's physiological parameters. Furthermore, it can discern various degrees of cognitive workload and stress severity, even when certain biological signals are absent. This framework lays the groundwork for creating a safety device that analyses physiological signals to assess the operator's condition, especially during critical moments. It could enhance safety by providing real-time evaluations of the operator's state, thereby reducing potential risks and ensuring safer operations.

Table of Contents

List of Tables

List of Figures

Acronyms

AI

Artificial Intelligence

BDN

Bayesian Dynamic Network

BVP

Blood Volume Pulse

CNN

Convolutional Neural Network

CWL

Cognitive WorkLoad

DBSCAN

Density-Based Spatial Clustering of Applications with Noise

DR

Dimensionality reduction

ECG

ElectroCardioGram

EDA

ElectroDermal Activity

EEG

ElectroEncephaloGraphy

EMG

ElectroMyoGraphy

FE

Facial Expressions

FS

Feature selection

FNIRS

Functional near-infrared spectroscopy

GA

Genetic Algorithm

GB

Gradient Boosting

GMM

Gaussian Mixture Model

GSR

Galvanic Skin Response

HMI

Human-Machine Interface

HMM

Hidden Markov Model

KNN

K-Nearest Neighbors

LDA

Linear Discriminant Analysis

LR

Logistic Regression

LSTM
Long Short-Term Memory

ML
Machine Learning

MLP
Multi-Layer Perceptron

MWL
Mental WorkLoad

NN
Neural Network

PCG
Phonocardiogram

PPG
photoplethysmography

RSP
Respiration

RR
Respiration Rate

SPC
Speech sensor

SVM
Support Vector Machine

ST
Skin Temperature

T-SNE
t-Distributed Stochastic Neighbor Embedding

XGB
eXtreme Gradient Boost

Chapter 1

Introduction

In recent decades, there has been a significant increase in research, development, and application of artificial intelligence [1, 2, 3]. Safety in human workplaces is a crucial application field of AI, aiming to enhance and maintain a secure working environment for individuals. Integrating intelligent technologies in workplaces delivers innovative solutions that contribute to accident prevention, risk mitigation, and overall safety [4].
In the context of Human-Machine Interaction (HMI), security development is crucial. Cooperation between humans and machines is widely spread in important fields such as manufacturing and industrial automation, agriculture, healthcare, process control systems, and automotive. The final goal of these interactions is to perform multiple and complex tasks. However, on the other hand, they could lead to an increase in mental workload and stress for the subjects involved [5]. Those two factors can negatively influence the task and increase the risk of accidents, so they should be further investigated.

Mental Workload concerns the cognitive capabilities and effort that an individual needs to dedicate to achieve one or multiple tasks, and the task's complexity influences it, the amount of information processed, and the individual's cognitive abilities. Stress results from preoccupation or mental tension arising from challenging situations, serving as a natural human response that forces us to confront and navigate the challenges and threats in our lives.
Considering the importance of evaluating these cognitive conditions in the HMI sector, the literature has primarily identified three methods for assessing mental workload and stress: subjective questionnaires, behavioural analysis, and physiological signal analysis [6].
Subjective questionnaires are a useful tool but lack real-time applicability. Behavioural analysis is a robust methodology but lacks scalability. The physiological approach, especially with the growth of the biomedical market in recent years, is

gaining increasing interest.
Presently, there isn't a reliable and robust solution that effectively correlates stress and Mental Workload (MWL) with the variations in physiological signals. This gap underscores the need for further research efforts to bridge this connection.
The absence of a reliable solution for correlating stress and Mental Workload (MWL) with physiological signals highlights the need for further research efforts to bridge this gap.
This necessitates comprehensive analyses that consider all relevant physiological signals through a multimodal approach, understanding the interplay between various indicators and their combined impact on assessing stress and mental workload. Additionally, statistical analysis alone has proven insufficient, emphasizing the need to integrate Machine Learning (ML) methodologies. ML techniques offer the potential to uncover patterns and relationships within physiological data that may elude traditional statistical methods.
This study aims to explore the feasibility of creating an artificial intelligence system capable of detecting and classifying different levels of stress and cognitive load levels.

The eLions group at the Politecnico di Torino conducted a collection of physiological signals on 61 different individuals using specific tests: the Stroop test and the N-Back test.
Those trials were chosen as they are established methods for inducing stress and Mental Workload (MWL), respectively, as evidenced in the literature. They were developed specifically to yield a significant amount of data for subsequent analyses, as will be further elucidated. In Figure 1.1, there is a photograph depicting how the tests were conducted.

The ability to identify the cognitive load and stress levels through signal analysis could evolve into a technology that helps prevent or alleviate associated risks during Human Machine Interaction (HMI) with the aid of appropriate sensors. Consequently, this might trigger supplementary emergency systems to enhance both operator safety and the overall surrounding environment.
One of the possible real-world applications of this research is in the aerospace sector. Despite regulations requiring the presence of both a pilot and a co-pilot, in recent years, there have been numerous studies regarding Single Pilot Operations (SPO). This involves the possibility of using a single pilot with advanced safety systems capable of real-time recognition of physical and psychological health conditions, aiming to achieve an adequate level of safety even in the absence of a physically present co-pilot.

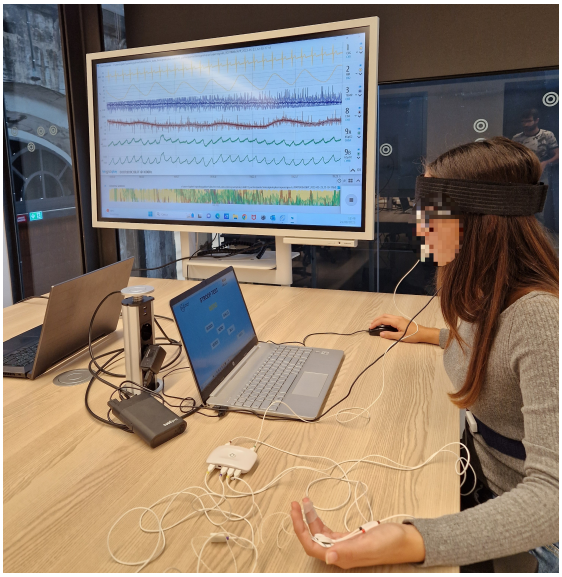

Figure 1.1: BiLoad test conducted by a volunteer.

In the initial phase of this project, the focus lies on data collection and preprocessing. This involves gathering four different datasets containing a substantial number of samples and features.
Once collected, the data undergoes thorough examination to identify and address any quality issues such as missing values, outliers, or inconsistencies. Data cleaning tasks, including normalization, and standardization, are carried out to ensure the dataset's integrity and suitability for analysis.
Following data preprocessing, statistical analysis techniques are employed to explore correlations between features and the target variable to guide subsequent modelling efforts.

Unsupervised algorithms are then employed to uncover patterns, structures, and relationships within the data without the need for labelled outcomes.
Suitable machine learning algorithms are chosen based on the problem type and dataset characteristics. The dataset is split into training, validation, and test sets for model evaluation. Multiple models are trained and cross-validated using various algorithms and hyperparameters to identify the most robust and highest-performing ones.

This thesis is structured as follows. In chapter 2, we provide a comprehensive review of existing literature and research relevant to the topic of our study. Chapter 3 details the materials and methods used in our research, including experimental procedures and data collection techniques. In chapter 4, we present the findings

obtained from our study, accompanied by analysis and interpretation. Finally, in chapter 5, we draw conclusions based on our results and discuss their implications for future research.

1.1 Objectives of the thesis project

Considering the information provided in the preceding section, to achieve the goal of developing a robust, multimodal, and flexible Machine Learning framework, the following objectives have been set:

- Perform a state-of-the-art analysis of machine learning models applied to the multimodal analysis of physiological signals.
- Develop a data management system to process data acquired by the eLions research group, to enhance flexibility and efficiency in subsequent project phases.
- Evaluate the correlation between physiological parameters and the variation in stress conditions and cognitive load, using statistical analysis.
- Conduct unsupervised research to analyze datasets and find patterns or intrinsic structures within the data itself
- Implement intelligent models that classify physiological data into different stress and cognitive load levels.
- Validate the performance of those models using ad hoc tests.

Chapter 2

Background

This chapter thoroughly explores fundamental concepts relevant to our study, establishing a strong basis for understanding the context and challenges. We will begin by investigating the concept of mental workload (MWL) and then examine the dimension of Stress, both of which are central to our research domain.
Subsequently, we will investigate the process of evaluating testing stress and MWL levels, underscoring the intricate dynamics involved in this assessment.
We will then inquire into the current state of research in the field, highlighting key studies and recent advancements that have shaped the existing landscape.
Moving forward, we will explore artificial intelligence algorithms, covering both traditional Machine Learning (ML) approaches and emerging technologies such as neural networks and ensemble learning.
In summary, this introductory chapter serves as a pivotal basis, providing a clear and comprehensive overview of essential concepts and technologies relevant to our study.

2.1 Mental workload

Despite numerous studies conducted on this topic, there is no single definition, universally accepted of mental workload, it largely depends on the specific research context, based on which it will have its origins, mechanisms, implications and measurements.
One of the earliest descriptions of cognitive load was first introduced in the 1940s by Bornemann and then cited by Manzey [7], and its definition was: "Mental workloads are classified as activations related to various performance functions of the human information processing system, forming the basis for the subjective feeling of strain when dealing with primarily cognitive performance requirements (e.g., updating memory contents, problem-solving, monitoring complex systems).

Subsequently, many definitions have been provided, such as mental workload refers to the portion of operator information processing capacity or resources that is actually required to meet system demands by Eggemeier et al.[8].
In scientific literature, various terms like cognitive workload or mental fatigue are used to indicate mental workload. They do not have precise forms and are thus employed as synonyms in this study, as explained by Luzzani et al.[6].

2.2 Stress

Similarly to mental workload, providing a single, unequivocal definition of stress is highly complex, as it varies depending on the study and the field of application, which could be psychological, physiological or behavioural [9].
One of the earliest definitions of stress comes from Hans Selye, who defines it as a nonspecific response of the body to any demand placed upon it [10]. A contemporary perspective on stress may be, according to George Fink [11] in 2016, perception of threat, emotional tension, and difficulty in adjustment that occurs when environmental demands exceed one's perception of the ability to cope.

2.3 Stress and MWL relationship

As previously observed, it's difficult to find a unique definition of Stress and MWL, but it's important to emphasize that the two conditions are different but closely related. One possible understanding of this phenomenon is explained by Debie et al.[12] which models the relationship of these factors into a multidimensional schema with four different components.

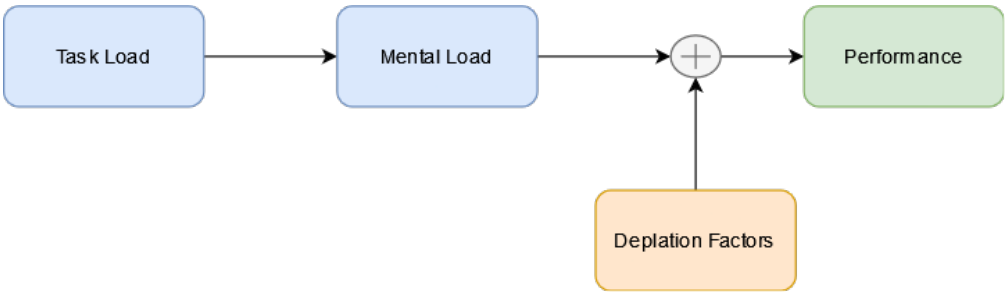

Figure 2.1: Relationship between MWL and Stress

Task load is the amount of effort needed to complete an assigned task. Mental load refers to the quantity of cognitive resources an individual possesses, which can vary based on factors such as genetics, age, health, and experience. Depletion

factors are additional elements that can be either positive or negative, such as stress, fatigue, determination, and mindset. Performance represents the outcome of this equation, reflecting how well the task is executed.

2.4 Statistical tests

Statistical tests are procedures used to make inferences about population parameters from sample data through hypothesis testing. These tests involve comparing sample statistics to population parameters and employing probability distributions to assess the likelihood of observing the data under the null hypothesis, aiding in determining whether there is sufficient evidence to reject or fail to reject a null hypothesis.
The null hypothesis (H0) represents a specific claim or assumption about a population parameter, such as a mean or proportion. Typically, it states that there is no effect, difference, or relationship between variables.
Alternatively, the alternative hypothesis (Ha or H1) represents the opposite claim and is often the focus of researchers' interest.

2.4.1 Analysis of Variance (Anova) Test

Anova is a robust statistical tool used to evaluate whether there are statistically significant differences among the means of three or more groups [13]. It relies on the assumption that observations within each group are independently and identically distributed (i.i.d.) with a common variance and that the group means are the sole sources of variation. The null hypothesis suggests that there are no differences in means across the groups, with any observed discrepancies attributed to random variation.
Anova partitions the total variability in the data into two components: variability between groups (explained variability) and variability within groups (unexplained variability). This partitioning is achieved by estimating the group means and assessing the extent to which the total variability can be ascribed to differences between these means, relative to the residual variability within each group.
F-statistic is the statistic test in Anova, it is calculated as the ratio of explained variability to unexplained variability. A larger F-value indicates that the observed differences between group means are unlikely to have occurred by chance alone, providing evidence against the null hypothesis. The statistical significance of the F-statistic is typically evaluated using the p-value.
The p-value quantifies the probability of observing a test statistic as extreme as, or more extreme than, the one computed from the data, assuming that the null

hypothesis is true. A small p-value suggests strong evidence against the null hypothesis, indicating that the observed differences in means are statistically significant. Conversely, a large p-value suggests that the observed differences could plausibly have arisen due to random variation alone, leading to failure to reject the null hypothesis.

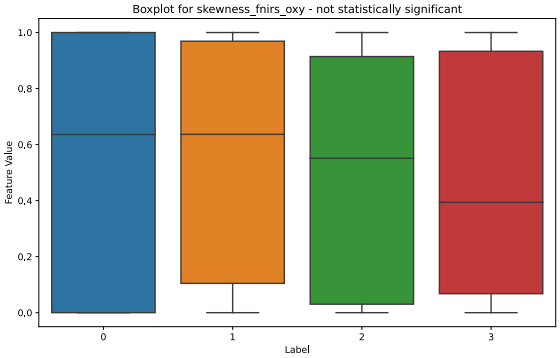

Figure 2.2: Boxplot of a feature statistically not significant

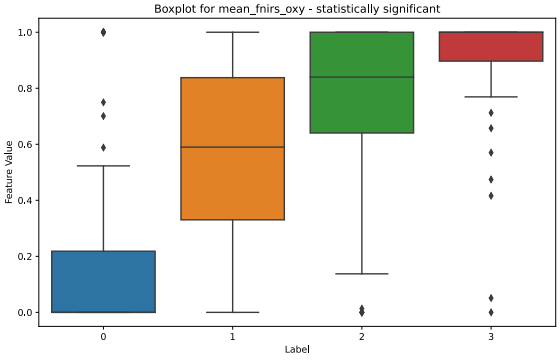

Figure 2.3: Boxplot of a feature statistically significant

Figure 2.2 illustrates the distribution of values across different classes for a feature lacking statistical significance, contrasting with Figure 2.3, which depicts the distribution for a feature that exhibits statistical significance.

2.4.2 Kruskal-wallis test

The Kruskal-Wallis test is a non-parametric method utilized to assess whether significant differences exist among two or more independent groups [14]. It serves as an extension of the Mann-Whitney U test[15], applicable when comparing more

than two groups.
The Kruskal-Wallis test ranks all the data points from smallest to largest, regardless of group membership. It then compares the average ranks among the groups to determine if there are significant differences.
The test statistic, denoted by H, is calculated based on the ranks of the data. It measures the amount of variability between the groups relative to the variability within the groups.
Unlike Anova, which assumes data normality, the Kruskal-Wallis test is non-parametric, making no assumptions about data distribution.
Another notable distinction from Anova lies in the focus of comparison: whereas Anova examines mean differences, the Kruskal-Wallis test scrutinizes disparities in population medians.

2.5 Artificial intelligence

According to Russell and Norvig [16] we can categorize various definitions of AI into four areas based on two dimensions as illustrated in Figure 2.4.

Thinking Humanly	**Thinking Rationally**
"The exciting new effort to make computers think . . . *machines with minds*, in the full and literal sense." (Haugeland, 1985)	"The study of mental faculties through the use of computational models." (Charniak and McDermott, 1985)
"[The automation of] activities that we associate with human thinking, activities such as decision-making, problem solving, learning . . ." (Bellman, 1978)	"The study of the computations that make it possible to perceive, reason, and act." (Winston, 1992)
Acting Humanly	**Acting Rationally**
"The art of creating machines that perform functions that require intelligence when performed by people." (Kurzweil, 1990)	"Computational Intelligence is the study of the design of intelligent agents." (Poole *et al.*, 1998)
"The study of how to make computers do things at which, at the moment, people are better." (Rich and Knight, 1991)	"AI . . . is concerned with intelligent behavior in artifacts." (Nilsson, 1998)

Figure 2.4: Organized definition of Artificial Intelligence

The definitions of thought and reason are found in the top row, while the bottom row pertains to behaviour. The columns differ based on purpose: whether the aim is to faithfully replicate human performance or to strive towards an ideal behaviour

called *rationality*.

In contemporary times, Artificial Intelligence (AI) has multiple branches, each making a unique contribution to the development of new technologies, below, some of them will be listed and briefly explained:

- Machine Learning (ML): ML is a subset of AI that focuses on developing algorithms that enable computers to learn from and make predictions or decisions based on data.
- Deep Learning (DL): Deep Learning, a subset of ML, involves neural networks with many layers, capable of learning complex patterns in large amounts of data.
- Natural Language Processing (NLP): NLP deals with the interaction between computers and humans using natural language. Applications include sentiment analysis, language translation, chatbots, and text summarization.
- Computer Vision (CV): CV focuses on enabling computers to interpret and understand the visual world. Applications include object detection, image classification, facial recognition, medical image analysis, and autonomous drones.
- Generative Models: Generative models are AI algorithms capable of generating new data samples similar to those in the training dataset. Applications include image generation, text generation, and data augmentation for training other models.

In the subsequent sections of this chapter, we will delve into the algorithms employed in this study, starting with an examination of the difference between supervised and unsupervised algorithms. Moreover, supervised algorithms can be further categorized into subgroups, of which we will explore those specifically applied in this project: classical machine learning algorithms, ensemble learning, and deep learning.

2.6 Supervised vs Unsupervised Learning

Machine Learning can be categorized into two main branches: supervised and unsupervised learning.

- **Supervised learning** refers to algorithms that learn from labelled data, meaning they are trained with both feature data and corresponding target labels. The objective is for the algorithms to understand the relationship

between the input and output. With sufficient training data, they can then classify unseen data accurately [16, 17].

- **Unsupervised learning** includes all those algorithms are trained without labelled output categories. The goal of these algorithms is to construct representations of the input data and uncover patterns, aiming to extract meaningful insights or representations from the data [18].

2.7 Unsupervised Learning

We have previously observed that the defining characteristic of unsupervised algorithms is the lack of labelled data. Now, let's explore the strengths and applications of this type of algorithm. According to Ankur Patel [19] unsupervised algorithms excel in scenarios where data patterns are ill-defined or change over time. Additionally, these algorithms are flexible and scalable, and they can be combined with supervised techniques. They also offer benefits in addressing challenging data science issues such as insufficient labelled data, overfitting, the curse of dimensionality, data drift, outlier problems, and data visualization.

2.7.1 T-Distributed Stochastic Neighbor Embedding

t-Distributed Stochastic Neighbor Embedding is an unsupervised learning algorithm invented in 2008 by van der Maaten and Hinton [20] that can high-dimensional data by giving each datapoint a location in a two or three-dimensional space.
t-SNE can incorporate the implicit structure of all data by employing random walks on neighbourhood graphs, thereby affecting the visualization of a subset of the data.
It is particularly beneficial for handling very large datasets due to its ease of optimization and ability to create visualizations that mitigate the tendency to cluster points densely in the centre of the map, a drawback present in some other algorithms.
However, it's worth noting that t-SNE has some limitations. It can be computationally expensive, especially for large datasets, and the results can be sensitive to the choice of parameters and random initialization.
Despite these limitations, t-SNE remains a powerful tool for visualizing and exploring complex datasets in a lower-dimensional space.

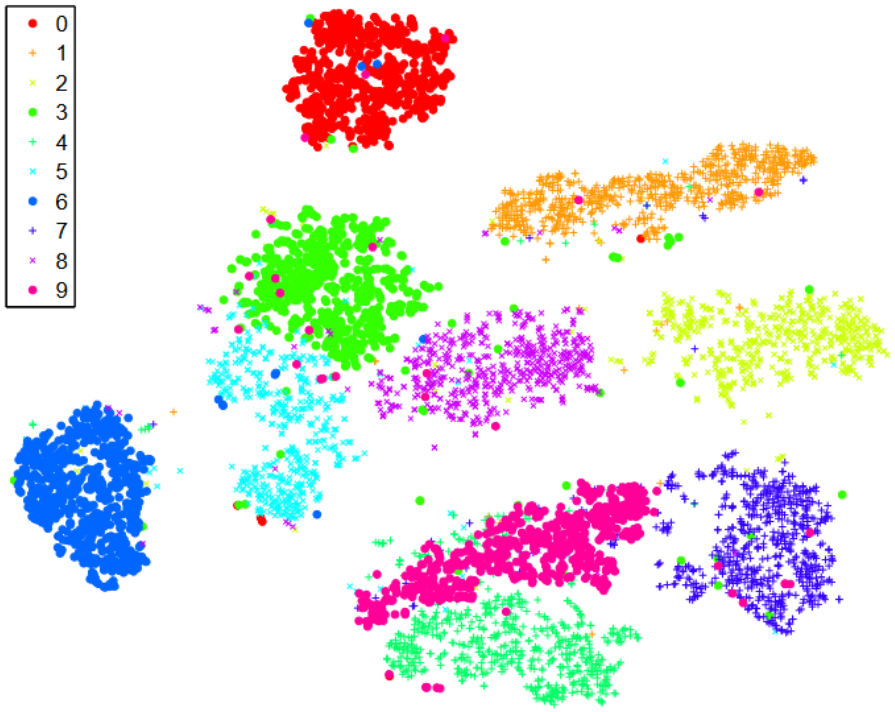

Figure 2.5: Example of t-SNE chosen from the original paper [20]

2.7.2 Density-Based Spatial Clustering of Applications with Noise

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is an unsupervised clustering algorithm introduced in 1996 by Ester, Kriegel, Sander e Xu [21].
Different from other clustering algorithms, DBSCAN can identify clusters of arbitrary sizes and, simultaneously, is capable of detecting outliers.
The algorithm in question employs a density-based approach to estimate the density surrounding individual data points. It accomplishes this by tallying the number of points falling within a designated distance parameter, denoted as ϵ. Subsequently, the algorithm utilizes specified thresholds termed minPts to discern three distinct categories: "core," "border," and "noise" points within the dataset.
Core points are identified based on their density, with each point possessing a sufficient number of neighbouring points within the designated ϵ range. These core points are fundamental in the clustering process.
Upon identifying core points, the algorithm then proceeds to group them into clusters if they satisfy the condition of being "density-reachable." This criterion implies the existence of a contiguous chain of core points, with each consecutive

point residing within the ϵ-neighborhood of the preceding one.
Border points, meanwhile, are points that do not meet the criteria to be considered core points but are located within the vicinity of a core point.
These border points are subsequently assigned to the clusters formed by the core points they are adjacent to.
Finally, the algorithm identifies noise points, which are data points that fail to meet the density requirements for core or border point classification. These noise points are typically treated as outliers within the dataset.
The selection of MinPts and ϵ parameters is critical for accurate clustering. To address this, the elbow method could be employed to optimize these parameters[22]. This method involves plotting the distances to the k-th nearest neighbour against the data points and identifying the point where there is a significant change in slope. This point corresponds to the optimal epsilon value, which determines the neighbourhood radius for density estimation in DBSCAN.

2.7.3 Principal component analysis

Principal Component Analysis (PCA) was invented in 1901 by Pearson during his work on correlation coefficient [23] and then Harold Hotelling further developed the method in 1933 [24], refining its concepts and applications to become more aligned with its modern interpretation and usage.

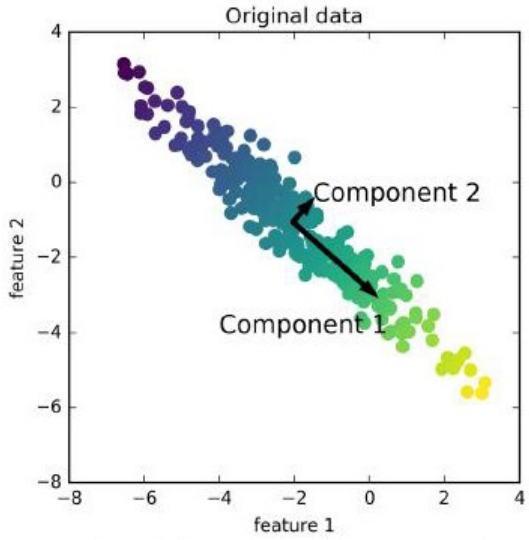

Figure 2.6: Example of PCA applied to 2D data [25].

The primary objective of PCA is to reduce the dimensionality of high-dimensional

data by transforming it into a new set of variables known as principal components. These components capture the majority of the variance present in the data. This algorithm could also be used for data visualization, noise reduction, and data compression. In Figure 2.6, there is a visual example of the application of PCA.

2.8 Supervised Learning

Supervised learning is the predominant approach in machine learning, utilizing labelled data to learn and predict new samples.
The two primary types of supervised learning tasks are regression, where the model learns to predict continuous output values, and classification, where the model learns to assign discrete labels or categories to input data [26]. In the following subsections, we will analyze some of the most common supervised algorithms.

2.8.1 Linear Discriminant Analisys

Ronald A. Fisher, a British statistician and geneticist, introduced Linear Discriminant Analysis (LDA) as a technique to identify a linear combination of features capable of distinguishing between two or more classes of objects or events [27]. Initially, Fisher developed LDA to classify different varieties of flowers based on their measurements. The core principle of Linear Discriminant Analysis (LDA) involves determining a linear combination of features that effectively distinguishes between different classes in the dataset. This is accomplished by maximizing the dispersion between classes while minimizing the dispersion within each class. Essentially, LDA seeks to reduce the dimensionality of the data while retaining the crucial discriminative details that distinguish one class from another. Linear Discriminant Analysis (LDA) is designed to achieve two primary goals: reducing dimensionality by decreasing the number of features or variables under consideration and serving as a classification algorithm. To classify it, it characterises each class's distribution using Gaussian distributions and computes the posterior probabilities of class membership for new instances. Subsequently, LDA assigns class labels to the instances based on these probabilities [26].

2.8.2 K-Nearest Neighbors

K-nearest neighbours algorithm (KNN) is a non-parametric supervised learning method initially developed by Evelyn Fix and Joseph Hodges in 1951 [28]. Thomas Cover and Peter Hart later expanded upon this method [29]. It is commonly employed for both classification and regression tasks.
The central concept of KNN revolves around the similarity between neighbouring

elements. When the algorithm predicts a new label or value, it computes the distance between the new point and all stored data points. Subsequently, it identifies the K nearest neighbours and, through a voting scheme, predicts the label or value for the new point [30].

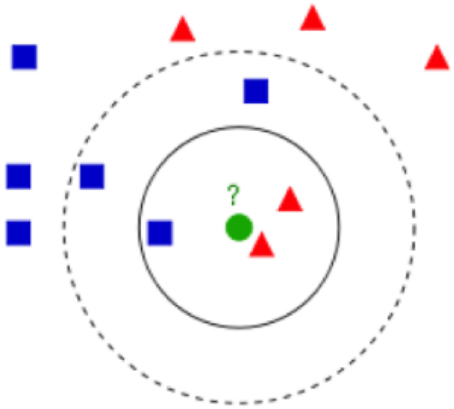

Figure 2.7: An example taken from Prasath et al.[31] of KNN with K=3 as a solid line and K=5 as a dashed line, we can observe that based on the K value the new point could be classified in two different categories.

Selecting the appropriate value for K is crucial; a small value may increase susceptibility to noise, whereas a larger K may result in smoother decision boundaries but lower local accuracy, an example of the importance of this choice can be found in Figure 2.7.
Another significant consideration is properly scaling the features. The objective is to mitigate the influence of varying scales on distance calculations.
Unlike traditional model-based approaches, KNN does not construct a model during training. Instead, it adopts a memory-based approach, retaining all available training data points along with their associated labels or values. Consequently, the computational complexity of KNN grows linearly with the size of the training set.

2.8.3 Support Vector Machine

Support Vector Machine (SVM) was initially theorized by Vapnik and Chervonenkis in 1964 [32] and later developed by Vapnik himself and other colleagues in the 1990s at the AT&T Bell laboratories [33]. SVM is a supervised learning algorithm used in classification and regression tasks and it could be linear or not linear. In

the case of linearly separable data, the SVM algorithm aims to find the optimal hyperplane that effectively divides the two classes while maximizing the margin between them. This optimization task involves solving a mathematical problem to determine the best position and orientation of the hyperplane. The goal is to achieve the widest possible margin, which increases the robustness of the classifier and enhances its ability to generalize to new data points.

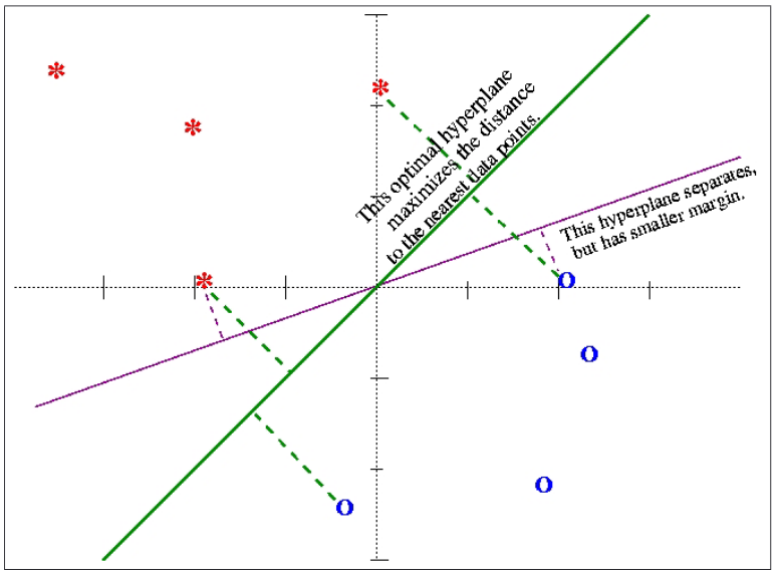

Figure 2.8: An example of linear hyperplane from Dustin Boswell's paper[34]

The kernel trick becomes valuable when dealing with data that is not linearly separable. This technique allows the SVM to effectively handle non-linear patterns by mapping the input features into a higher-dimensional space using a kernel function. By transforming the data into a space where linear separation is feasible, the algorithm can create a linear boundary that effectively separates the classes. This approach is advantageous because it avoids the explicit calculation of the transformation into higher-dimensional spaces, thus maintaining computational efficiency.

The most commonly used kernel functions include the polynomial kernel, Gaussian (RBF) kernel, and sigmoidal kernel. Each of these functions offers different properties and is suitable for different types of data and classification tasks.
Furthermore, the efficiency of SVM is enhanced by its ability to compute the projection of feature vectors using dot products instead of explicitly calculating the transformation into higher-dimensional spaces.

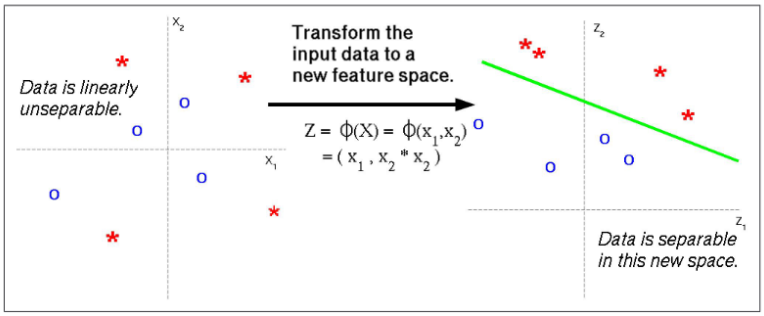

Figure 2.9: An example of kernel trick from Dustin Boswell's paper[34]

This computational strategy allows SVM to efficiently handle large datasets and complex classification problems.
SVM, originally a binary classification algorithm, can be adapted to handle multiclass classification tasks using techniques such as one-vs-one and one-vs-all approaches.
In the one-vs-one approach, multiple binary classifiers are created, each trained to distinguish between a pair of classes. During prediction, the class receiving the most votes from these binary classifiers is selected as the final prediction.
Alternatively, in the one-vs-all approach, SVM trains a separate binary classifier for each class, treating samples from that class as positive examples and samples from all other classes as negative examples. The class with the highest decision score among all classifiers is then assigned to the input sample.
These adaptations enable SVM to effectively address multiclass classification problems by extending its binary classification capabilities.

2.9 Ensemble Learning

Ensemble learning algorithms are a type of supervised learning method in which multiple base learners are trained to address the same problem and then combined to enhance overall performance.
The key distinction between ensemble learning algorithms and traditional machine learning methods lies in their approach to hypothesis generation. Instead of seeking the single best hypothesis, ensemble learning constructs a set of hypotheses and assigns weights to each, effectively creating a classifier through a weighted voting process.

2.9.1 Random Forest

The origins of this ensemble learning algorithm are due to more than one researcher, in particular, Tin Kam Ho [35], Leo Breiman [36, 37], and Adele Cutler[37] contributed actively to the research and development in different stage and methodologies.
To properly understand how Random Forest works, it's fundamental to first comprehend the mechanics of a decision tree. A decision tree is a classifier or regressor that constructs a hierarchical partition of the instance space through recursive splits [38]. Conceptually, it resembles a flowchart structure, where each node corresponds to a decision based on a feature, and each leaf represents the final classification of an instance. These trees are constructed by recursively partitioning the training data into clusters based on the values of features, aiming to create branches that maximize the uniformity of the target labels in each subset.

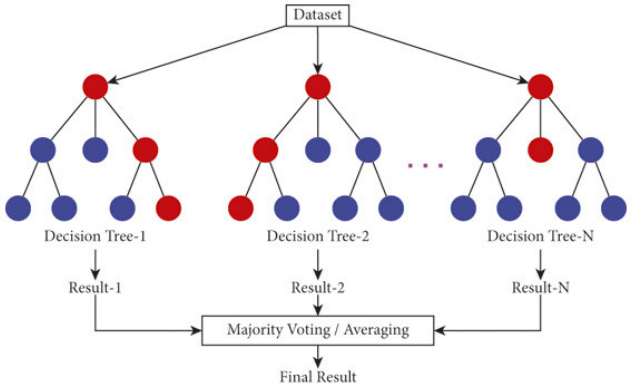

Figure 2.10: Visualization of a Random Forest model [39]

Random forest builds a multitude of decision trees that perform several predictions or regressions, which are aggregated to choose the outcome. These decision trees are all different and are trained only on a subgroup of randomly selected features. This randomness helps to reduce overfitting compared to standalone decision trees. Additionally, it improves the generalization and robustness of the model, being able to leverage the influence of a more assorted set of features.

2.9.2 Adaptive boost

Adaptive Boosting, also known as AdaBoost, is an ensemble learning algorithm designed for classification tasks. It was developed by Yoav Freund and Robert Schapire in 1996 [40][41] .

A weak learner is a model with limited predictive power that performs slightly better than random guessing on a given task, typically they are small decision trees. AdaBoost combines multiple weak learners to create a robust ensemble model capable of effectively handling complex data distributions.

AdaBoost operates through a process that iteratively trains weak learners, each subsequent learner focusing more on the instances misclassified by preceding weak learners. Through this adaptive boosting mechanism, AdaBoost assigns higher weights to misclassified instances, forcing subsequent weak learners to prioritize their correct classification.
One of the key advantages of AdaBoost is its ability to build a strong ensemble model by combining the outputs of these weak learners, effectively leveraging their collective wisdom to overcome individual shortcomings. This ensemble approach enhances predictive performance and improves generalization to unseen data

2.9.3 eXtreme Gradient Boost

Extreme Gradient Boosting (XGBoost) was developed by Tianqi Chen in 2014 [42] and gained considerable attention by winning numerous machine learning competitions. It is available as open-source software in various frameworks.
Similarly to other ensemble learning algorithms, XGBoost combines multiple weak learners (usually decision trees) sequentially to create a strong learner, but it improves the predictive model by minimizing a loss function with a gradient-based method. It is widely adopted due to its scalability and speed, thanks to highly parallelizable computations. Furthermore, it implements regularization techniques like L1 and L2 regularization to prevent overfitting and improve generalization.

2.10 Deep Learning

Deep Learning constitutes a subset of machine learning algorithms that heavily rely on artificial neural networks. The term "deep" denotes the multiple layers of interconnected nodes within these networks [43].
Artificial Neural Networks (ANNs) are composed of artificial neurons, mathematical abstractions inspired by biological neurons. Similar to their biological counterparts, artificial neurons receive input signals, process them, and generate output signals. These neurons are organized into layers and interconnected within and across layers. The learning process in ANNs mirrors that of biological systems, adjusting connection weights based on experiences to enhance performance. However, it's crucial to note that even the most complex ANNs are significantly simpler than the human brain, which contains billions of neurons and trillions of synaptic connections [44]. Despite their relative simplicity compared to the brain, neural networks exhibit remarkable flexibility and can be applied across various domains. They excel in

tasks such as image and speech recognition, natural language processing, robotics, and more.

2.10.1 Multi-Layer Perceptron

In 1957, Frank Rosenblatt introduced the perceptron, a type of single-layer neural network capable of binary classification tasks [45]. The perceptron represented one of the earliest practical implementations of neural network models. After years of refinement and significant advancements, we arrived at the modern-day Multi-Layer Perceptron (MLP) [46, 47, 48, 49].

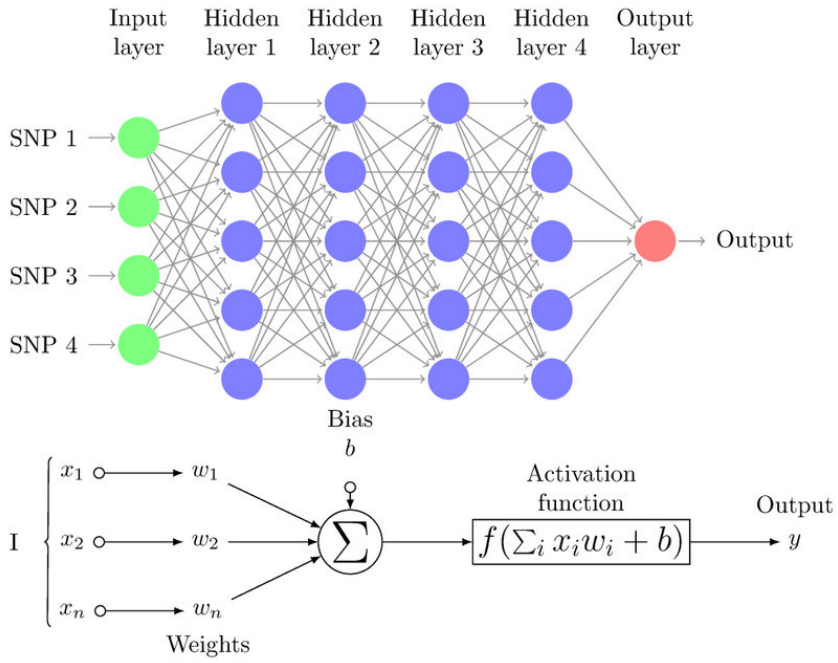

Figure 2.11: Above: diagram depicting a Multi-Layer Perceptron (MLP) architecture featuring four hidden layers. Below: An illustration depicting a fundamental "neuron" model with n inputs, where a neuron is obtained by applying nonlinear transformations to linear combinations of inputs [50].

MLPs are feedforward artificial neural networks composed of interconnected artificial neurons with nonlinear activation functions. They are trained using backpropagation algorithms, which adjust the weights between nodes to minimize the error between predicted and actual outputs. This enables MLPs to effectively

handle nonlinear patterns in data. Figure 2.11 provides a visual representation of both an MLP schema and an artificial neuron.

2.11 State of the art

In this section, the state-of-the-art analysis will focus on the utilization of artificial intelligence (AI) in studying stress and mental workload (MWL), particularly through the analysis of physiological signals.

2.11.1 Biosignals for physiological mental workload and stress evaluation

Extensive research was conducted on scientific papers that examined or suggested methods for assessing mental workload and/or stress through the utilization of physiological signals and artificial intelligence algorithms. Starting from a sample of hundreds of articles, they were found, analyzed, and screened until a subset of 61 papers that dealt with the analysis of mental workload (MLW) and stress using artificial intelligence algorithms on features generated from physiological signals.

Model	**Signals**	**Author**	**N**
SVM	ECG, EDA, EMG	Zhang et al. [51]	9
	ECG, EDA, EMG, RSP	Katsis et al. [52]	-
	EDA	Setz et al. [53]	62
	ECG	Wei et al. [54]	16
	EDA	Ghaderyan et al. [55]	35
	EDA, HR, ST	Romine et al. [56]	7
	ECG, RSP, ST	Ghaderi et al. [57]	7
	ECG, EDA, ST	Kim et al. [58]	50
	BVP, EDA, EYE, ST, SPC	Zhai et al. [59]	32
	ECG, EDA, FE, RSP	Katsis et al. [60]	10
	BVP, EDA, EEG, HRV, RR	Hosseini et al. [61]	15
	EDA, SPC	Kurniawan et al. [62]	10
	PPG	McDuff et al. [63]	10
	EDA, EEG, ST, FE	Sharma et al. [64]	13
	EEG	Hou et al. [65]	9
	EEG, fNIRS	Al-Shargie et al. [66]	22
	PPG	Maaoui et al. [67]	12

Model	Signals	Author	N
SVM	PPG , FE	Giannakakis et al. [68]	23
	EEG	Khosrowabadi et al. [69]	26
	ECG, PCG	Cheema et al. [70]	30
	ECG, EEG	Xia et al. [71]	22
	EDA, EYE	Nourbakhsh et al.[72]	13
	ECG, EDA, SPC	Mijic et al.[73]	40
	ECG, EDA, EEG, EYE, RESP	Barua et al.[74]	66
	ECG, EDA, ST	Gjoreski et al.[75]	23
	EDA, EEG, PPG	Arsalan et al.[76]	28
	ECG, EEG	Pratiher et al.[77]	31
LDA	EDA	Setz et al. [53]	62
	fNIRS	Herff et al. [78]	10
	ECG, EDA, EMG, RSP	Healey et al. [79]	24
	BVP	Nhan et al. [80]	12
	ECG, EDA, Motion	Giakoumis et al. [81]	21
	HRV	Melillo et al. [82]	42
	ECG, EDA, EEG, EMG	Minguillon et al. [83]	10
KNN	EDA, HR, ST	Romine et al. [56]	7
	EMG	Karthikeyan et al. [84]	10
	ECG, EDA, EMG, RSP	Wijsman et al. [85]	30
	EEG	Hou et al. [65]	9
	FE, PPG	Giannakakis et al. [68]	23
	ECG, EDA, ST	Anusha et al. [86]	34
	EEG	Khosrowabadi et al. [69]	26
	EDA, HR, ST	Airij et al. [87]	35
	ECG, EDA, EEG, EYE, RESP	Barua et al.[74]	66
Neural Networks	ST, HR, EDA	Romine et al. [56]	7
	EEG, EDA, ST, FE	Sharma et al. [64]	13
	Pupil, EDA	Pedrotti et al. [88]	33
	ECG, EDA, BVP	Huysmans et al. [89]	12
	EEG	Yin et al.[90]	7
	EDA,EEG,ECG,R	Han et al.[91]	8
	ECG,EDA,RSP	Alic et al.[92]	77
MLP	EEG,EDA,PPG	Arsalan et al.[76]	28
	EEG	Asif et al. [93]	27

Model	Signals	Author	N
LSTM	EEG, ECG, EDA, RSP, HR	Huang et al.[94]	15
CNN	EEG, ECG, EDA, RSP, HR	Huang et al.[94]	15
Dynamic BN	Heart, EYE, Skin, FE	Liao et al. [95]	5
Graph NN	EDA, ST, HR	Lee et al. [96]	80
Fuzzy Network	EDA, ST, HR	Lee et al. [96]	80
	Heart, EYE, Brain, RSP, FE, Voice	Pongsakornsathien et al. [97]	-
	FE, ECG, EDA, RSP	Katsis et al. [60]	10
	HR, EDA, ST	Airij et al. [87]	35
Elman NN	BVP, HR, EEG, EDA, PPG	Khalilzadeh et al. [98]	9
	EEG, BVP, EDA, HRV, RR	Hosseini et al. [61]	15
Probabilistic NN	ST	Karthikeyan et al. [99]	60
LR	EDA, HR, ST	Romine et al. [56]	7
	EEG	Asif et al. [93]	27
	EDA, HR, ST	Gjoreski et al.[75]	23
GenLR	FE, PPG	Giannakakis et al. [68]	23
Random Forest	SPC	Simantiraki et al. [100]	9
	Brain	McKendrick et al. [101]	34
	ST, HR, EDA	Romine et al. [56]	7
	Pupil, EDA	Ren et al. [102]	30
	ECG, EDA, RESP, EEG, EYE	Barua et al.[74]	66
	HR, EDA, ST	Gjoreski et al.[75]	23
	ECG, EEG	Pratiher et al.[77]	31
Naive Bayes	EDA, ST, HR	Romine et al. [56]	7
	Pupil, EDA	Ren et al. [102]	30
	PPG	McDuff et al. [103]	10
	PPG, FE	Giannakakis et al. [68]	23
	EDA,EYE	Nourbakhsh et al.[72]	13
	EEG,EDA,PPG	Arsalan et al.[76]	28
Decision Tree	ST, HR, EDA	Romine et al. [56]	7
	Pupil, FE	Baltaci et al. [104]	11
	SPC, EDA	Kurniawan et al. [62]	10
	HR, EDA, ST	Gjoreski et al.[75]	23

Model	Signals	Author	N
Bayes classifier	EDA, ECG, RSP, EMG	Wijsman et al. [85]	30
ElasticNet	Brain	McKendrick et al. [101]	34
FisherDA	BVP, RR	Nhan et al. [80]	12
	ECG, EDA	Vila et al. [105]	20
	ECG, EDA, RSP, EMG	Wijsman et al. [85]	30
QuadraticDA	ECG, EDA, ST	Anusha et al. [86]	34
AdaBoost	ST, HR, EDA	Romine et al. [56]	7
AdaBoost	FE, EYE	Baltaci et al. [104]	11
XGB	ECG, PPG, RSP, ST	Momeni et al.[106]	24
XGB	ECG, EDA, EEG, RSP, HR	Huang et al.[94]	15
GB	ECG, EEG	Pratiher et al.[77]	31
NCC	EDA	Setz et al. [53]	62
K-Means	EDA, SPC	Kurniawan et al. [62]	10
Clustering	-	Iqbal et al.[107]	-
GMM	EDA, SPC	Kurniawan et al. [62]	10
HMM	SPC	Womack et al. [108]	11
	SPC	Zhou et al. [109]	16
	SPC	Shukla et al. [110]	15
GEE	ECG, EDA, EMG, RSP	Wijsman et al. [111]	30
GA	EDA, EEG, FE, ST	Sharma et al. [64]	13
SelfOrgMap	BVP, ECG, EDA	Huysmans et al. [89]	12

Table 2.1: Comparative Table of machine learning algorithms, signals, and paper citations

In Table 2.11.1, are listed the sixty-one different papers that were collected. The Table also includes the analyzed signals, authors, paper citations and the number of samples treated. All abbreviations referenced in the Table are detailed within the glossary.
Upon analysis of these publications, it is apparent that the physiological signals showing a stronger correlation with variations in stress and mental workload include heart activity, electrodermal activity, temperature, eye movement, brain activity, respiration, facial and speech recognition.

In Figure 2.12, the distribution of different AI algorithms used for classifying states of mental workload and stress is shown. The predominant usage in the classification of stress and cognitive load from biological signals is Support Vector Machine (SVM), followed by Deep Networks, KNN, Linear Discriminant Analysis (LDA), and Bayesian classifiers, as indicated by the graph.

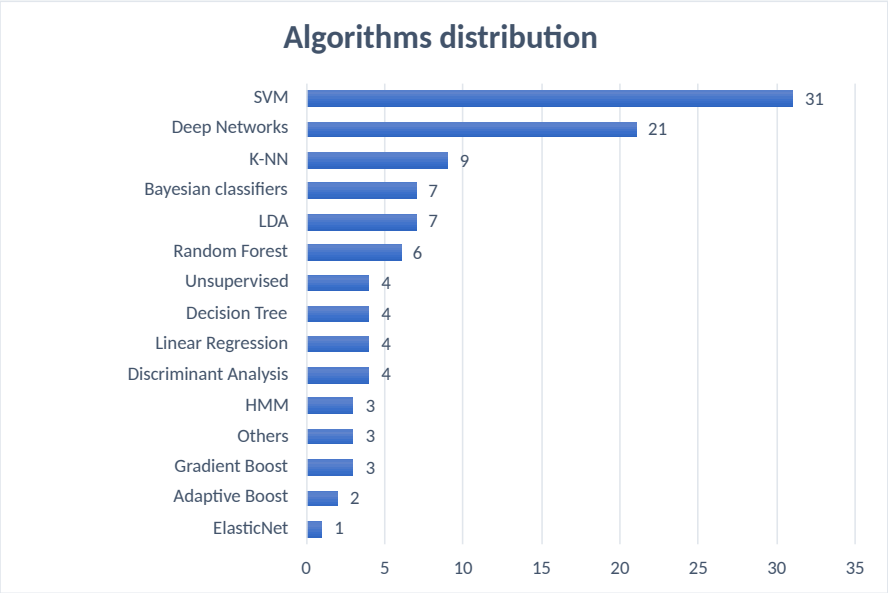

Figure 2.12: Algorithms used for MWL and Stress classification

The aforementioned algorithms have been used to classify stress or cognitive workload, and some of them will be discussed for each of the two categories.
Regarding the CWL stand-alone, there are several ways multimodal classification has been performed. For example, Nourbakhsh et al. [72] used features from electrodermal activity (EDA) and eye blinks. They classified different levels of Mental Workload during arithmetic tasks using SVM and Naive-Bayes algorithms for both the binary and multiclass classification.
Another mixture of different biological signals was employed by Mijic et al. [73] that used arithmetic tasks to induce cognitive load. This analysis integrated the electrocardiogram (ECG) and electrodermal activity (EDA) with paralinguistic speech features. They conducted evaluations on both individual signals and their combined forms, utilizing SVM for classification.
One of the most heterogeneous studies in terms of signals and algorithms was conducted by Barua et al. (2020) within the context of a car driving simulator. They utilized classifiers such as k-nearest neighbour (KNN), support vector machine (SVM), and random forest (RF), integrating data from electroencephalography (EEG), electrooculography (EOG), ECG, respiration, and Galvanic Skin Response (EDA). This data was combined with contextual information from the simulator, including details about vehicles and the driving environment. The study aimed to classify cognitive load ranges by employing various driving tasks, including 1-back and 2-back tasks.
In this context, there was also a challenge proposed by Gjoreski et al [75] where

participants were tasked with implementing a machine learning model capable of binary classification of cognitive load presence in 23 subjects performing the N-Back test, using multimodal features from heart, skin conductance, and skin temperature. Thirteen models were proposed, with SVM and decision trees emerging as the best-performing ones.
Several other examples can be referenced regarding the classification of CWL, employing techniques of varying complexity [106, 90, 112, 91, 94].
Regarding stress assessment and classification, numerous studies can be found in the literature. Sharma et al.[113] examined and reviewed the most reliable sensors and significant features by comparing previous studies, identifying the Support Vector Machine (SVM) as the most accurate classification algorithm.
Alic et al. in 2016 explored the use of more complex algorithms, such as neural networks, to detect elevated stress levels in a sample of 77 individuals. They utilized data from ECG signals, EDA, and respiration for their analysis.
Moreover, in 2019 Arsalan et al. [76] used electroencephalography (EEG), electrodermal activity (EDA) and photoplethysmography (PPG) to classify perceived human stress using SVM, the Naive Bayes classifier, and multi-layer perception (MLP).
A study aiming to render classification more objective was carried out by Iqbal et al.[107], who conducted an analysis comparing various types of unsupervised algorithms to achieve stress level classifications without the use of subjective questionnaires.
One of the most recent and innovative studies was conducted by Pratiher et al. [77] utilized VR gaming as a method to induce stress in patients, extracting multimodal electrocardiogram (ECG) and electroencephalogram (EEG) signals to classify stress induced by gaming difficulty.

As highlighted by the cited studies and past comparative research, there is no single method for inducing and measuring stress and cognitive load, nor is there an algorithm that behaves uniformly. Both of these aspects depend on the type of tests, signals involved, and methodologies used.

2.12 Tests

In psychological and cognitive assessment, various tests are available to evaluate variations in stress and mental workload, which are the two states investigated in this study. The Stroop Color and Word Test were selected to induce and assess stress, while the N-Back Test was chosen to evaluate mental workload.

2.12.1 Stroop test

The Stroop Color and Word Test is a psychological assessment tool used to measure cognitive processing speed and selective attention invented by John Ridley Stroop in 1935 [114].

Figure 2.13: Incongruent ink colours used for Stroop test

During the test, participants are presented with a series of colour words, such as 'red,' 'blue,' or 'green,' printed in incongruent ink colours, for example, the word 'red' written in blue ink as shown in Figure 2.13. The participant's objective is to identify the colour of the ink while disregarding the written word itself.
According to Tulen et al.[115], the Stroop test is a test for the study of stress-induced sympathetic effects, based on psychological, physiological, and biochemical responses.

2.12.2 N-Back test

The N-Back Test is indeed utilized to modulate Mental Workload (MWL) by altering cognitive demand levels. Widely acknowledged in the literature, this test is adaptable across diverse sensory modalities, encompassing both auditory and visual stimuli [116].
Its iterations involve participants in tasks necessitating the recollection of previously presented items within a sequence. Irrespective of variations, the fundamental objective remains consistent: to evaluate and strain working memory capacity by tasking individuals with the accurate recall and identification of items encountered at different stages of the sequence. Figure 2.14 depicts a sequence of letters arranged horizontally, with some of them connected based on the corresponding N-Back.

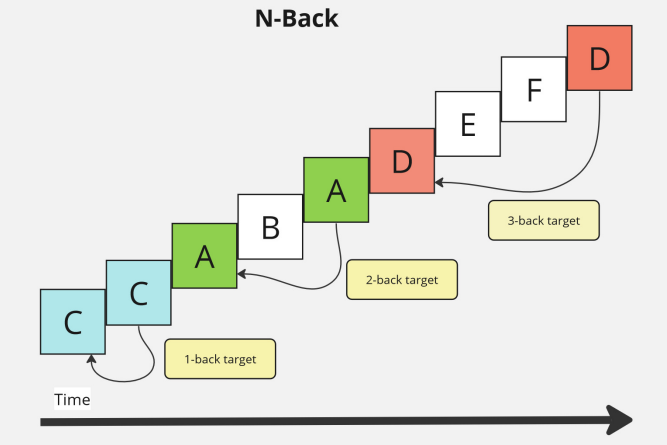

Figure 2.14: Different N-Back stages

2.13 BiLoad project

This study is originated from the BiLoad project of the eLions group at Politecnico di Torino. In earlier stages, the group created a test that induced stress and cognitive load based on the Stroop and N-Back tests. Sixty-one volunteers participated in the study.

2.13.1 BiLoad test

The BiLoad test consists of a Stroop test and three different versions of the N-Back task: visual, auditory, and dual, all performed at three different levels of complexity. During test sessions, all participants were required to follow a standardized protocol for recording various physiological signals while engaging in cognitive tasks.
Before each session, participants were briefed on the study's objectives and asked to provide informed consent.
Before the start of the experiment, participants were equipped with all sensors included in the BiosignalPlux Professional KIT [117], which encompass electrodermal activity (EDA), electrocardiogram (ECG), respiration, temperature, and FNIRS sensors. Additionally, they wore Tobii Glasses 3 to assess eye movements [118].

Before the actual testing phase begins, the user could try two demos to become familiar with the equipment and ensure comprehension of the test procedures. All the tests were previously developed in MATLAB.
Figure 2.15 provides an example of how the Stroop and N-Back tests are displayed.

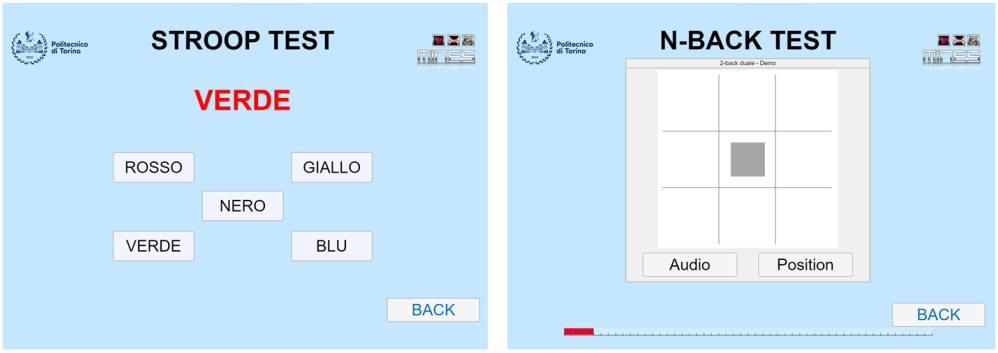

Figure 2.15: Screenshots from Biload test[119]

The test begins with a resting phase called Rest 1, during which the user is asked to remain silent and relaxed for 3 minutes. The first evaluation involves the Stroop test, performed with three levels of difficulty:

- Stroop 1: Words and colours are congruent, and the positions of the buttons remain unchanged.
- Stroop 2: Words and colours are incongruent, and the position of the buttons varies with each response.
- Stroop 3: An auditory distractor is added, randomly pronouncing the names of the colours involved.

Throughout both the Stroop and N-Back tests, a recognizable visual and auditory stimulus is presented in case of correct or incorrect responses. Following the completion of all Stroop tests, a subsequent rest phase (Rest 2) lasting 3 minutes was implemented. Afterwards, the N-Back tests were conducted, also divided into three levels of difficulty (1,2 e 3), each requiring the participant to recall the N previous steps in three different methods:

- Visual N-Back: For each question, a square on a 3x3 grid is coloured, and if the same position was highlighted in the previous N steps, the user must click a "position" button.
- Audio N-Back: A letter is announced audibly, and if the same letter was pronounced in the previous N steps, the user must press the "audio" button.
- Dual N-Back: Both previous stimuli are used simultaneously, requiring the participant to memorize both the positions of the blocks and the letters pronounced in the previous N steps. If either stimulus is repeated in the last n steps, the user must press the correct button: "Audio" or "Position".

Finally, another rest period of 40 seconds is observed (Rest 3). Following the conclusion of both tests, individuals are required to respond to a questionnaire. This questionnaire investigates the extent of difficulty felt by the user concerning stress and cognitive load for each test and its levels of complexity, using a scale comprising three different degrees.

Chapter 3

Materials and Methods

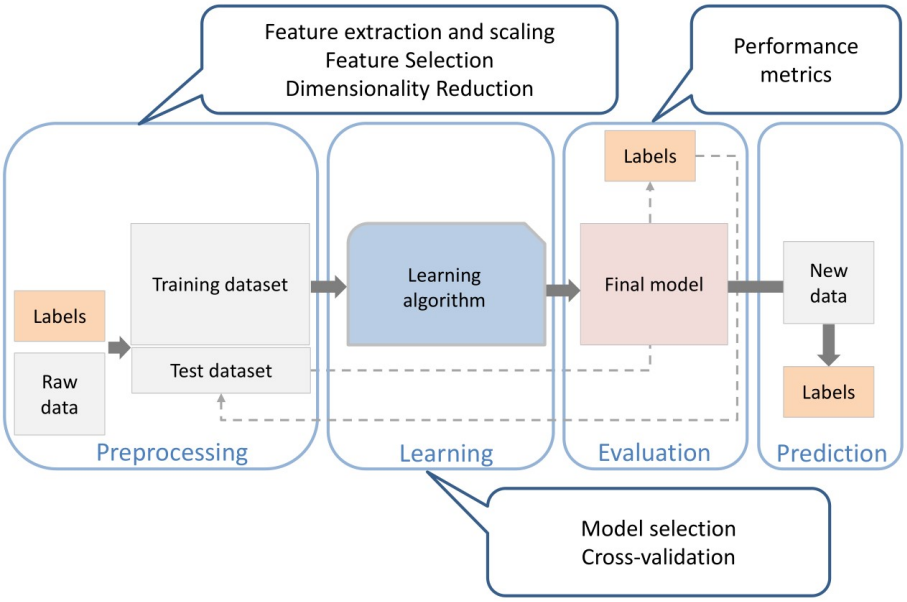

Figure 3.1: Machine learning project workflow [120]

The illustration depicted in Figure 3.1 outlines the standard approach utilized in applying machine learning for predictive modelling, which has been adopted during development. This approach is segmented into four key components: pre-processing, learning, evaluation, and prediction.

The following sections of this chapter will provide a detailed examination of the approaches employed for each stage.

3.1 Preprocessing

Preprocessing in a machine learning project refers to the steps taken to prepare and clean the raw data before feeding it into the ML model.
It involves transforming the data into a format that is suitable for analysis and training. It helps ensure the quality of the data and enhances the performance of the model.

3.1.1 Datasets generation

Starting from the data obtained by the BiLoad project, a Python program was developed to manage and create datasets. The records, stored in .csv files across 61 different folders, correspond to various users on whom tests were conducted. Each user's data includes entries for three different test levels, along with reference values measured during REST 1. Furthermore, in the process of merging the data related to a specific test, the subjective responses provided by the subjects in the questionnaire were merged and used as labels.
These responses were also contained in a CSV file containing all the provided subjective answers. At the end of this process, four datasets were obtained in Pandas DataFrame format, each related to a test: Stroop, Visual N-Back, Audio N-Back, and Dual N-Back. Each dataset contains four instances per subject (one for the rest phase and one for each difficulty stage) resulting in 244 samples, each containing 111 physiological features.

Population

The volunteer population cover ages from 19 to 41 years old, with a mean age of 23.51 years and a standard deviation of 3.19.
In terms of gender distribution, data was collected from 50.8% men and 49.2% women.
The Figure depicted in 3.2 presents a balanced distribution of genders, with a notable concentration of individuals aged between 20 and 29.

Class distribution

As can be seen by observing Figure 3.3, the datasets related to Visual N-Back and Audio N-Back have balanced classes, while the Stroop test dataset exhibits a deficiency in class number 3 and the dual N-Back dataset lacks samples for class 1.

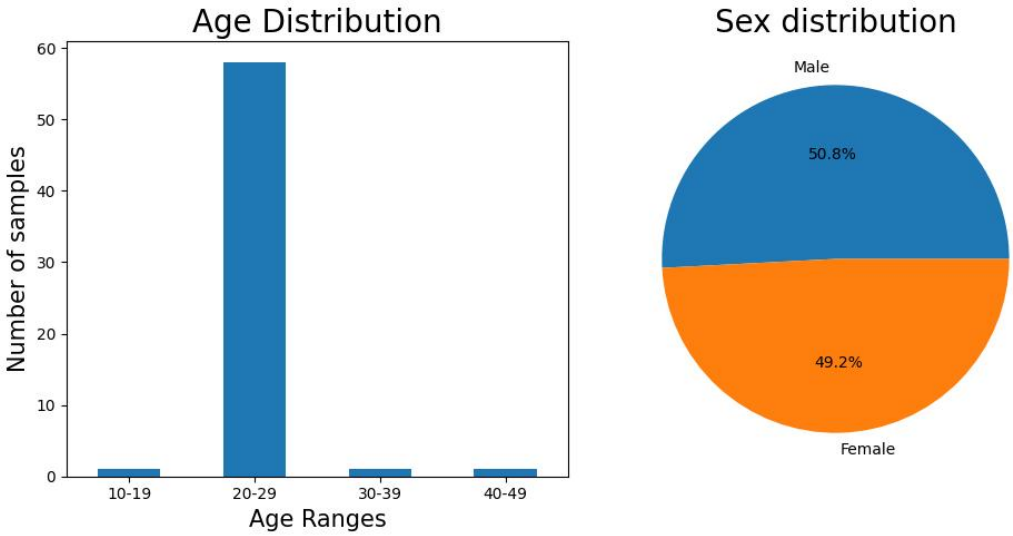

Figure 3.2: Age and gender distribution

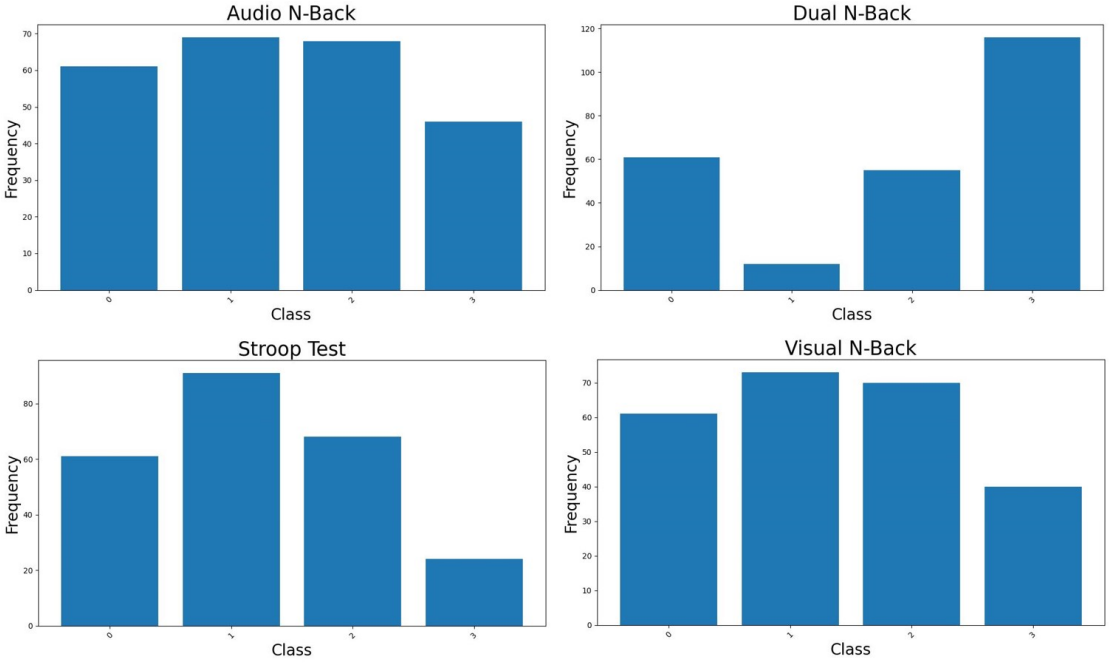

Figure 3.3: Class Distribution

3.1.2 Features

The features contained in the dataset were previously extracted during the development of the BiLoad project. Each feature can be categorized based on the sensor that detected the physiological signal.

Initially, there were six different categories present in the dataset:

- **ECG** [10 feature]: Electrocardiogram features related to the electrical activity of the heart.
- **EDA** [8 feature]: Electrodermal activity, also known as Galvanic Skin Response (GSR). It is a physiological measure that reflects the activity of the sweat glands in the skin, influenced by the autonomic nervous system.
- **TEMPERATURE** [14 feature]: Body temperature, regulated by the body's thermoregulatory system, helps maintain the internal temperature within a narrow range despite changes in external conditions.
- **RESP** [18 feature]: Respiration features, refer to the process of inhaling and exhaling air, vital for exchanging oxygen and carbon dioxide in the body.
- **EYE** [38 feature]: Features extracted by Tobii glasses related to eye movements.
- **FNIRS** [23 feature]: Stands for functional Near-Infrared Spectroscopy. It is a non-invasive neuroimaging technique used to measure brain activity by detecting changes in blood oxygenation levels in the brain.

Appendix A.1 contains a Table with all the features and their acronyms.

3.1.3 Feature scaling

Feature scaling in data analysis refers to the process of adjusting numerical values within a dataset to a consistent scale. This ensures that all variables are comparable and it helps in tasks like visualization and model training. It's important to note that it doesn't alter the distribution of the original data, only its scale.

BiLoad feature scaling

During the project, two different methods of feature scaling have been utilized: min-max normalization and standardization.
Across each dataset, both methods were applied intra-sample, covering every participant's four samples.

Min-max normalization Min-max normalization scales data between 0 and 1 and is commonly used when the distribution of the data doesn't follow a Gaussian distribution. The formula for min-max normalization is:

$$X_{\text{norm}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}}$$

where X is the original value, X_{norm} is the normalized value, X_{min} and X_{min} are respectively the minimum and maximum value in the set.

Standardization Standardization or z-score normalization is a technique that transforms features so they have a mean of 0 and a standard deviation of 1. The formula for standardization is:

$$z = \frac{x - \mu}{\sigma}$$

where z is the standardized value, x is the original value of the feature, μ is the mean of the feature and σ is the standard deviation of the feature. Standardization is especially resilient to outliers and is advised when the data distribution closely approximates a Gaussian distribution.

3.1.4 Feature selection

Feature selection is the process of choosing a subset of pertinent features for constructing models.
The goal of this method is to reduce training time, avoid the curse of dimensionality, remove irrelevant features, and reduce noise.

BiLoad feature selection

In this project, two different tests have been used to evaluate feature importance: the Anova test and the Kruskal-Wallis test. They are both statistical tests used to calculate the p-values of the features.
The p-value, or probability value, is a measure used in statistical hypothesis testing to determine the significance of an observed result. It quantifies the strength of evidence against the null hypothesis.
In both cases, features were selected if they had a p-value lower than a threshold of 0.05; otherwise, they were not included and were not used for model training and evaluation.

3.1.5 Dimensionality Reduction

Dimensionality reduction is a technique in machine learning employed to decrease the number of features in a dataset while preserving essential information.
It is particularly useful when a dataset has a high number of features compared to the number of samples. This process helps in simplifying models, improving algorithm performance, and aiding data visualization.

BiLoad Dimensionality Reduction

In this work, two different approaches have been used: Principal Component Analysis and Linear Discriminant Analysis, both of which have already been introduced in the background chapter (2). Although both can be used as dimensionality reduction techniques, it is important to emphasize that PCA is an unsupervised learning technique, while LDA is a supervised algorithm, meaning it can utilize labels to perform its task.

Feature Selection vs Dimensionality Reduction

Feature selection and dimensionality reduction are two techniques for reducing features, that aim to address the challenges posed by high-dimensional data, which may appear similar. Still, they achieve this goal through different modalities.

Feature selection involves choosing a subset of relevant features while discarding others, whereas dimensionality reduction aims to capture the property of the data in a lower-dimensional space.

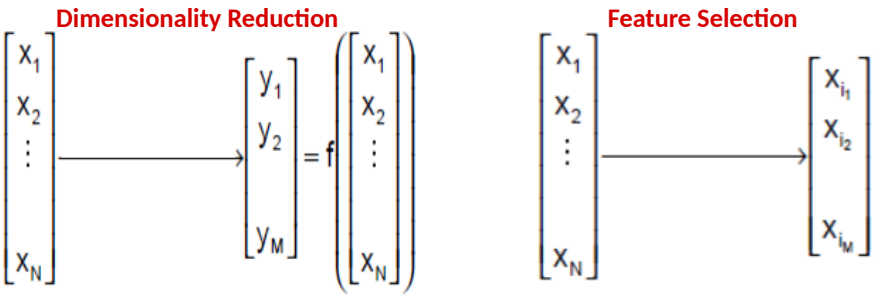

Figure 3.4: Dimensionality reduction and feature selection mapping

3.2 Learning and evalutation

In the learning and evaluation phases, the model is first trained on pre-processed data to learn patterns and relationships between input features and the target variable. This involves selecting an appropriate algorithm, tuning hyperparameters, and minimizing error between predicted and actual outcomes using a training dataset. Subsequently, the model's performance is evaluated using a separate validation dataset to assess its ability to generalize to unseen data. Evaluation metrics such as accuracy, precision, recall, or F1-score are employed to compare the model's performance against predefined criteria

3.2.1 Pipelines

Pipelines are a way to streamline and automate the process of applying a sequence of data transformations followed by a model fitting.
They are especially valuable for organizing and simplifying machine learning workflows, particularly in situations with multiple preprocessing steps.

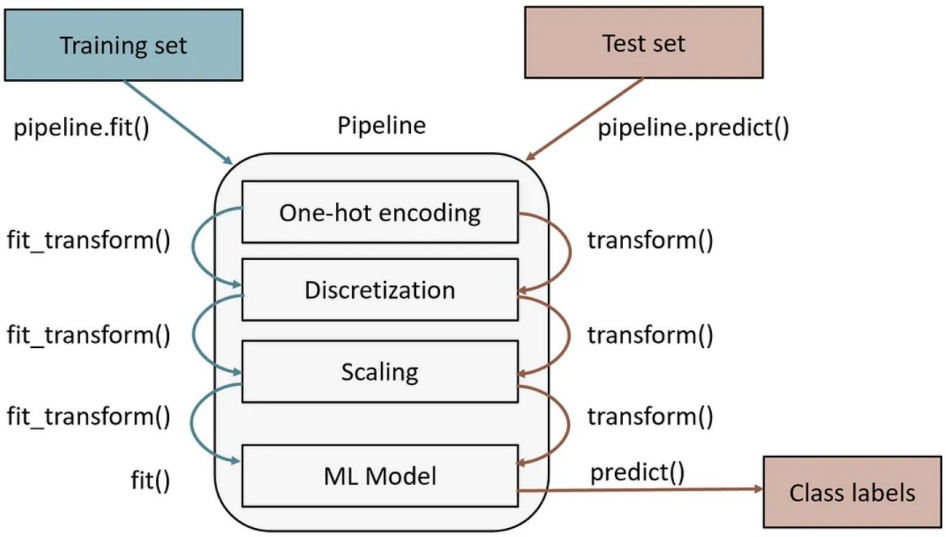

Figure 3.5: Diagram of the Machine Learning Pipeline used on training and test [121].

Pipelines consist of two main components: transformers and estimators.
Transformers are responsible for preprocessing data before it is fed into an estimator.
They handle tasks like scaling features, encoding categorical variables, or generating

new features, effectively transforming the data from one representation to another.

In the training phase, each step in the pipeline, consisting of transformers, should include a method called *fit_transform()*, which both fits the transformer to the data and transforms it. On the other hand, estimators are the actual machine learning models that are trained on the preprocessed data to make predictions. Estimators have a *fit()* method used for training on the provided data, and a *predict()* method for generating predictions on new, unseen data.
Figure 3.5 illustrates a schematic representation where the training phase is depicted on the left side and the testing phase on the right. Pipelines have several advantages: they are efficient, readable and easy to manage.

3.2.2 BiLoad Pipelines

During the development of this research, Scikit-learn pipelines have been utilized [122]. They were chosen for the convenience of being able to employ pre-developed algorithms from the same library, providing optimized and reliable off-the-shelf solutions.
Typically, tasks have been accomplished using pipelines consisting of three stages: feature selection, dimensionality reduction, and ultimately, the implementation of the classification algorithm.

3.2.3 Grid Search

Grid search is a hyperparameter optimization technique commonly used in machine learning to tune the parameters of a model.
It involves searching through a predefined grid of hyperparameters and evaluating the model's performance for each combination of hyperparameters. This process helps identify the optimal set of hyperparameters that result in the best performance for the given dataset and model architecture.

BiLoad grid search

Grid search is implemented in this project by utilizing the GridSearchCV class provided by the Scikit-learn library[122]. This class combines both grid search and cross-validation, making it a powerful tool for hyperparameter tuning.

3.2.4 Cross-Validation

Cross-validation is a statistical technique employed to compare and choose a model for a specific predictive modelling task. It is easy to understand, easy to implement, and results in skill estimates that typically have a lower bias compared to alternative methods.

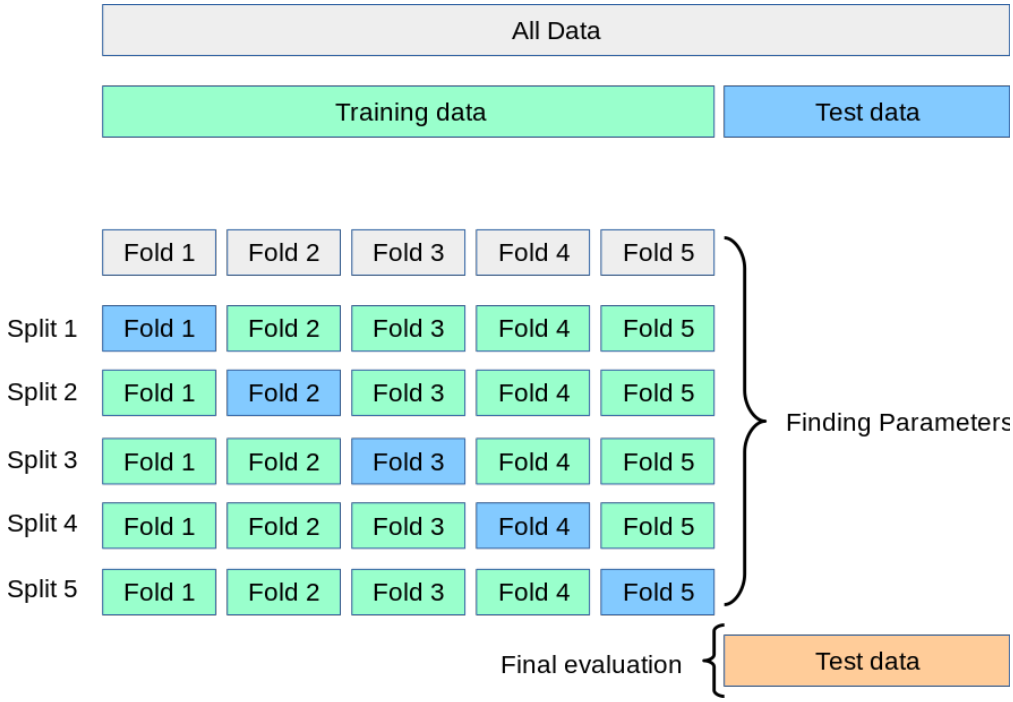

Figure 3.6: An example of 5-fold cross-validation schema [122]

The steps for performing cross-validation will be listed below.

1. Dataset is partitioned into k equal-sized, each fold contains an approximately equal number of samples and ideally represents the overall distribution of the data.
2. The model is trained k times and for each iteration it uses a different fold as a validation set and the remaining as a training set.
3. After each training iteration, the model's performance is evaluated on the validation set using a predefined metric, such as accuracy or F1-score.
4. After completing all iterations, the final performance estimate of the model is determined by averaging the performance scores obtained from each iteration.

Cross-validation aids in reducing the risk of overfitting by offering a more precise assessment of a model's performance on unseen data.
Additionally, it facilitates hyperparameter tuning and model selection by furnishing a more dependable estimate of the model's performance on new, unseen data.

Group-based cross-validation

Group-based cross-validation [123] is a methodology frequently employed in machine learning to assess model performance when working with data exhibiting inherent grouping or clustering characteristics.
Unlike conventional cross-validation where data points are randomly partitioned into training and testing sets, group-based cross-validation acknowledges the presence of groupings in the data, such as temporal or spatial dependencies. This approach aims to prevent data leakage or biased evaluations that may arise from random splitting.
In group-based cross-validation, data points within the same group are either retained together in the training set or the testing set.
This ensures that the model does not inadvertently learn from future data when making predictions on past observations, or vice versa. By maintaining the integrity of group structures during cross-validation, the evaluation of model performance becomes more reliable and reflective of real-world scenarios.

Biload cross-validation

There are several different cross-validation typologies, in our case, two different strategies were adopted: a stratified k-fold cross-validation with k=5 and a leave-one-group-out (LOGO) strategy.
Ideally, the strategy of training the model on all groups except one is theoretically the best approach. However, this is only sustainable in the case of light algorithms, in our case, it is used for binary classification, LDA, KNN and SVM.
Otherwise, when complex algorithms such as ensemble methods or neural networks are employed the use of group-based k-fold has been preferred.

3.2.5 Datasets splits

Dataset split refers to the process of dividing a dataset into multiple subsets for training, validation, and testing purposes.
In this project, our four datasets are divided using a stratified and group-based strategy, allocating 20% to the test set and 80% to the training set. Subsequently,

this split is further employed in a leave-one-group out validation process, where each participant is held out as the validation set while the remaining groups are used for training, allowing for comprehensive evaluation of the model's performance on diverse data subsets.
All these splits have been made in a group-based manner so that data from the same patient cannot be separated during train and test, thus avoiding the risk of data leakage.

3.2.6 Metrics

In machine learning, assessing the effectiveness of an algorithm requires careful evaluation using various metrics such as accuracy, precision, recall, and the F1-score.
Accuracy, the proportion of correctly classified instances among the total, is a fundamental metric calculated by dividing the number of correct predictions by the total number of predictions.

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \tag{3.1}$$

Precision, which focuses on the accuracy of positive predictions, is determined by dividing true positive predictions by the total number of positive predictions made by the classifier.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{3.2}$$

Recall, also known as sensitivity, measures the classifier's ability to correctly identify all positive instances by dividing true positive predictions by the total number of actual positive instances.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{3.3}$$

The F1 score, a harmonic mean of precision and recall, balances both metrics and is particularly useful for datasets with class imbalance.

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.4}$$

BiLoad metrics

In Chapter 4, where the results of various experiments will be reported, performance will be measured using both accuracy and F1-score.

Accuracy is a straightforward and intuitive measure of the overall correctness of the model and it is easily interpretable for stakeholders unfamiliar with technical metrics.
The F1 score, meanwhile, is valuable for addressing class imbalances and provides a balanced evaluation of classifier performance.

3.3 Experiments

Throughout the progression of this project, a variety of experiments and investigations were accomplished, which will be categorized into six different Sections and presented in Chapter 4.
For each element of the list, the methods/algorithms used to solve the task are specified within square brackets.

1. In Section 4.1, we conducted a thorough statistical analysis, employing various techniques to characterize the datasets and the relationships between variables. [Anova,Kruskal-Wallis]

2. Section 4.2 explores unsupervised analysis techniques, such as clustering and dimensionality reduction, to reveal latent structures and groupings within the data without the need for labelled outcomes. [t-SNE, DBSCAN]

3. Section 4.3 delves into supervised learning, where we tackle binary classification tasks. [LDA, KNN]

4. Section 4.4 performs multiclass classification using a variety of algorithms and evaluation metrics. We assess the performance and predictive capabilities of our models in distinguishing between different classes present in the datasets. [LDA, KNN, SVM, RF, ADABOOST, XGB, MLP]

5. In section 4.5, a new task is described where the less significant classes were removed for the Stroop and Visual N-Back datasets, and new supervised algorithms were developed for their classification. [LDA, KNN, SVM, RF]

6. In Section 4.6, additional research will be conducted on the performance of feature combinations derived from the use of one or more sensors in numbers fewer than the six studied previously. [LDA, KNN, SVM]

Chapter 4

Results

This chapter reports the results of our thorough investigation. We used different statistical methods and machine learning techniques to analyze our datasets. Please refer to section 3.3 for general project frameworks and methodology regarding the conducted experiments.

4.1 Statistical analysis

In this section, we perform statistical analysis utilizing Anova (Analysis of Variance) and Kruskal-Wallis tests to assess the significance of features within our dataset. Additional information on these two tests can be found in Section 2.4. Through this analysis, we aim to enhance our understanding of the dataset and make informed decisions based on robust statistical evidence.

4.1.1 Anova test

As mentioned in 2.4.1, Anova is a statistical method employed to identify significant features and rank them accordingly.
In this project we used α=0.05 as a threshold for p-value, it is often chosen as a standard threshold for statistical significance, providing a balance between the risk of Type I errors (false positives) and the sensitivity of detecting true effects.
Below are Tables 4.1, 4.2, 4.3 and 4.4 containing the total number of features and the number of significant features for each signal, using min-max normalization, however, similar results can be achieved by utilizing standardization.

As we can see from the Tables, the Anova test identified 82 significant features for the Stroop dataset, 86 for the Visual N-Back, 83 for the Audio N-Back, and 82 for the Dual N-Back.

Stroop		
Signal	Significant features	Total
ECG	9	10
EDA	3	8
TEMP	9	14
RESP	13	18
fNIRS	28	38
EYE	20	23
ALL	82	111

Table 4.1: Significant features for Stroop dataset using Anova

NBack Visual		
Signal	Significant features	Total
ECG	9	10
EDA	7	8
TEMP	6	14
RESP	17	18
fNIRS	28	38
EYE	19	23
ALL	86	111

Table 4.2: Significant features for Visual N-Back dataset using Anova

NBack Audio		
Signal	Significant features	Total
ECG	9	10
EDA	5	8
TEMP	8	14
RESP	17	18
fNIRS	30	38
EYE	14	23
ALL	83	111

Table 4.3: Significant features for NBack Audio dataset using Anova

NBack Dual		
Signal	Significant features	Total
ECG	9	10
EDA	5	8
TEMP	5	14
RESP	17	18
fNIRS	29	38
EYE	17	23
ALL	82	111

Table 4.4: Significant features for NBack Dual dataset using Anova

Analyzing the relevant features for the various datasets, we can observe that 64 of these are common to all datasets.
Meanwhile, 12 features were analyzed as statistically uncorrelated in all four datasets.
All results and values obtained are presented in the Appendix, divided by signal, and displayed in horizontal bar graphs. The y-axis contains the feature names, while the x-axis represents the p-values, as shown in figure 4.1. Additionally, a dashed vertical line indicating alpha=0.05 serves as the significance threshold.

4.1.2 Kruskal-Wallis test

The Kruskal-Wallis test serves as a non-parametric statistical method to determine if there are significant differences among the medians of three or more independent groups. Its primary utility lies in scenarios where the assumptions of Anova, such as normal distribution and equal variances, are not fulfilled.

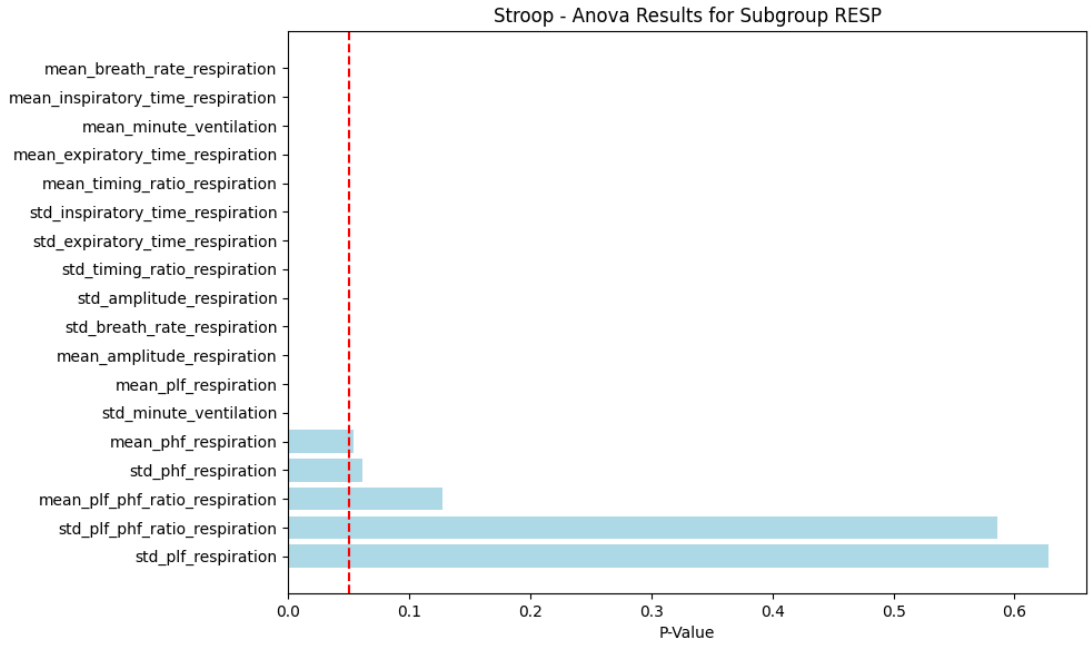

Figure 4.1: Anova Test Results: P-values for Features in respiration Signals

When considering feature importance, particularly in the context of non-parametric methods like decision trees or random forests, the Kruskal-Wallis test can be valuable.
If the Kruskal-Wallis test yields a low p-value for a particular feature, it suggests that the feature may significantly influence the target variable.
This implies that the feature contains valuable information for predicting the target and should be considered important for modelling purposes.
Below are Tables containing the total number of features and the number of significant features for each signal.
The Tables 4.5, 4.6, 4.7 and 4.8 illustrate both the total count of features and the number of features identified as statistically significant for each signal with the Kruskal-Wallis test.

The Kruskal-Wallis test revealed significant features across multiple datasets: 82 for the Stroop dataset, 86 for the Visual N-Back, 85 for the Audio N-Back, and 81 for the Dual N-Back.
Upon close examination of these features, it has been discovered that 65 are shared among all datasets. Additionally, 12 features were statistically uncorrelated across all four datasets.

Stroop		
Signal	Significant features	Total
ECG	9	10
EDA	3	8
T	9	14
RESP	13	18
fNIRS	28	38
EYE	20	23
ALL	82	111

Table 4.5: Significant Stroop dataset features by Kruskal-Wallis

NBack Visual		
Signal	Significant features	Total
ECG	9	10
EDA	7	8
T	6	14
RESP	17	18
fNIRS	28	38
EYE	19	23
ALL	86	111

Table 4.6: Significant Visual N-Back dataset features by Kruskal-Wallis

NBack Audio		
Signal	Significant features	Total
ECG	9	10
EDA	5	8
T	8	14
RESP	17	18
fNIRS	31	38
EYE	15	23
ALL	85	111

Table 4.7: Significant Audio N-Back dataset features by Kruskal-Wallis

NBack Dual		
Signal	Significant features	Total
ECG	9	10
EDA	5	8
T	5	14
RESP	17	18
fNIRS	29	38
EYE	16	23
ALL	81	111

Table 4.8: Significant Dual N-Back dataset features by Kruskal-Wallis

4.1.3 Comparison between Anova and Kruskal-Wallis tests

Analyzing the features through Anova and Kruskal-Wallis tests, it can be observed that 63 features are considered significant and correlated with the class membership for both tests across all four datasets. Furthermore, it is noted that 11 of these features were rejected by both tests across all datasets from the initial set of 111 features.
This evidence suggests consistency in selecting significant features across different statistical testing methodologies, indicating that these 63 features may be particularly relevant for predicting class membership in the considered datasets.
However, the rejection of 11 features across all datasets might suggest that these characteristics may not be as informative or may be influenced by factors not considered in the tests.

As we can see from Table 4.9, the data concerning the Visual N-Back features

	Anova	Kruskal	Common	Rejected
Stroop	82	82	81	28
Visual N-Back	86	86	86	24
Audio N-Back	83	85	83	26
Dual N-Back	82	81	81	28

Table 4.9: Anova vs Kruskal-Wallis feature significance

are the same for both tests, while the data relating to the other datasets show slight differences. However, it was still useful to apply both tests to avoid assuming the Gaussian distribution of the data.

4.2 Unsupervised Analysis

This analysis is fulfilled using unsupervised learning algorithms to study data without knowing the outcome variable or specific labels beforehand.
Dissimilar to supervised learning, where algorithms are trained on labelled data to make predictions, unsupervised learning algorithms work on unlabeled data to find patterns, structures, or relationships within the data.

4.2.1 t-SNE

As previously addressed in Section 2.7.1, t-distributed stochastic neighbour embedding is a popular dimensionality reduction technique commonly used for visualizing high-dimensional data in a lower-dimensional space.
In our case, samples are composed of 111 features and offer a way to project this complex data into a lower-dimensional space while preserving its inherent structure. However, it's important to consider some factors when applying t-SNE to our dataset. Only the 65 features considered significant by both statistical tests across all 4 datasets were utilized for algorithm execution. This approach will reduce noise in visualization and increase efficiency.

Figure 4.2 shows a distinct separation of the rest class compared to the other classes. Additionally, there is an aggregation in the lower part of the plane related to low-level stress samples. The classes related to medium and high levels of stress are instead mixed.
In Figure 4.3, the separation of the rest class is confirmed. However, it is important to note that a 3D representation on a 2D support does not accurately convey spatial relationships. Therefore, for the other datasets, only 2D Figures will be presented, while the 3D ones can be viewed in the Appendix.

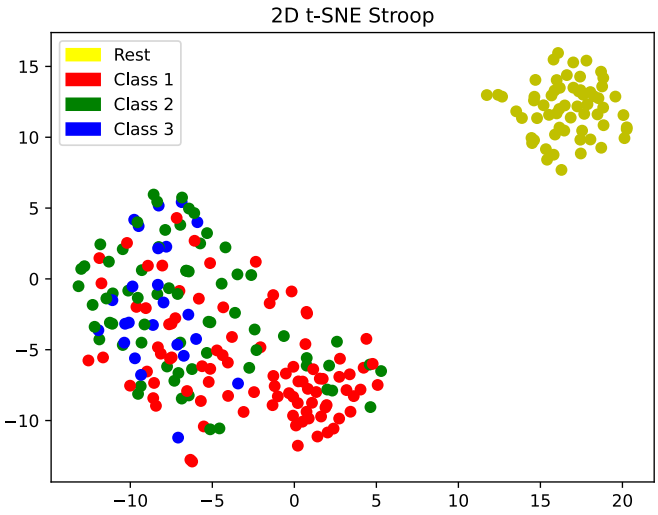

Figure 4.2: 2D t-SNE Stroop dataset

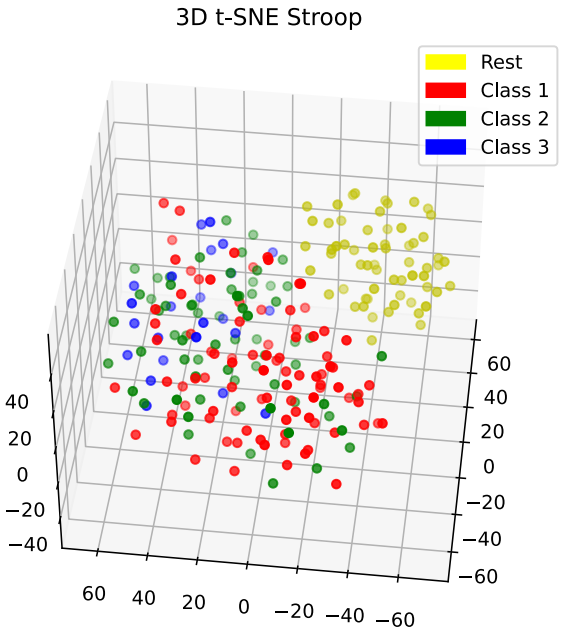

Figure 4.3: 3D t-SNE Stroop dataset

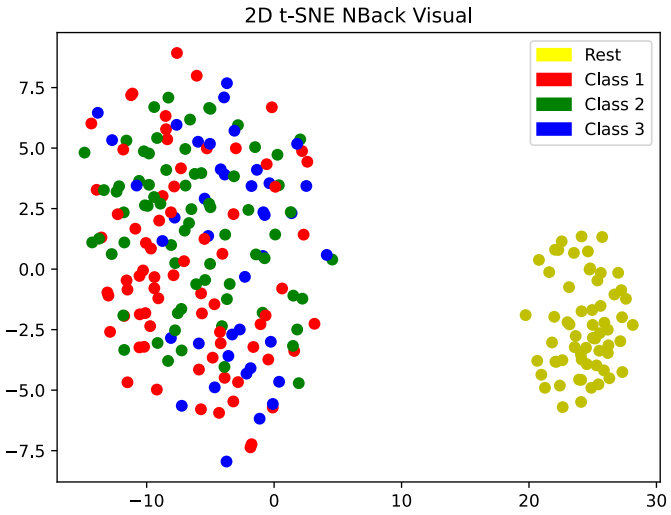

Figure 4.4: 2D t-SNE Visual N-Back dataset

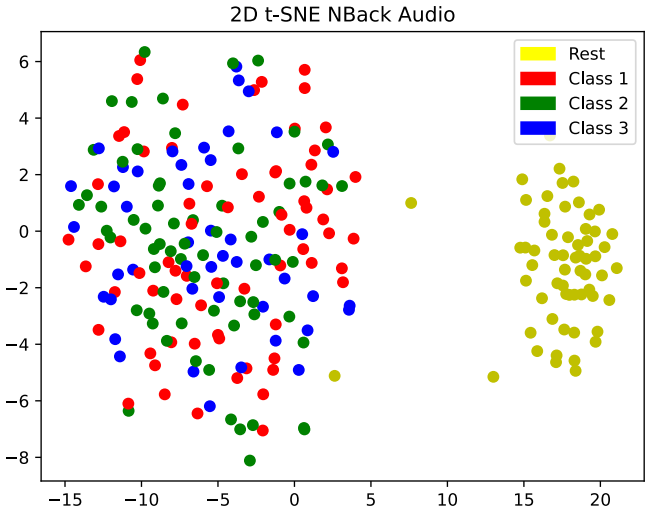

Figure 4.5: 2D t-SNE Audio N-Back dataset

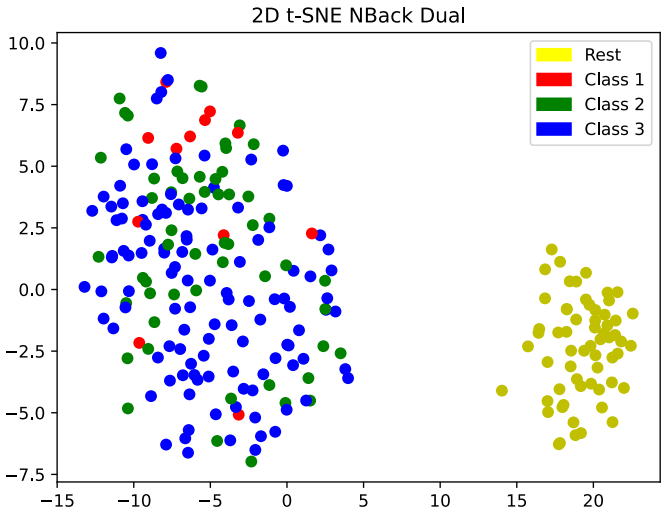

Figure 4.6: 2D t-SNE Dual N-Back dataset

As we can again observe in Figures 4.4 and 4.6 related to Visual and Audio N-Back, the rest class is completely separated from the samples belonging to the classes of altered cognitive load.
Regarding the Audio dataset in Figure 4.6, all rest samples are completely separated, except for two that are slightly close to the other samples.
Following this type of unsupervised analysis, we can anticipate that distinguishing samples related to the resting state from the altered states will be straightforward in future classification tasks.
However, classifying the three classes of altered states in a multiclass manner is expected to be significantly more challenging.

4.2.2 DBSCAN

Following the discussion provided in Section 2.7.2, DBSCAN is an unsupervised clustering algorithm. Optimizing the minPts and ϵ parameters is crucial to obtain good results, and the elbow method, which is also employed in this project, is a common approach for making informed choices. An example of knee-elbow optimization is shown in Figure 4.11.

As depicted in Figure 4.7, for the Stroop dataset, the application of the DBSCAN algorithm identifies only one cluster, while it classifies a group of closely spaced

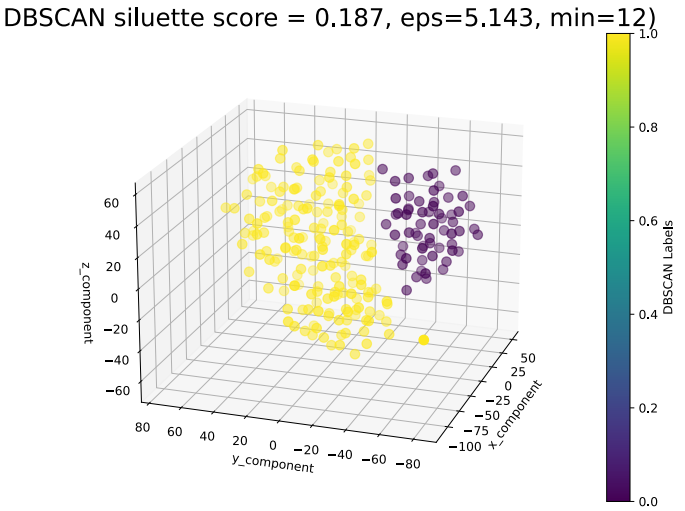

Figure 4.7: DBSCAN applied to Stroop dataset

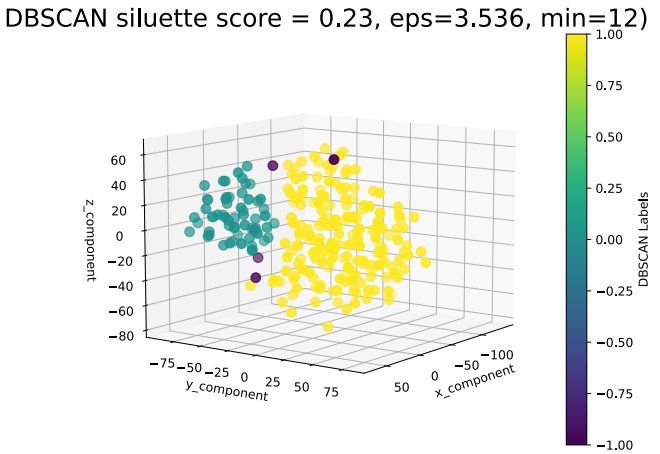

Figure 4.8: DBSCAN applied to Visual N-Back dataset

samples as noise because they do not exceed the threshold of minimum points within the epsilon radius.
Meanwhile, in Figure 4.8, about the Visual N-Back dataset, two clusters are identified, and 5 points are classified as noise.
The same considerations can be made regarding the datasets related to the Audio and Visual N-Back in Figures 4.9 and 4.10, where two clusters are identified, and a handful of points are instead considered outliers.

Apart from the Stroop dataset, where only one cluster was identified by the

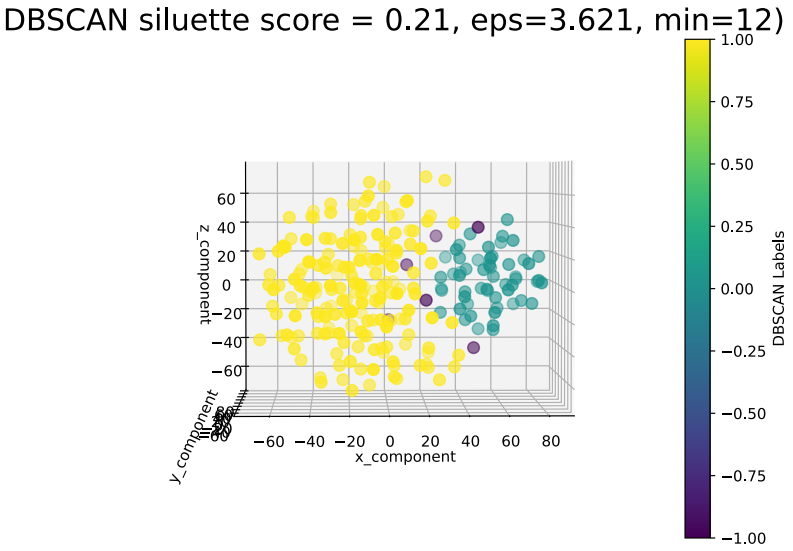

Figure 4.9: DBSCAN applied to Audio N-Back dataset

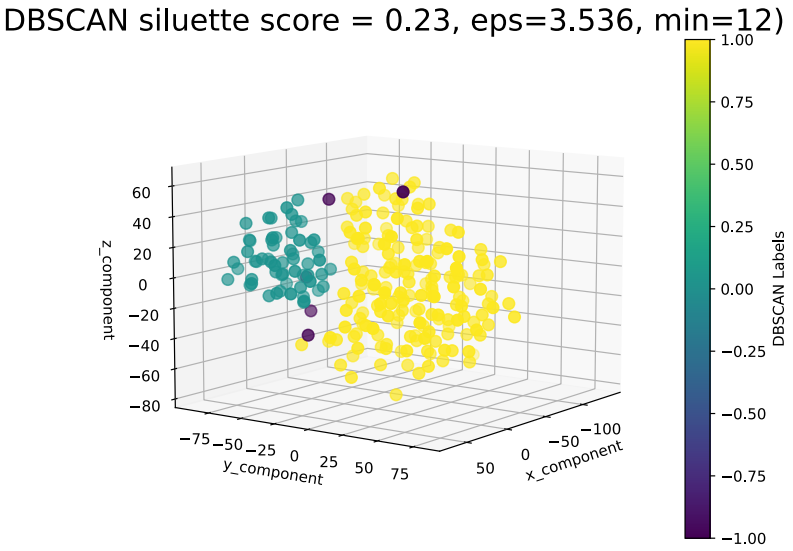

Figure 4.10: DBSCAN applied to Dual N-Back dataset

algorithm but another could be idealized by grouping the noise points, all other datasets have produced two clusters and some outliers.
Combining this information with that related to the t-SNE algorithm, we can hypothesize that our datasets have an intrinsic structure formed by two categories that we can assume to be the rest state and the altered state.
If this preliminary hypothesis were correct, future classification algorithms would easily be able to separate the samples at rest from those in an altered state, but

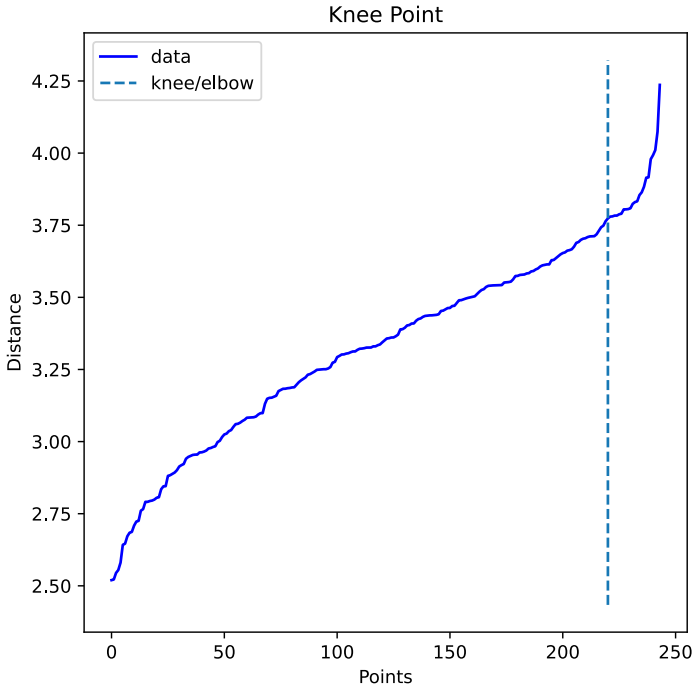

Figure 4.11: Knee-Elbow optimization

they would have more difficulty in identifying the various classes related to different levels of stress and MWL.

4.3 Binary classification

For this preliminary experiment, algorithms will be developed to differentiate between categories associated with rest and those associated with altered states of stress and cognitive load.
To accomplish this objective, all categories related to low, medium, and high states will be aggregated into a unified class.
A low number of algorithms were applied and investigated due to the excellent performance obtained from the simpler algorithms used.
All results refer to precision and F1 score relative to the prediction of the test dataset using the model with the combination of hyperparameters that achieved the highest F1 score during cross-validation with the leave-one-group-out (LOGO) method.

4.3.1 Stroop binary dataset

The binary classification of this dataset involves the ability to differentiate between a state of relaxation and a state of stress, irrespective of its level of severity.
The data was normalized in two different ways: Min-Max and standardized. Three different feature selection possibilities were then used: Anova, Kruskal-Wallis, and no selection. Additionally, two different dimensionality reduction algorithms were applied: PCA and k-PCA (in their versions with Gaussian, polynomial, and sigmoidal kernels). Finally, two different classification algorithms were employed: LDA and KNN.

Table 4.10: Binary classification Stroop dataset

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + LDA	100	100	100	100	100	100
k-PCA + LDA	100	100	100	100	100	100
PCA + KNN	100	100	100	100	100	100
k-PCA + KNN	100	100	100	100	100	100

In Figure 4.10, the results obtained by classifying the Stroop dataset in a binary manner using min-max normalization are collected.
Despite the remarkable results, it is important to emphasize that this outcome had already been predicted during the unsupervised analysis in section 4.2, where the separability between the rest class and all other samples was perfectly noted. For brevity, the Table related to standardization has been omitted as it contained identical results.

4.3.2 Visual N-Back binary dataset

The binary classification of this dataset entails distinguishing between a state of relaxation and a state of cognitive load induced through Visual N-Back, regardless of severity level.

Once again, models capable of achieving 100% classification accuracy on samples have been obtained. Certainly, the problem type and the approach used to perform these tasks contribute to this type of classification success.

Table 4.11: Binary classification Visual N-Back dataset

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + LDA	100	100	100	100	100	100
k-PCA + LDA	100	100	100	100	100	100
PCA + KNN	100	100	100	100	100	100
k-PCA + KNN	100	100	100	100	100	100

4.3.3 Audio N-Back binary dataset

The binary classification of this dataset involves discerning between a state of relaxation and a state of cognitive load induced auditorily via the N-Back task, without consideration of severity level.

Table 4.12: Binary classification Audio N-Back dataset

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + LDA	98.08	97.37	98.08	97.37	98.08	97.37
k-PCA + LDA	98.08	97.37	98.08	97.37	98.08	97.37
PCA + KNN	98.08	97.37	98.08	97.37	98.08	97.37
k-PCA + KNN	98.08	97.37	98.08	97.37	98.08	97.37

The obtained results, illustrated in Table 4.12, are nearly perfect; in fact, all models fail to correctly classify a sample belonging to the rest class but are misclassified as belonging to the altered state class. This indicates how the N-Back Audio dataset is the most complex to analyze and classify.

4.3.4 Dual N-Back binary dataset

The primary objective of this study involves distinguishing between states of relaxation and cognitive load induced by the N-Back task. This classification is conducted using auditory and Visual stimuli while disregarding severity levels.

In Table 4.13, the results obtained for this dataset are indicated, which once again, just like Stroop and Visual N-Back, are equal to 100%.

Table 4.13: Binary classification Dual N-Back dataset

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + LDA	100	100	100	100	100	100
k-PCA + LDA	100	100	100	100	100	100
PCA + KNN	100	100	100	100	100	100
k-PCA + KNN	100	100	100	100	100	100

4.4 Multiclass classification

This section will dig into the outcomes of multiclass classification.
To achieve this objective, categories representing low, medium, and high cognitive states, obtained from subjective questionnaires will be delineated into distinct classes, forming a multiclass classification task. Additionally, the class related to rest will be included in the classification scheme.
Given the complexity of the task, a focused set of simpler and more complex algorithms will be applied and evaluated.
The evaluation will be carried out on the test dataset using the model with hyper-parameters optimized for the highest F1 score during cross-validation.
For cross-validation, the leave-one-group-out (LOGO) method will be exclusively utilized for lighter algorithms such as LDA and KNN, while for heavier algorithms, a grouped 5-fold strategy will be applied. Both techniques ensure the model's robustness and generalizability across different cognitive states.
The best results will be bolded and underlined to channel the reader's attention.

4.4.1 Linear Discriminant Analysis

The results obtained using LDA preceded by dimensionality reduction through PCA and k-PCA, with features selected through Anova and Kruskal-Wallis tests, are presented in Tables, two for each dataset.
Each Table represents the chosen normalization for experimenting: min-max or standardization.

Stroop dataset

In Table 4.14 and 4.15, the results for the Stroop dataset are presented. The best outcomes under both types of normalization are achieved using k-PCA and Kruskal-Wallis as feature selection algorithms, yielding an accuracy rate of up to 73%. However, the corresponding F1 score falls below 60%.
It's worth noting that the performance is slightly better when using Standardization.

Table 4.14: Stroop - LDA - Min-Max Normalization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + LDA	67.31	59.89	65.38	53.16	67.31	54.58
k-PCA + LDA	67.31	59.56	65.38	53.16	**69.23**	56.25

Table 4.15: Stroop - LDA - Standardization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + LDA	65.38	53.75	69.23	56.25	71.15	57.87
k-PCA + LDA	61.54	51.28	71.15	57.39	**73.08**	58.94

Visual N-Back dataset

In Table 4.16 and Table 4.17, the outcomes for the Visual N-Back dataset are exhibited.
As observed with the Stroop dataset, performance related to Standardization is better than min-max normalization. The best combination achieved almost 77% of correctly predicted samples with an F1 score nearly equal to 74%.

Table 4.16: Visual N-Back - LDA - Min-Max Normalization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + LDA	71.15	68.53	69.23	67.75	67.31	65.24
k-PCA + LDA	71.15	68.53	67.31	66.96	69.23	68.15

Table 4.17: Visual N-Back - LDA - Standardization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + LDA	71.15	69.58	73.08	67.78	75.0	72.37
k-PCA + LDA	71.15	65.94	**76.92**	73.98	73.08	69.93

Audio N-Back dataset

As previously hypothesized, the Audio N-Back dataset is the most complex to classify.
In Figure 4.18 and 4.19, it can be observed that at most an accuracy and an F1 score lower than 66% are achieved.

Table 4.18: Audio N-Back - LDA - Min-Max Normalization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + LDA	63.46	63.01	63.46	60.82	59.62	58.76
k-PCA + LDA	65.38	64.21	61.54	58.95	59.62	58.76

Table 4.19: Audio N-Back - LDA - Standardization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + LDA	63.46	61.96	65.38	65.2	**65.38**	65.65
k-PCA + LDA	59.62	57.39	57.69	55.69	59.62	58.74

Dual N-Back dataset

The Dual N-Back yields the best results among all datasets, achieving an accuracy of 80% and a nearly equivalent F1 score, as depicted in Figures 4.20 and 4.21.
Once again, standardization outperforms min-max normalization.
The confusion matrix depicted in Figure 4.12 illustrates the performance of the best model derived from this dataset.
While the model demonstrates good metrics, it is noteworthy that the matrix's effectiveness is compromised by a significant class imbalance.
Specifically, there is a pronounced disproportion towards the category associated with high cognitive load, coupled with a scarcity of instances attributed to the low class.

Table 4.20: Dual N-Back - LDA - Min-Max Normalization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + LDA	69.23	53.42	69.23	53.82	67.31	53.11
k-PCA + LDA	73.08	56.25	69.23	54.85	71.15	64.59

Table 4.21: Dual N-Back - LDA - Standardization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + LDA	75.0	57.75	**80.77**	77.46	78.85	76.44
k-PCA + LDA	73.08	55.54	73.08	56.65	75.0	57.94

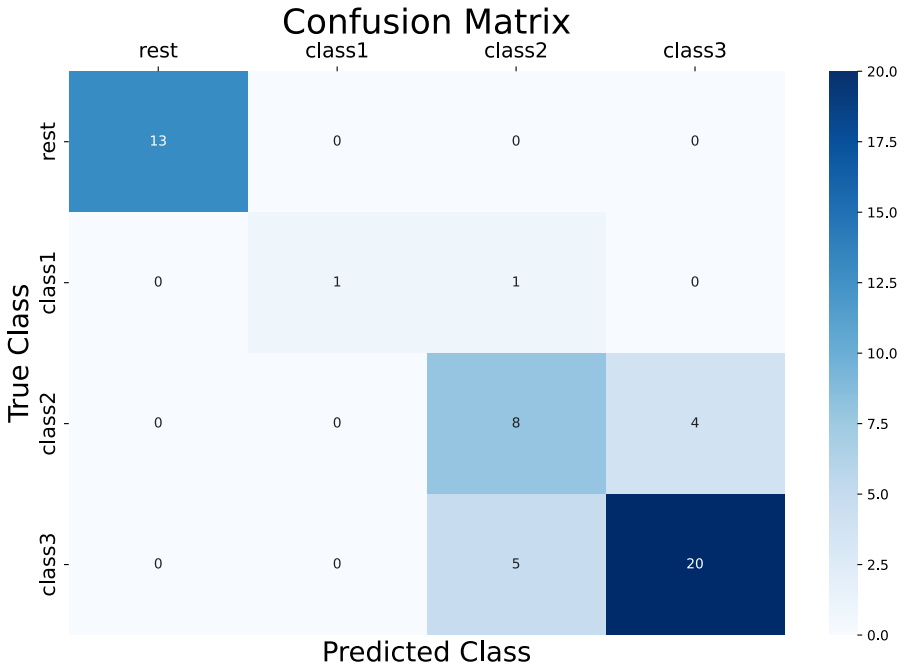

Figure 4.12: Confusion matrix for PCA + LDA with Kruskal-Wallis as feature selection on Dual N-Back dataset

4.4.2 K-Nearest Neighbors

The outcomes derived from employing K-NN following dimensionality reduction via PCA, k-PCA and LDA, utilizing features selected through Anova and Kruskal-Wallis tests, are showcased in pairs of Tables for each dataset.
Each Table delineates the used normalization method for experimentation: either min-max scaling or standardization.
The results are the benchmarks obtained on the test set by the best combination of hyperparameters after cross-validation.

Stroop dataset

As can be observed in Tables 4.22 and 4.23, better results are obtained for the first time using min-max normalization compared to standardization.
The best model achieves an accuracy of 75% and an F1 score of 70% in the combination utilizing LDA as dimensionality reduction and Anova as feature selector. Those results are slightly inferior to the combinations obtained on the same dataset by the LDA algorithm.

Examining the confusion matrix depicted in Figure 4.13 for this algorithm is interesting. It offers insights into the model's proficiency in predicting rest and low cognitive load accurately, while also highlighting challenges it faces in classifying medium and high states.

Table 4.22: Stroop - KNN - Min-Max Normalization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + KNN	71.15	62.89	65.38	52.53	61.54	55.29
k-PCA + KNN	67.31	58.03	65.38	52.86	63.46	57.01
LDA + KNN	63.46	51.67	**75.0**	70.23	67.31	59.03

Table 4.23: Stroop - KNN - Standardization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + KNN	61.54	50.85	69.23	60.65	67.31	53.09
k-PCA + KNN	63.46	52.84	**71.15**	64.57	63.46	50.78
LDA + KNN	65.38	53.76	69.23	56.14	67.31	54.52

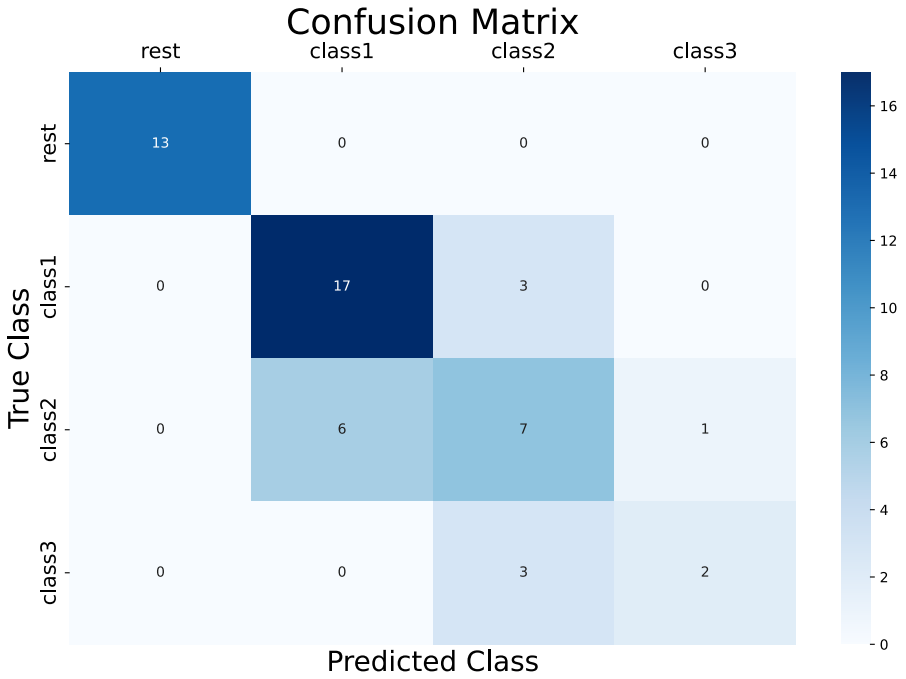

Figure 4.13: Confusion matrix for LDA + KNN with Anova as feature selection on Stroop dataset

Visual N-Back dataset

Using the K-NN algorithm and the aforementioned combinations of elements in the pipelines, the results shown in Tables 4.24 and 4.25 are inferior compared to those obtained previously, with accuracy peaks reaching only 70%.
Even the F1 score deviates significantly from the accuracy values, making the models unattractive.

Table 4.24: Visual N-Back - KNN - Min-Max Normalization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + KNN	55.77	51.94	65.38	59.46	63.46	60.44
k-PCA + KNN	61.54	56.34	61.54	56.39	**69.23**	62.53
LDA + KNN	65.38	63.53	63.46	62.36	61.54	58.7

Table 4.25: Visual N-Back - KNN - Standardization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + KNN	57.69	53.72	**71.15**	63.87	63.46	57.35
k-PCA + KNN	65.38	61.97	57.69	55.52	67.31	63.02
LDA + KNN	57.69	54.22	65.38	65.02	67.31	65.4

Audio N-Back dataset

Similar to the Visual N-Back, the performance for this dataset does not improve compared to the previous results; in some combinations, there are poor outcomes, as shown in Tables 4.26 and 4.27.
It seems probable that this algorithm encounters challenges in modelling this particular type of problem.

Table 4.26: Audio N-Back - KNN - Min-Max Normalization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + KNN	44.23	43.1	55.77	54.81	53.85	51.37
k-PCA + KNN	40.38	39.69	57.69	50.87	55.77	54.99
LDA + KNN	**65.38**	64.68	57.69	56.9	63.46	61.79

Table 4.27: Audio N-Back - KNN - Standardization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + KNN	53.85	52.75	55.77	54.75	53.85	50.76
k-PCA + KNN	51.92	50.56	55.77	54.05	55.77	55.27
LDA + KNN	53.85	52.94	63.46	63.67	63.46	63.67

Dual N-Back dataset

As can be seen from Tables 4.28 and 4.29, once again, the performance is not superior to that achieved previously, although the combination of results obtained

through PCA + KNN with feature selection via Anova presents interesting results, achieving 80.77% accuracy and 75.43% in F1 score.
Another time, the results obtained through min-max normalization are superior to those obtained with standardization.

Table 4.28: Dual N-Back - KNN - Min-Max Normalization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + KNN	69.23	52.65	**80.77**	75.43	73.08	56.33
k-PCA + KNN	71.15	54.54	75.0	56.0	63.46	50.91
LDA + KNN	69.23	61.53	75.0	64.98	76.92	59.2

Table 4.29: Dual N-Back - KNN - Standardization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + KNN	75.0	72.67	67.31	50.98	69.23	53.0
k-PCA + KNN	76.92	74.52	75.0	56.61	71.15	70.55
LDA + KNN	69.23	55.3	73.08	55.96	71.15	53.28

4.4.3 Support Vector Machine - One vs All

The experimental setup involved applying SVM post-dimensionality reduction through PCA, k-PCA, and LDA, incorporating features selected via Anova and Kruskal-Wallis tests.
As previously explained in subsection 3, SVM is a binary classification algorithm that can be extended to multiclass using two methods: One-Vs-One and One-Vs-All. In this subsection, the outcomes achieved via the One-vs-All approach will be delineated and scrutinized for their superior accuracy and adaptability, while outcomes arising from the One-vs-One method will be reported in the Appendix C.2.1.
The outcomes are presented in pairs of Tables for each dataset, illustrating the chosen normalization technique for experimentation: either min-max scaling or standardization.
The presented results represent the performance benchmarks achieved on the test set by the optimal combination of hyperparameters determined through cross-validation.

Stroop dataset

As shown in Tabels 4.30 and 4.31, the performances concerning the two normalizations are similar, but the best model is obtained using LDA as dimensionality reduction and without the use of feature selection.
Despite this model achieving 71% accuracy and almost 67% F1 score, it does not improve the performance of the models analyzed previously for this dataset.

Table 4.30: Stroop - One vs All - Min-Max Normalization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + SVM	63.46	57.37	69.23	61.45	67.31	59.65
k-PCA + SVM	71.15	57.85	61.54	55.24	71.15	57.87
LDA + SVM	69.23	63.6	71.15	57.28	69.23	62.93

Table 4.31: Stroop - One vs All - Standardization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + SVM	65.38	54.82	67.31	64.64	63.46	58.23
k-PCA + SVM	67.31	64.26	63.46	56.52	69.23	64.68
LDA + SVM	**71.15**	66.93	61.54	51.73	69.23	62.19

Visual N-Back dataset

The results are presented in Tables 4.32 and 4.33.
Through the combination of kernel-PCA and Anova feature selection, a result is achieved where the accuracy reaches nearly 79% and the F1 score reaches 76%. Once again, upon analyzing the confusion matrix, it can be noted that the major issue concerns the classification of the class related to high levels of cognitive load. Positively, these results represent the best outcomes obtained for the Visual dataset up to this point, enabling the correct classification of 3 out of 4 samples.

Table 4.32: Visual N-Back - One vs All - Min-Max Normalization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + SVM	71.15	68.78	75.0	72.52	73.08	66.79
k-PCA + SVM	69.23	66.8	75.0	71.79	71.15	68.99
LDA + SVM	63.46	61.75	59.62	54.87	63.46	60.22

Table 4.33: Visual N-Back - One vs All - Standardization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + SVM	71.15	68.99	71.15	66.51	75.0	71.53
k-PCA + SVM	69.23	67.09	**78.85**	76.19	76.92	72.88
LDA + SVM	15.38	6.67	65.38	64.21	67.31	66.31

Audio N-Back dataset

The performance shown in Tables 4.34 and 4.35 particularly underlines the difficulty that SVM faces when the dimensionality is reduced via LDA. However, through Kruskal-Wallis and k-PCA, results are obtained that approach 68% in both metrics, the best model achieved for this dataset so far.
This model has obtained the confusion matrix shown in Figure 5, where can be observed the correctness of the classification of the rest class and the issues related to the classes of altered cognitive load.

Table 4.34: Audio N-Back - One vs All - Min-Max Normalization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + SVM	59.62	59.0	63.46	63.55	61.54	62.13
k-PCA + SVM	63.46	64.12	57.69	58.51	57.69	58.22
LDA + SVM	63.46	63.69	61.54	61.19	65.38	63.23

Table 4.35: Audio N-Back - One vs All - Standardization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + SVM	55.77	55.64	67.31	67.3	59.62	59.95
k-PCA + SVM	59.62	59.54	65.38	65.78	**67.31**	67.97
LDA + SVM	25.0	10.0	50.0	48.91	55.77	53.98

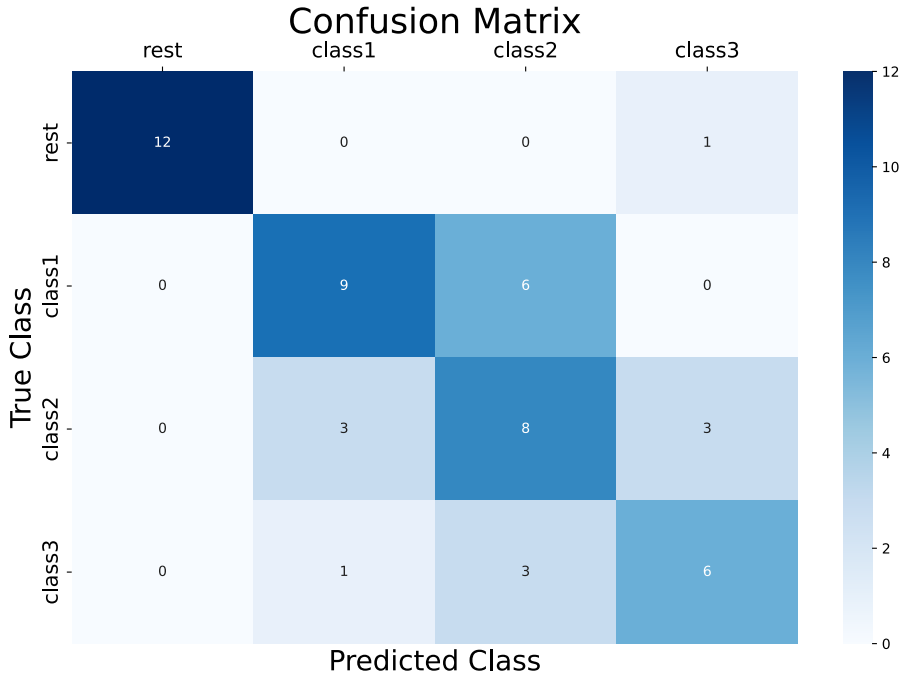

Figure 4.14: Confusion matrix for k-PCA + SVM with Kruskal-Wallis as feature selection on Audio N-Back dataset

Dual N-Back dataset

The results shown in Tables 4.36 and 4.37 for the Dual N-Back dataset are quite disappointing, not even remotely reaching the performance previously achieved in accuracy and F1 score.

Table 4.36: Dual N-Back - One vs All - Min-Max Normalization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + SVM	63.46	56.88	69.23	61.14	71.15	63.38
k-PCA + SVM	63.46	49.25	69.23	60.45	65.38	55.94
LDA + SVM	71.15	56.57	**78.85**	61.28	71.15	55.87

Table 4.37: Dual N-Back - One vs All - Standardization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + SVM	73.08	56.71	69.23	55.38	75.0	66.42
k-PCA + SVM	63.46	57.5	67.31	60.21	65.38	57.77
LDA + SVM	67.31	51.4	67.31	60.76	71.15	65.67

4.4.4 Ensemble learning introduction

Ensemble learning is a popular approach in machine learning that involves combining multiple base learners to improve prediction accuracy and robustness as already explained in Section 2.9.
These techniques, such as bagging, boosting, and stacking, harness the diversity of individual models to collectively enhance overall performance.
In this study, for brevity and clarity of presentation, only the Tables corresponding to the normalization method yielding the best results are showcased.
However, it's important to note that all experimental outcomes, including those with alternative normalization techniques, are fully documented in the Appendix C.2.2.
This ensures a concise yet thorough exploration of the ensemble learning strategies employed and their comparative effectiveness.

Random Forest

In Subsection 2.9.1, we discussed random forest, which is a popular machine learning technique used for both classification and regression tasks. A random forest is an ensemble learning method that operates by constructing a multitude of decision trees during training.
Each tree in the forest is built using a random subset of the training data and a

random subset of features, which helps to reduce overfitting and improve generalization performance.

In this algorithm, there isn't a dominant normalization method over the others. As shown in Table 4.38, the Stroop dataset achieves a 73% accuracy, which is a commendable result but loses significance when combined with the 59% of the F1 score.
Similarly, the other datasets presented in Tables 4.39, 4.40 and 4.41 yield results comparable to those of the best-performing models but show no potential for improvement in either accuracy or F1 score.
It's intriguing to note that for datasets utilizing standardization (Stroop and Audio N-Back), PCA performs better as a dimensionality reduction technique, whereas for datasets employing Min-Max normalization (Visual and Dual N-Back), k-PCA achieves superior results.

Table 4.38: Stroop - Random forest - Standardization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + RF	67.31	59.05	**73.08**	58.99	67.31	60.56
k-PCA + RF	69.23	54.98	69.23	55.43	65.38	58.45

Table 4.39: Visual N-Back - Random forest - Min-Max Normalization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + RF	69.23	61.35	67.31	59.99	59.62	55.69
k-PCA + RF	69.23	65.73	**75.0**	69.14	61.54	58.54

Table 4.40: Audio N-Back - Random forest - Standardization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + RF	53.85	52.47	**65.38**	64.1	61.54	61.14
k-PCA + RF	51.92	49.3	57.69	58.26	50.0	45.44

Table 4.41: Dual N-Back - Random forest - Min-Max Normalization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + RF	75.0	51.25	65.38	48.5	76.92	57.14
k-PCA + RF	75.0	55.81	71.15	67.33	**78.85**	75.11

Adaptive Boosting

Adaptive boosting, also known as AdaBoost, is a machine learning algorithm primarily used for classification tasks.
It works by iteratively training weak classifiers on subsets of the data and adjusting the weights of misclassified samples to prioritize them in subsequent iterations.
For further details on AdaBoost, please refer to Subsection 2.9.2 for a more in-depth exploration.

Before analyzing these results, it is worth noting that the best results were consistently achieved using Min-Max scaling for all datasets using AdaBoost.
In the context of Table 4.42, we see that, much like Random Forest, the Stroop test delivers positive accuracy results without commensurate F1 score performance.
A similar observation can be made for the results obtained on the Visual N-Back dataset in Table 4.45.
Regarding the Visual and Dual N-Back datasets, represented in Tables 4.43 and 4.44, both achieve matched values of accuracy and F1 score. However, these do not improve upon the performance previously achieved on these datasets.

Table 4.42: Stroop - Adaptive Boosting - Min-Max Normalization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + AB	**73.08**	58.9	63.46	58.0	65.38	57.95
k-PCA + AB	67.31	54.32	65.38	60.52	67.31	53.9

Table 4.43: Visual N-Back - Adaptive Boosting - Min-Max Normalization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + AB	59.62	54.45	71.15	68.41	59.62	50.43
k-PCA + AB	61.54	54.16	63.46	55.09	65.38	59.35

Table 4.44: Audio N-Back - Adaptive Boosting - Min-Max Normalization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + AB	50.0	44.33	59.62	61.75	53.85	54.51
k-PCA + AB	57.69	53.58	53.85	54.14	59.62	57.24

Table 4.45: Dual N-Back - Adaptive Boosting - Min-Max Normalization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + AB	75.0	51.25	73.08	64.58	65.38	51.31
k-PCA + AB	57.69	47.35	71.15	49.53	**78.85**	59.66

eXtreme Gradient Boosting

eXtreme Gradient Boosting (XGB) is a powerful machine learning algorithm known for its efficiency and effectiveness in solving regression, classification, and ranking problems, to know more details about it, please refer to Subsection 2.9.3.

Despite the algorithm's touted advantages such as enhanced performance and regularization, our empirical analysis across multiple datasets failed to demonstrate significant improvements over alternative ensemble learning methods.
In fact, the algorithm's performance often proved comparable or even inferior to these alternatives, as evidenced by the results presented in Tables 4.46, 4.47, 4.48, and 4.49.

Table 4.46: Stroop - XG Boosting - Standardization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + XGB	69.23	65.25	65.38	59.54	65.38	58.64
k-PCA + XGB	65.38	56.44	63.46	52.42	57.69	51.69

Table 4.47: Visual N-Back - XG Boosting - Min-Max Normalization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + XGB	59.62	54.95	69.23	61.86	69.23	67.0
k-PCA + XGB	65.38	61.43	71.15	66.71	61.54	58.05

Table 4.48: Audio N-Back - XG Boosting - Standardization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + XGB	51.92	46.78	63.46	63.58	53.85	53.98
k-PCA + XGB	57.69	53.23	61.54	62.06	46.15	46.06

Table 4.49: Dual N-Back - XG Boosting - Standardization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + XGB	**78.85**	60.41	65.38	50.0	73.08	55.54
k-PCA + XGB	67.31	49.56	75.0	56.73	76.92	59.08

4.4.5 Multi-Layer Perceptron

The Multilayer Perceptron (MLP) is a type of artificial neural network characterized by multiple layers of neurons.
Each neuron in a layer is connected to every neuron in the subsequent layer, forming a densely connected network. These connections are associated with weights that determine the strength of the connections and influence the signal transmission.
An in-depth analysis of this algorithm could be found in the background subsection 2.10.1.

As evident from Tables 4.50 and 4.53, this type of algorithm did not perform well on the Stroop dataset and the Dual N-Back dataset.
Conversely, in the Visual dataset (Table 4.51), it achieved almost 77% accuracy with a 75% F1 score, while in the Audio N-Back dataset (Table 4.52), it achieved 67.3% accuracy and a 67.7% F1 score, both of which can be considered among the best performances obtained in the project.
It is also worth considering that training deep learning algorithms requires a large volume of data for proper performance, thus these results are by no means guaranteed.

Table 4.50: Stroop - Multi-layer perceptron - Min-Max Normalization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + MLP	67.31	59.03	65.38	65.61	55.77	54.05
k-PCA + MLP	71.15	62.62	61.54	59.11	55.77	45.47

Table 4.51: Visual N-Back - Multi-layer perceptron - Standardization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + MLP	69.23	68.02	65.38	63.67	71.15	68.83
k-PCA + MLP	**76.92**	74.98	69.23	69.3	71.15	70.22

Table 4.52: Audio N-Back - Multi-layer perceptron - Standardization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + MLP	65.38	64.31	**67.31**	67.69	65.38	64.16
k-PCA + MLP	46.15	44.45	57.69	57.49	57.69	57.57

Table 4.53: Dual N-Back - Multi-layer perceptron - Standardization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + MLP	71.15	54.86	69.23	59.84	73.08	67.32
k-PCA + MLP	75.0	57.11	71.15	53.69	73.08	56.33

4.4.6 Top performing pipelines

This section provides a concise overview of the top three performing machine learning pipelines for each dataset considered in our study.
By analyzing the results obtained from various algorithms applied to different datasets, we identify the most effective combinations of preprocessing steps and model architectures.

Table 4.54: Stroop dataset - Top 3 pipelines

Scaling	FS	DR	Algorithm	Metrics	
				ACC (%)	F1 (%)
Min-Max	Anova	LDA	KNN	75.00	70.23
Standardization	-	k-PCA	LDA	73.08	58.94
Standardization	-	LDA	SVM (OvA)	71.15	66.93

Table 4.55: Visual N-Back dataset - Top 3 pipelines

Scaling	FS	DR	Algorithm	Metrics	
				ACC (%)	F1 (%)
Standardization	Anova	k-PCA	SVM (OvA)	78.85	76.19
Standardization	-	k-PCA	MLP	76.92	74.98
Standardization	Anova	k-PCA	LDA	76.92	73.98

Table 4.56: Audio N-Back dataset - Top 3 pipelines

Scaling	FS	DR	Algorithm	Metrics	
				ACC (%)	F1 (%)
Standardization	K-W	k-PCA	SVM (OvA)	67.31	67.97
Standardization	Anova	PCA	MLP	67.31	67.69
Standardization	K-W	PCA	LDA	65.38	65.65

Table 4.57: Dual N-Back dataset - Top 3 pipelines

Scaling	FS	DR	Algorithm	Metrics	
				ACC (%)	F1 (%)
Standardization	Anova	PCA	LDA	80.77	77.46
Min-Max	Anova	PCA	KNN	80.77	75.43
Min-Max	K-W	k-PCA	RF	78.85	75.11

Hyperparameters In Tables 4.58, 4.59, 4.60 and 4.61, the hyperparameters related to the best models for each dataset are reported. For brevity, the remaining hyperparameters will be omitted.

Table 4.58: Hyperparameters for Stroop's top model

Hyperparameter	Value
LDA_n_components	3
LDA_solver	"svd"
KNN_n_neighbors	6
KNN_weights	"uniform"

Table 4.59: Hyperparameters for Visual N-Back's top model

Hyperparameter	Value
k-PCA_kernel	"rbf"
k-PCA_n_components	9
SVM_df_shape	"ovr"
SVM_kernel	"linear"
SVM_C	10
SVM_class_weight	"balanced"
SVM_gamma	0.001

Table 4.60: Hyperparameters for Audio N-Back's top model

Hyperparameter	Value
k-PCA_kernel	"poly"
k-PCA_n_components	6
SVM_df_shape	"ova"
SVM_kernel	"linear"
SVM_C	5
SVM_class_weight	"balanced"
SVM_gamma	0.001

Table 4.61: Hyperparameters for Dual N-Back's top model

Hyperparameter	Value
PCA_n_components	25
LDA_n_components	25
LDA_solver	"svd"

4.5 Reduced datasets

As previously introduced in Section 3.3, this chapter will present the results obtained for the reduced datasets, with the transition from 4 to 3 classes, for both Stroop and Dual N-Back tasks.

4.5.1 Stroop

Here are the results obtained for the multiclass classification after removing the class related to high-stress load.
For the sake of conciseness, only the best pipelines for each model used are reported in Tables 4.62 and 4.63, sorted by accuracy, with one Table dedicated to each normalization technique.
Analyzing the two Tables, it can be observed that, on average, better results are achieved using standardization. However, the best performance was obtained by a pipeline composed of feature selection using Kruskal, PCA for dimensionality reduction, and KNN as the classification algorithm. This pipeline achieved an accuracy of 80% and an F1 score of 80.31%, representing a 5% and 10% improvement in performance compared to the baseline.
As can be observed in the confusion matrix in Figure 4.15, the model successfully classifies all samples related to the resting state while encountering more difficulty in distinguishing between the other two classes. Specifically, it misclassifies some samples from the medium class as belonging to the low class.

Table 4.62: Stroop dataset - Three classes - Min-Max Normalization - Best results

Method	Metrics	
	ACC (%)	F1 (%)
NoFS + LDA + KNN	64.44	65.45
NoFS + LDA + SVM (OvO)	64.44	65.45
Anova + LDA + SVM (OvR)	64.44	65.45
Anova + k-PCA + LDA	75.56	75.93
NoFS + PCA + SVM (OvR)	75.56	76.51
KRUSKAL + PCA + SVM (OvO)	75.56	76.88
Anova + PCA + RF	77.78	78.41
Anova + k-PCA + KNN	**80.0**	**80.31**

Table 4.63: Stroop dataset - Three classes - Standardization - Best results

Method	Metrics	
	ACC (%)	F1 (%)
KRUSKAL + LDA + KNN	68.89	69.77
NoFS + LDA + SVM (OvO)	71.11	71.67
NoFS + LDA + SVM (OvR)	71.11	72.61
Anova + PCA + SVM (OvR)	75.56	76.88
NoFS + PCA + RF	77.78	76.91
KRUSKAL + PCA + LDA	77.78	77.78
Anova + k-PCA + SVM (OvO)	77.78	78.84
KRUSKAL + PCA + KNN	**77.78**	**79.08**

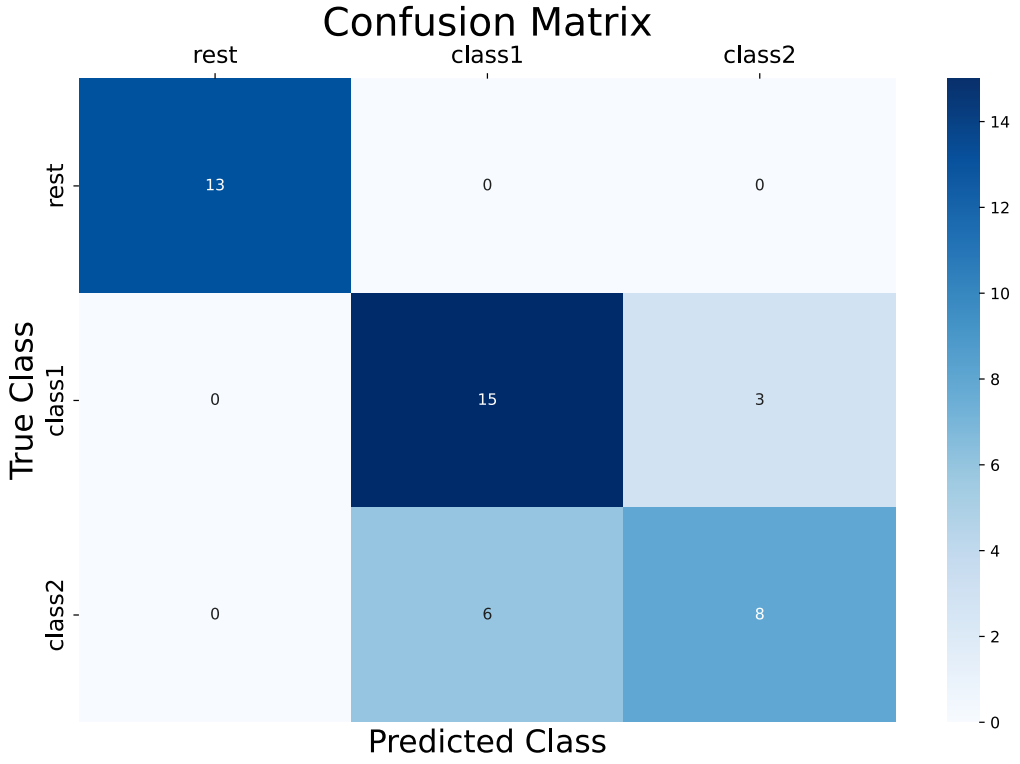

Figure 4.15: Confusion matrix - Stroop three class - k-PCA + KNN

4.5.2 Dual N-Back

In this section, the results obtained by removing the class related to low cognitive load are presented, thus transforming the multiclass task into a 3-label classification.

Contrary to the Stroop task, the best results are achieved with Min-Max normalization.
Through the model composed of PCA and LDA, it achieves 85% accuracy and 81.7% F1 score, improving both performance metrics by 4% compared to the baseline.
As can be observed in Figure 4.16, the issue with this algorithm lies in the massive classification of samples into the high cognitive load class when they belong to the medium cognitive load class. Once again, the samples belonging to the rest states are all classified correctly.

Table 4.64: Dual N-Back dataset - Three classes - Min-Max Normalization -

Method	Metrics	
	ACC (%)	F1 (%)
NoFs + LDA + KNN	70.21	69.95
Anova + LDA + SVM (OvO)	70.21	69.95
NoFS + LDA + SVM (OvR)	74.47	59.52
KRUSKAL + PCA + SVM (OvR)	74.47	75.09
KRUSKAL + PCA + RF	75.56	75.93
KRUSKAL + k-PCA + KNN	80.85	71.47
Anova + k-PCA + SVM (OvO)	80.85	80.29
NoFS + PCA + LDA	**85.11**	**81.7**

Table 4.65: Dual N-Back dataset - Three classes - Standardization - Best results

Method	Metrics	
	ACC (%)	F1 (%)
Anova + LDA + KNN	65.96	65.66
Anova + LDA + SVM (OvR)	68.09	66.78
Anova + LDA + SVM (OvO)	68.09	67.15
Anova + PCA + RF	72.34	68.35
NoFS + PCA + LDA	76.6	71.24
Anova + k-PCA + KNN	78.72	77.6
NoFS + PCA + SVM (OvO)	80.85	78.09
NoFS + PCA + SVM (OvR)	80.85	79.33

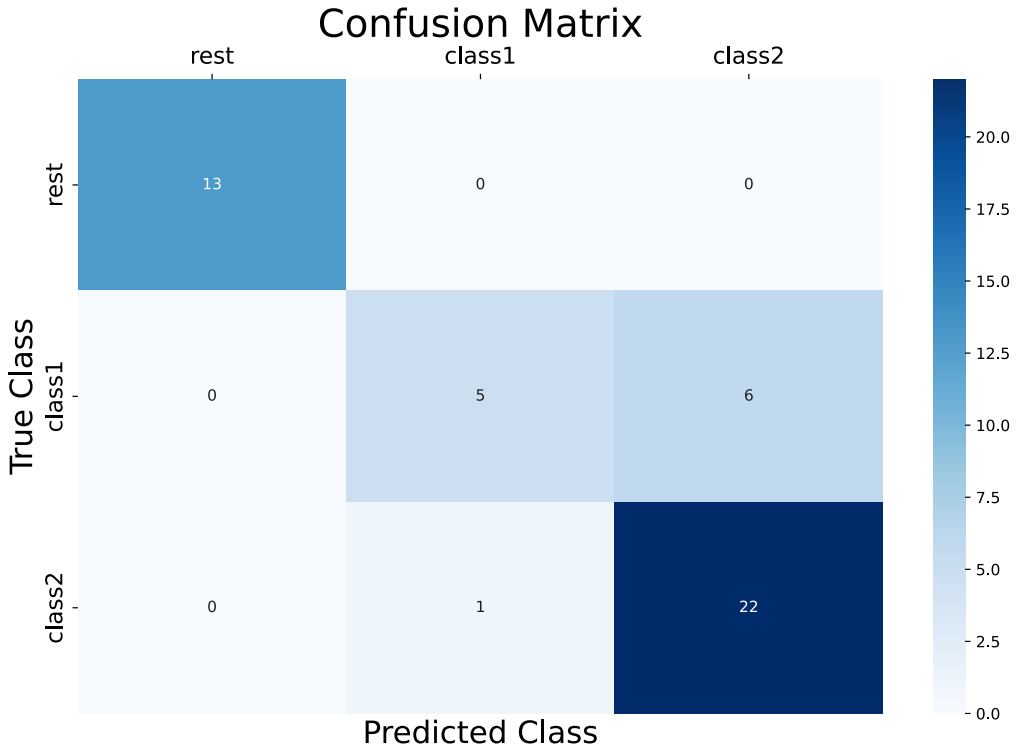

Figure 4.16: Confusion matrix - Dual N-Back three class - PCA + LDA

4.6 Subsets

In this section, the results of experiments related to subsets will be listed and analyzed.
For each dataset, models were trained using only the features related to one or more sensors and all their possible combinations.
The combinations of all possible signals using from 1 to 5 signals amount to 62. For each of these combinations, 33 different pipelines were applied, each trained on multiple hyperparameters.
Training a single dataset required more than ten days of training, carried out stoically by a computer located at the Politecnico di Torino: PC- LSE-1856.

4.6.1 Stroop

As can be observed in Table 4.66, the best sensor subsets consist in a maximum of three signals and they all include ocular sensor.
Despite the first two configurations having higher accuracy values, the most desirable model remains the third, which achieves an F1-score of 78.89%.

Table 4.66: Top performing subsets - Stroop

Sensors	Method	Metrics	
		ACC (%)	F1 (%)
T - EYE	Anova + PCA + SVM (OvO)	78.85	75.55
RESP - fNIRS - EYE	Anova + PCA + KNN	78.85	73.25
ECG - EYE	Kruskal + PCA + SVM (OvO)	76.92	78.89

4.6.2 Visual N-Back

Analyzing Table 4.67, it can be deduced that ECG and EDA are the sensors providing the most exploited features by the subsets for the Visual N-Back dataset. Additionally, it is interesting to note that all three pipelines are composed of PCA for dimensionality reduction and LDA as the classifier.

Table 4.67: Top performing subsets - Visual N-Back

Sensors	Method	Metrics	
		ACC (%)	F1 (%)
ECG - EDA - EYE	Kruskal + PCA + LDA	80.77	80.3
ECG - EDA - T - FNIRS	Anova + PCA + LDA	80.77	79.07
ECG - EDA - T - RESP - EYE	Kruskal + PCA + LDA	80.77	79.07

4.6.3 Audio N-Back

In Table 4.68, the results for the Audio N-Back dataset are presented, where, once again, ECG stands out as the most exploited sensor.
All three models utilize PCA for dimensionality reduction and SVM (different ones) as classifiers. It is also noteworthy that these performances exceed those of the best models using all six signals by more than 5%.

Table 4.68: Top performing subsets - Audio N-Back

Sensors	Method	Metrics	
		ACC (%)	F1 (%)
ECG - RESP	Anova + PCA + SVM(OvR)	73.08	73.7
ECG - EDA - T - RESP	Kruskal + PCA + SVM(OvO)	73.08	73.12
ECG - EDA - EYE	Kruskal + PCA + SVM(OvR)	73.08	72.45

4.6.4 Dual N-Back

In the results presented in Table 4.69 for the Dual N-Back, there are no predominant sensors or models. However, once again, the results are better compared to those obtained with all six sensors.

Table 4.69: Top performing subsets - Dual N-Back

Sensors	Method	Metrics	
		ACC (%)	F1 (%)
T - EYE	Anova + LDA + SVM(OvR)	84.62	83.1
RESP - FNIRS	Kruskal + PCA + LDA	82.69	85.81
ECG - RESP - FNIRS - EYE	Kruskal + PCA + LDA	82.69	85.81

4.6.5 Insights and observations

The results obtained and presented in the subsections preceding this one may be surprising.
The question arises: why do we achieve better results using fewer sensors and fewer features compared to using all six signals?
One explanation for this phenomenon is that simplifying models and reducing noise contributes to better generalization and training improvement. This is particularly important in mitigating the curse of dimensionality, where high-dimensional and sparse data make it difficult to identify significant patterns without substantial amounts of data.
Additionally, reducing the number of features increases the relevance of the remaining features, enabling the model to better comprehend and utilize them.
Reducing the number of sensors and features not only simplifies the models but also enhances their ability to generalize and extract meaningful patterns from the

data. This, in turn, leads to better performance on unseen data.
This experiment had significant practical and applicative implications because allowed BiLoad to simplify the acquisition process removing sensors while maintaining a high level of reliability in stress and mental workload prediction.

Chapter 5

Conclusions

This research conducted with the eLions group at Politecnico di Torino has made significant progress in understanding the relationship between physiological signals and cognitive states such as stress and mental workload.
The study's innovative approach, which involved the use of machine learning algorithms to classify these states based on multimodal physiological signals, yielded promising results.

The initial use of statistical tests such as ANOVA and Kruskal-Wallis has enabled identifying the most significant features closely correlated with states of stress and cognitive load.
The development of unsupervised algorithms has facilitated a better understanding of the intrinsic structure of various datasets and their visualization through projection into lower-dimensional spaces.

Models developed using supervised algorithms demonstrated high accuracy in classifying samples into two states: "rest" and "altered state," achieving nearly 100% accuracy across all four datasets.
Additionally, novel models were successfully constructed to classify samples into four states: "rest" and three levels of cognitive load or stress. These models utilized various algorithms and methodologies, achieving encouraging results.
Specifically, they surpassed 80% accuracy for the Dual N-Back dataset, exceeded 75% for the Visual N-Back and Stroop tests, and achieved slightly below 70% for the Audio N-Back. Notably, these achievements were attained despite the challenge of limited objective labeled data available for model training.

The study also addressed the challenge of heavily imbalanced classes in the Stroop test and Dual N-Back dataset by identifying and removing outliers, leading to a transition to a 3-class classification. This adjustment resulted in improved

accuracy by 5% and 4% for the Stroop test and Dual N-Back tasks, respectively.

Furthermore, all possibilities for utilizing various subsets of signals have been investigated so that future applications can be developed in an application-driven manner based on the needs of different application domains. This includes addressing issues such as the absence of certain inconvenient sensors or making the system more cost-effective by eliminating some.

The project's outcome is a comprehensive and versatile framework capable of accurately classifying the differentiation between a state of rest and an altered state based on an individual's physiological parameters.
It can discern various degrees of cognitive workload and stress severity, even when certain biological signals are absent.
This framework lays the groundwork for creating a safety device that analyses physiological signals to assess the operator's condition, especially during critical moments.
It could enhance safety by providing real-time evaluations of the operator's state, reducing potential risks and ensuring safer operations.

In conclusion, this research has made significant contributions to the field of human-machine interaction by developing a robust, multimodal, and flexible machine learning framework that can accurately assess an operator's cognitive state based on physiological signals.
This work not only advances our understanding of the relationship between physiological signals and cognitive states but also has practical implications for enhancing safety in various industries where human-machine interaction is prevalent.

5.1 Future work

Future research can leverage the findings from this study to further advance the application of the developed machine learning framework in real-world settings.
One avenue for future exploration involves deploying the models in practical scenarios to evaluate their effectiveness and reliability in assessing an operator's cognitive state in diverse contexts.
Furthermore, the adaptability of the framework opens up opportunities for customization and refinement to suit specific industry requirements and operational environments.
Upcoming investigations may emphasize tailoring machine learning algorithms to address the distinct challenges and complexities encountered within different

industries, including aviation, healthcare, manufacturing, and transportation.
Another fundamental task will be implementing these models in real-time, addressing the issues concerning acquisition, processing, and classification, potentially leveraging microchips or alternative hardware solutions.
In addition to refining the existing models, future research could also explore the integration of additional modalities or features to enhance the robustness and accuracy of cognitive state assessment. For example, incorporating contextual information, environmental factors, or behavioural cues alongside physiological signals could provide a more comprehensive understanding of human-machine interaction dynamics.

Future research efforts should focus on bridging the gap between theory and practice by translating the insights gained from this study into tangible applications that enhance safety, efficiency, and overall human-machine interaction across diverse industries.
By building upon these foundations, researchers can continue to push the boundaries of human-centred technology and contribute to the advancement of human-machine collaboration in the digital age.

Appendix A

Dataset analysis

A.1 Features by signal

Category	Feature
ECG	mean_bpm
ECG	std_bpm
ECG	median_bpm
ECG	pnn50
ECG	mean_ecg_plf
ECG	std_ecg_plf
ECG	mean_ecg_phf
ECG	std_ecg_phf
ECG	mean_ecg_plf_phf
ECG	std_ecg_plf_phf
EDA	mean_eda_scl
EDA	std_eda_scl
EDA	slope_eda_scl
EDA	mean_amplitude_eda_scr
EDA	std_amplitude_eda_scr
EDA	mean_rise_time_eda_scr
EDA	std_rise_time_eda_scr
EDA	average_peaks_number_eda_scr
T	initial_temperature
T	final_temperature
T	delta_temperature
T	mean_temperature
T	std_temperature
T	delta_over_time_temperature

Category	**Feature**
T	interpolated_slope_temperature
T	initial_first_derivative_temperature
T	final_first_derivative_temperature
T	delta_derivative_temperature
T	mean_first_derivative_temperature
T	std_first_derivative_temperature
T	delta_first_derivative_over_time_temperature
T	interpolated_slope_first_derivative_temperature
RESP	mean_breath_rate_respiration
RESP	std_breath_rate_respiration
RESP	mean_inspiratory_time_respiration
RESP	std_inspiratory_time_respiration
RESP	mean_expiratory_time_respiration
RESP	std_expiratory_time_respiration
RESP	mean_timing_ratio_respiration
RESP	std_timing_ratio_respiration
RESP	mean_amplitude_respiration
RESP	std_amplitude_respiration
RESP	mean_minute_ventilation
RESP	std_minute_ventilation
RESP	mean_plf_respiration
RESP	std_plf_respiration
RESP	mean_phf_respiration
RESP	std_phf_respiration
RESP	mean_plf_phf_ratio_respiration
RESP	std_plf_phf_ratio_respiration
FNIRS	min_fnirs_oxy
FNIRS	max_fnirs_oxy
FNIRS	mean_fnirs_oxy
FNIRS	variance_fnirs_oxy
FNIRS	std_fnirs_oxy
FNIRS	skewness_fnirs_oxy
FNIRS	kurtosis_fnirs_oxy
FNIRS	power_band_fnirs_oxy
FNIRS	max_frequency_fnirs_oxy
FNIRS	median_frequency_fnirs_oxy
FNIRS	spectral_entropy_fnirs_oxy
FNIRS	mean_difference_fnirs_oxy
FNIRS	peak_to_peak_fnirs_oxy
FNIRS	slope_fnirs_oxy

Category	**Feature**
FNIRS	zero_crossing_fnirs_oxy
FNIRS	polarity_fnirs_oxy
FNIRS	entropy_fnirs_oxy
FNIRS	auc_fnirs_oxy
FNIRS	te_fnirs_oxy
FNIRS	min_fnirs_deoxy
FNIRS	max_fnirs_deoxy
FNIRS	mean_fnirs_deoxy
FNIRS	variance_fnirs_deoxy
FNIRS	std_fnirs_deoxy
FNIRS	skewness_fnirs_deoxy
FNIRS	kurtosis_fnirs_deoxy
FNIRS	power_band_fnirs_deoxy
FNIRS	max_frequency_fnirs_deoxy
FNIRS	median_frequency_fnirs_deoxy
FNIRS	spectral_entropy_fnirs_deoxy
FNIRS	mean_difference_fnirs_deoxy
FNIRS	peak_to_peak_fnirs_deoxy
FNIRS	slope_fnirs_deoxy
FNIRS	zero_crossing_fnirs_deoxy
FNIRS	polarity_fnirs_deoxy
FNIRS	entropy_fnirs_deoxy
FNIRS	auc_fnirs_deoxy
FNIRS	te_fnirs_deoxy
EYE	eye_duration_blinking
EYE	eye_frequency_blinking
EYE	eye_interval_blinking
EYE	eye_duration_saccade
EYE	eye_frequency_saccade
EYE	eye_velocity_x_saccade
EYE	eye_velocity_y_saccade
EYE	eye_duration_fixation
EYE	eye_frequency_fixation
EYE	eye_value_si
EYE	eye_velocity_si
EYE	eye_duration_si
EYE	eye_frequency_si
EYE	eye_velocity_single_si
EYE	eye_velocity_over_duration_phase_si
EYE	eye_relative_diameter_left

Category	**Feature**
EYE	eye_relative_diameter_right
EYE	eye_phf_x
EYE	eye_plf_x
EYE	eye_plf_phf_ratio_x
EYE	eye_phf_y
EYE	eye_plf_y
EYE	eye_plf_phf_ratio_y
RT	reaction_time

Table A.1: Features and categories

Appendix B

Statistical tests

B.1 Stroop dataset

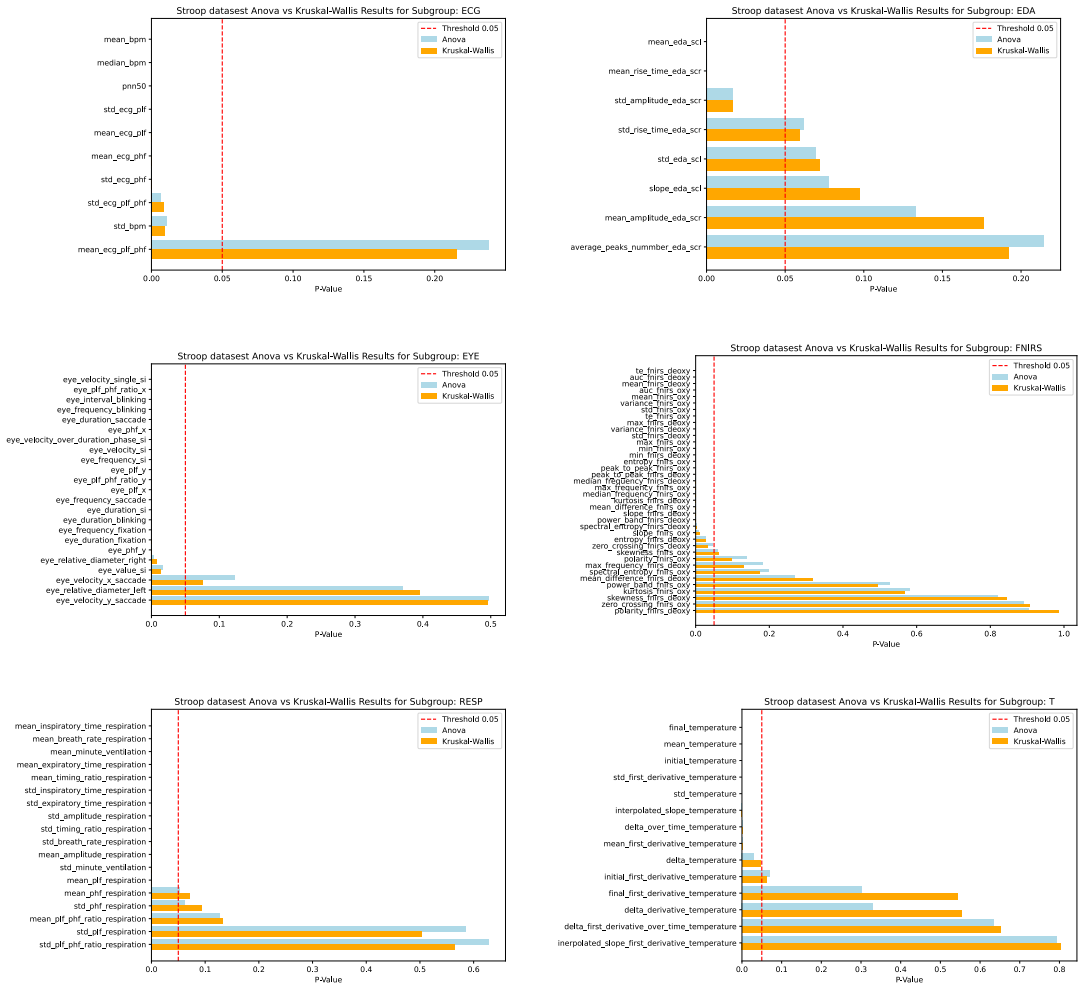

Figure B.1: Anova vs Kruskal-Wallis comparison for different signals on Stroop Dataset

B.2 Visual n-back dataset

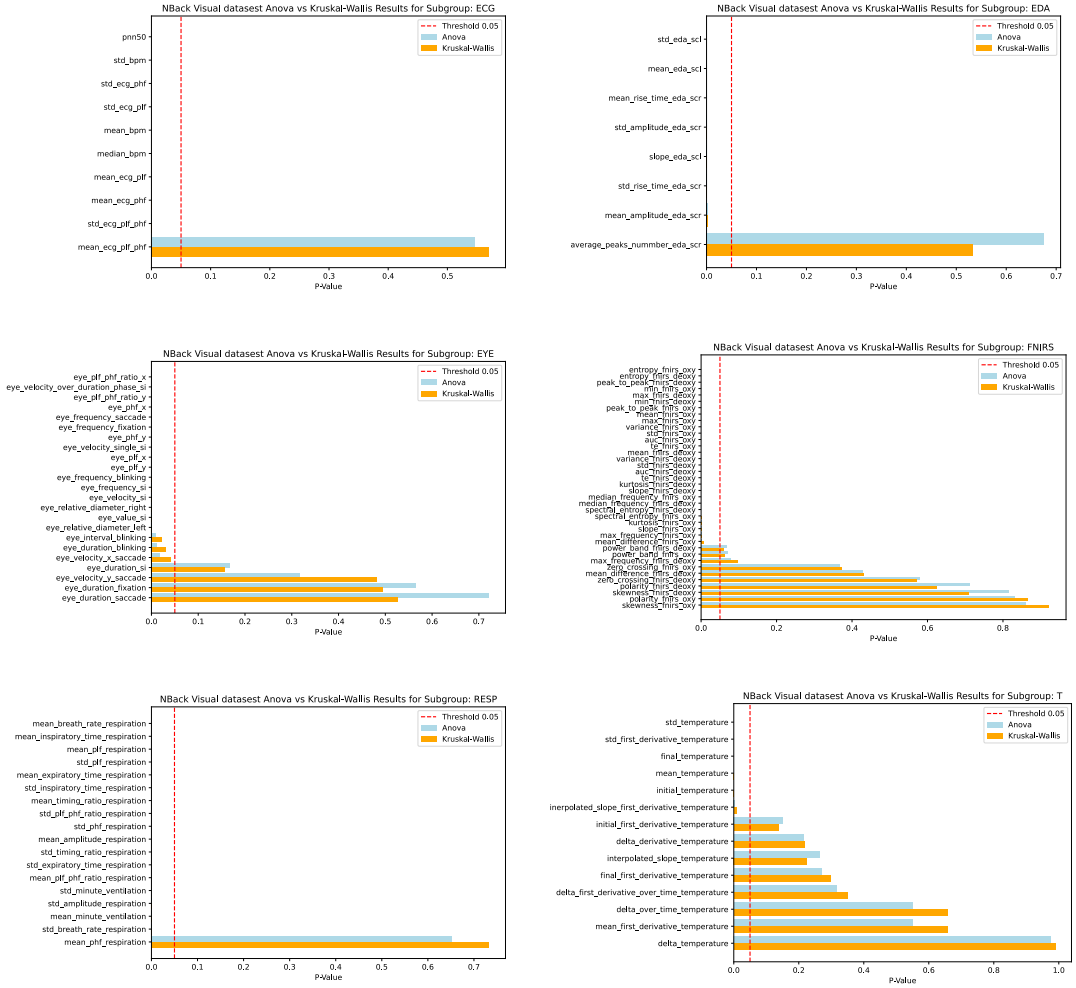

Figure B.2: Anova vs Kruskal-Wallis comparison for different signals on visual n-back dataset

B.3 Audio n-back dataset

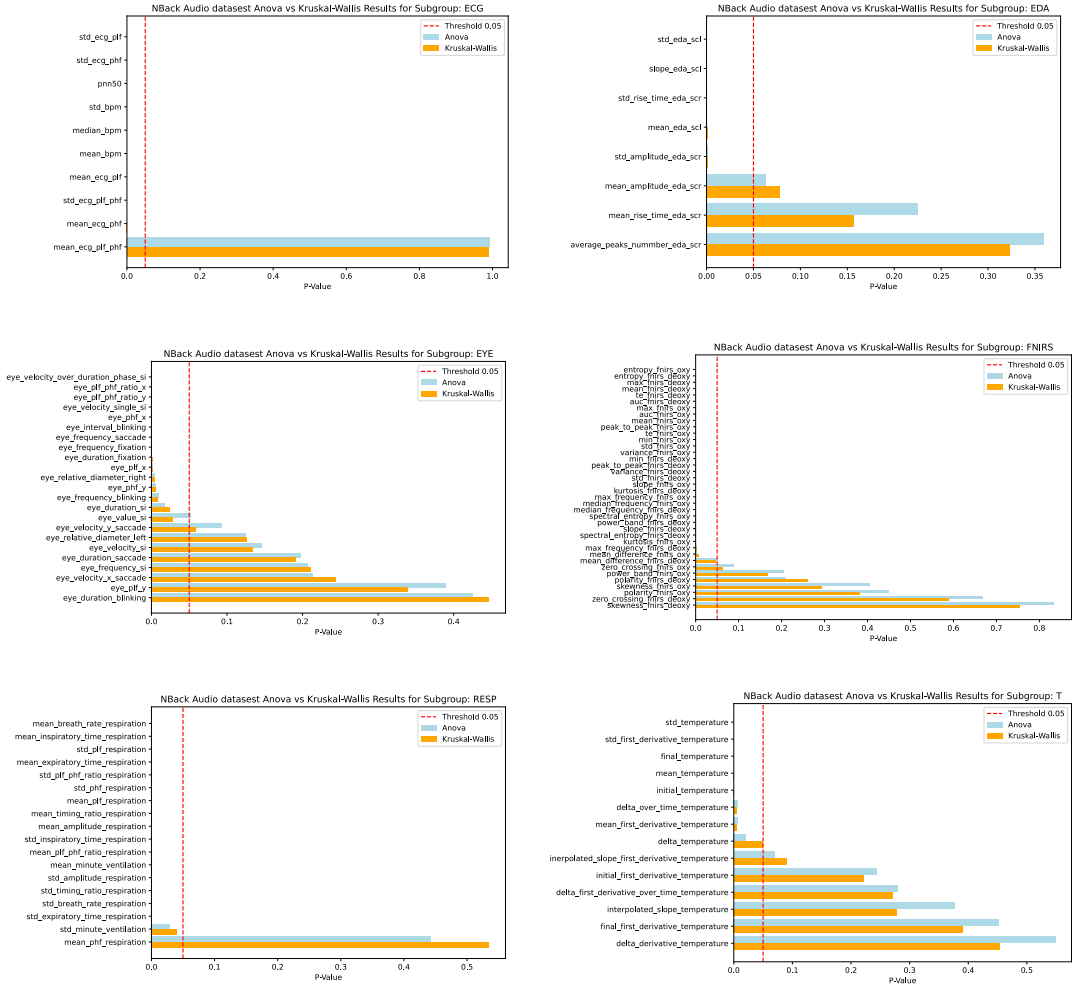

Figure B.3: Anova vs Kruskal-Wallis comparison for different signals on audio n-back dataset

B.4 Dual n-back dataset

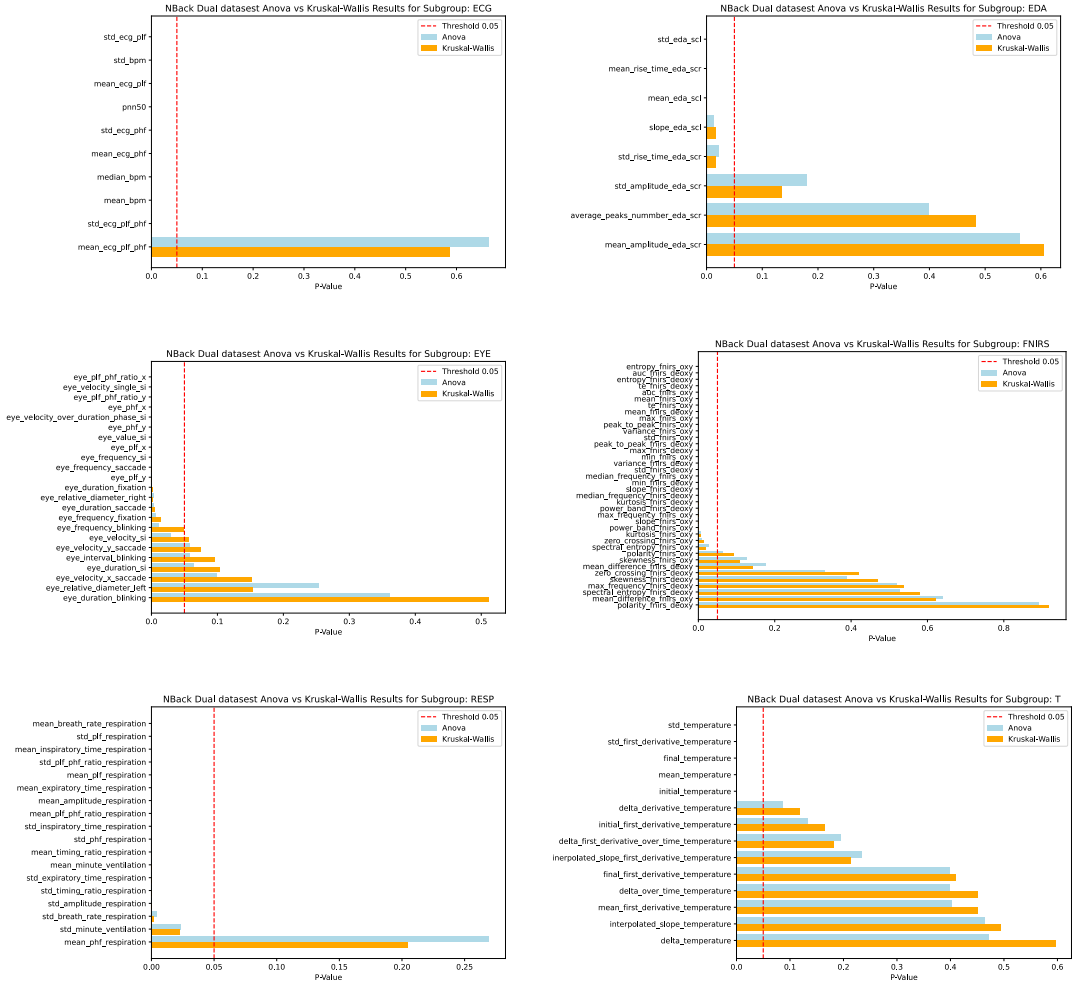

Figure B.4: Anova vs Kruskal-Wallis comparison for different signals on dual n-back Dataset

Appendix C

Unsupervised algorithms

C.1 3D t-sne graphs

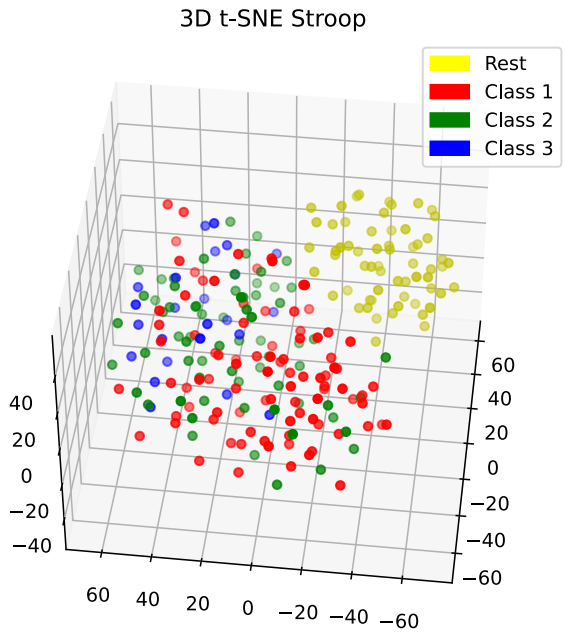

Figure C.1: 3D t-sne for Stroop dataset

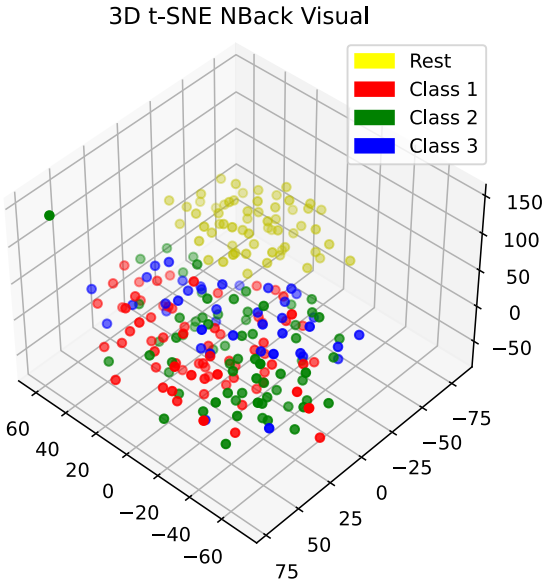

Figure C.2: 3D t-sne for visual n-back dataset

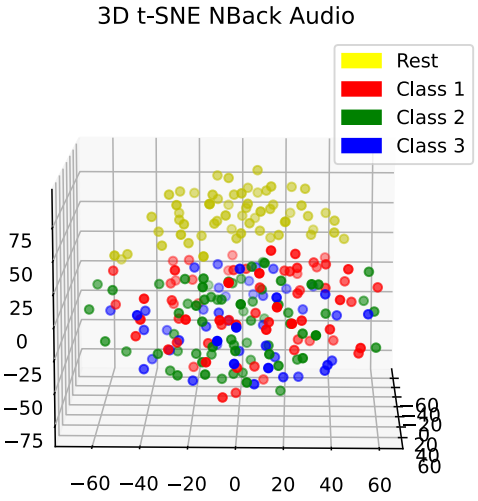

Figure C.3: 3D t-sne for audio n-back dataset

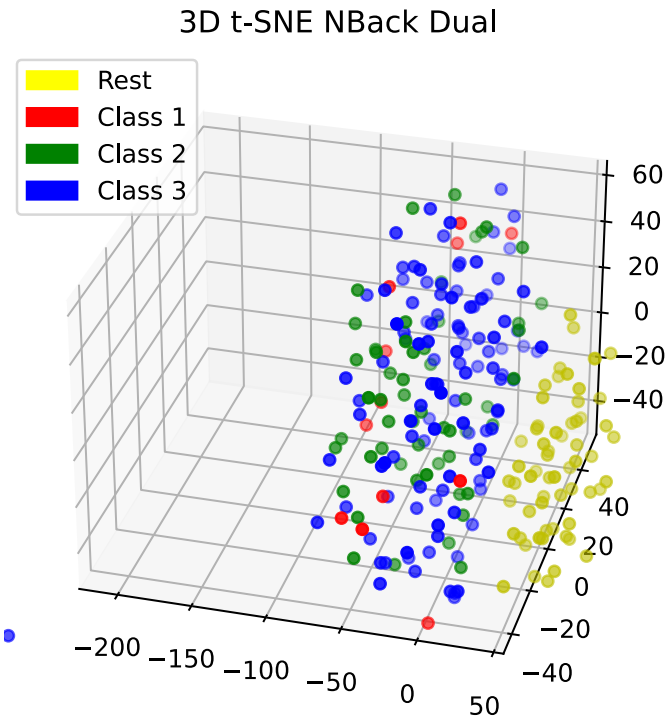

Figure C.4: 3D t-sne for dual n-back dataset

C.2 Multiclass classification

C.2.1 Support Vector Machine - One vs One

Table C.1: Stroop - One vs One - Min-Max Normalization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + SVM	59.62	53.65	63.46	58.09	65.38	59.62
k-PCA + SVM	57.69	51.1	65.38	60.1	61.54	54.76
LDA + SVM	67.31	65.44	69.23	62.04	69.23	62.78

Table C.2: Stroop - One vs One - Standardization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + SVM	63.46	61.07	59.62	54.04	65.38	61.55
k-PCA + SVM	61.54	58.54	63.46	60.58	67.31	64.02
LDA + SVM	53.85	54.01	65.38	59.76	63.46	58.13

Table C.3: Visual N-Back - One vs One - Min-Max Normalization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + SVM	59.62	58.77	73.08	67.78	75.0	68.72
k-PCA + SVM	67.31	65.52	75.0	68.59	76.92	72.88
LDA + SVM	67.31	65.15	59.62	58.07	61.54	60.12

Table C.4: Visual N-Back - One vs One - Standardization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + SVM	71.15	63.74	73.08	69.44	73.08	71.19
k-PCA + SVM	63.46	59.62	75.0	69.12	75.0	69.17
LDA + SVM	51.92	51.54	71.15	70.66	65.38	64.5

Table C.5: Audio N-Back - One vs One - Min-Max Normalization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + SVM	63.46	64.24	57.69	58.14	57.69	58.28
k-PCA + SVM	61.54	61.38	59.62	59.52	65.38	65.78
LDA + SVM	59.62	58.88	63.46	62.37	63.46	62.07

Table C.6: Audio N-Back - One vs One - Standardization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + SVM	55.77	55.29	61.54	61.57	61.54	62.17
k-PCA + SVM	57.69	57.04	65.38	66.17	59.62	58.56
LDA + SVM	51.92	51.66	59.62	59.66	59.62	59.88

Table C.7: Dual N-Back - One vs One - Min-Max Normalization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + SVM	67.31	51.7	65.38	50.94	67.31	62.98
k-PCA + SVM	55.77	48.33	67.31	53.9	69.23	60.97
LDA + SVM	63.46	47.12	76.92	70.93	75.0	58.43

Table C.8: Dual N-Back - One vs One - Standardization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + SVM	69.23	65.61	63.46	55.29	65.38	57.44
k-PCA + SVM	63.46	59.54	65.38	58.16	65.38	57.48
LDA + SVM	65.38	52.22	67.31	60.66	73.08	66.61

C.2.2 Ensemble learning - alternatives normalization

Table C.9: Stroop - Random forest - Min-Max Normalization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + RF	67.31	59.05	61.54	55.46	67.31	54.52
k-PCA + RF	57.69	53.96	61.54	55.56	65.38	57.53

Table C.10: Visual N-Back - Random forest - Standardization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + RF	67.31	60.44	59.62	57.36	69.23	64.69
k-PCA + RF	67.31	60.56	65.38	60.97	69.23	64.25

Table C.11: Audio N-Back - Random forest - Min-Max Normalization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + RF	50.00	45.24	59.62	53.13	59.62	58.22
k-PCA + RF	59.62	57.45	55.77	51.84	50.00	48.67

Table C.12: Dual N-Back - Random forest - Standardization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + RF	75.0	58.04	73.08	53.61	73.08	53.61
k-PCA + RF	69.23	52.65	73.08	54.74	76.92	58.46

Table C.13: Stroop - Adaptive boost - Standardization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + AB	73.08	64.41	57.69	48.12	65.38	52.53
k-PCA + AB	65.38	53.79	61.54	57.54	65.38	52.86

Table C.14: Visual N-Back - Adaptive boost - Standardization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + AB	59.62	57.68	65.38	62.36	71.15	65.32
k-PCA + AB	63.46	60.92	71.15	66.98	69.23	65.36

Table C.15: Audio N-Back - Adaptive boost - Standardization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + AB	55.77	51.36	53.85	52.92	53.85	54.95
k-PCA + AB	50.0	43.04	53.85	52.39	59.62	57.24

Table C.16: Dual N-Back - Adaptive boost - Standardization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + AB	71.15	49.56	65.38	60.66	65.38	42.5
k-PCA + AB	73.08	53.75	73.08	53.9	71.15	53.69

Table C.17: Stroop - XG boost - Min-Max Normalization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + XGB	67.31	60.13	59.62	49.76	67.31	60.88
k-PCA + XGB	53.85	50.27	59.62	54.5	65.38	58.45

Table C.18: Visual N-Back - XG boost - Standardization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + XGB	67.31	62.7	69.23	64.71	63.46	60.34
k-PCA + XGB	71.15	66.46	65.38	64.19	67.31	62.94

Table C.19: Audio N-Back - XG boost - Min-Max Normalization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + XGB	59.62	57.59	55.77	56.77	61.54	63.11
k-PCA + XGB	46.15	45.16	59.62	57.37	61.54	63.11

Table C.20: Dual N-Back - XG boost - Min-Max Normalization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + XGB	71.15	47.29	67.31	49.41	71.15	53.69
k-PCA + XGB	63.46	45.2	69.23	50.68	73.08	52.19

Table C.21: Stroop - Multi-layer perceptron - Standardization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + MLP	67.31	59.6	59.62	49.49	59.62	43.23
k-PCA + MLP	57.69	48.62	69.23	57.15	65.38	58.79

Table C.22: Visual N-Back - Multi-layer perceptron - Min-Max Normalization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + MLP	69.23	67.07	63.46	62.01	69.23	64.48
k-PCA + MLP	63.46	60.14	67.31	61.37	61.54	59.07

Table C.23: Audio N-Back - Multi-layer perceptron - Min-Max Normalization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + MLP	63.46	62.65	61.54	62.19	59.62	58.33
k-PCA + MLP	26.92	10.61	57.69	57.57	57.69	57.57

Table C.24: Dual N-Back - Multi-layer perceptron - Min-Max Normalization

Method	NoFs		Anova		Kruskal-Wallis	
	ACC (%)	F1 (%)	ACC (%)	F1 (%)	ACC (%)	F1 (%)
PCA + MLP	59.62	46.31	71.15	70.63	67.31	64.53
k-PCA + MLP	67.31	52.74	71.15	64.76	69.23	63.0

Bibliography

[1] Nikesh Muthukrishnan, Farhad Maleki, Katie Ovens, Caroline Reinhold, Behzad Forghani, Reza Forghani, et al. «Brief history of artificial intelligence». In: *Neuroimaging Clinics of North America* 30.4 (2020), pp. 393–399 (cit. on p. 1).

[2] Michael Haenlein and Andreas Kaplan. «A brief history of artificial intelligence: On the past, present, and future of artificial intelligence». In: *California management review* 61.4 (2019), pp. 5–14 (cit. on p. 1).

[3] Yanqing Duan, John S Edwards, and Yogesh K Dwivedi. «Artificial intelligence for decision making in the era of Big Data–evolution, challenges and research agenda». In: *International journal of information management* 48 (2019), pp. 63–71 (cit. on p. 1).

[4] Chin-Feng Fan, Ching-Chieh Chan, Hsiang-Yu Yu, and Swu Yih. «A simulation platform for human-machine interaction safety analysis of cyber-physical systems». In: *International journal of industrial ergonomics* 68 (2018), pp. 89–100 (cit. on p. 1).

[5] Changxu Wu and Yili Liu. «Development and evaluation of an ergonomic software package for predicting multiple-task human performance and mental workload in human–machine interface design and evaluation». In: *Computers & Industrial Engineering* 56.1 (2009), pp. 323–333 (cit. on p. 1).

[6] Gabriele Luzzani, Irene Buraioli, Danilo Demarchi, and Giorgio Guglieri. «Preliminary Study of a Pilot Performance Monitoring System Based on Physiological Signals». In: *Aerospace Europe Conference 2023 – 10 EUCASS – 9 CEAS*. 2023, pp. 154–162. DOI: 10.13009/EUCASS2023-349 (cit. on pp. 1, 6).

[7] Dietrich Manzey. «Psychophysiologie mentaler Beanspruchung.» In: *F. Rosler (Hg): Ergebnisse und Anwendungen der Psychophysiologie. Enzyklopadie der Psychologie* 100 (1997), pp. 799–864 (cit. on p. 5).

[8] F Thomas Eggemeier, Glenn F Wilson, Arthur F Kramer, and Diane L Damos. «Workload assessment in multi-task environments». In: *Multiple task performance*. CRC Press, 2020, pp. 207–216 (cit. on p. 6).

[9] Roomana N Siddiqui and Shabana Mazhar. «The multidimensional aspect of stress and its management.» In: *Indian Journal of Health & Wellbeing* 5.8 (2014) (cit. on p. 6).

[10] Hans Selye. «Stress without distress». In: *Psychopathology of human adaptation*. Springer, 1974, pp. 137–146 (cit. on p. 6).

[11] George Fink. *Stress: Concepts, Cognition, Emotion, and Behavior: Handbook of Stress Series, Volume 1*. Vol. 1. Academic Press, 2016 (cit. on p. 6).

[12] Essam Debie, Raul Fernandez Rojas, Justin Fidock, Michael Barlow, Kathryn Kasmarik, Sreenatha Anavatti, Matt Garratt, and Hussein A Abbass. «Multimodal fusion for objective assessment of cognitive workload: a review». In: *IEEE transactions on cybernetics* 51.3 (2019), pp. 1542–1555 (cit. on p. 6).

[13] Rupert G Miller Jr. *Beyond ANOVA: basics of applied statistics*. CRC press, 1997 (cit. on p. 7).

[14] William H Kruskal and W Allen Wallis. «Use of ranks in one-criterion variance analysis». In: *Journal of the American statistical Association* 47.260 (1952), pp. 583–621 (cit. on p. 8).

[15] Henry B Mann and Donald R Whitney. «On a test of whether one of two random variables is stochastically larger than the other». In: *The annals of mathematical statistics* (1947), pp. 50–60 (cit. on p. 8).

[16] Stuart J Russell and Peter Norvig. *Artificial intelligence a modern approach*. London, 2010 (cit. on pp. 9, 11).

[17] Pádraig Cunningham, Matthieu Cord, and Sarah Jane Delany. «Supervised learning». In: *Machine learning techniques for multimedia: case studies on organization and retrieval*. Springer, 2008, pp. 21–49 (cit. on p. 11).

[18] Zoubin Ghahramani. «Unsupervised learning». In: *Summer school on machine learning*. Springer, 2003, pp. 72–112 (cit. on p. 11).

[19] Ankur A Patel. *Hands-on unsupervised learning using Python: how to build applied machine learning solutions from unlabeled data*. O'Reilly Media, 2019 (cit. on p. 11).

[20] Laurens Van der Maaten and Geoffrey Hinton. «Visualizing data using t-SNE.» In: *Journal of machine learning research* 9.11 (2008) (cit. on pp. 11, 12).

[21] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. «A density-based algorithm for discovering clusters in large spatial databases with noise». In: *kdd*. Vol. 96. 34. 1996, pp. 226–231 (cit. on p. 12).

[22] Erich Schubert, Jörg Sander, Martin Ester, Hans Peter Kriegel, and Xiaowei Xu. «DBSCAN revisited, revisited: why and how you should (still) use DBSCAN». In: *ACM Transactions on Database Systems (TODS)* 42.3 (2017), pp. 1–21 (cit. on p. 13).

[23] Karl Pearson. «LIII. On lines and planes of closest fit to systems of points in space». In: *The London, Edinburgh, and Dublin philosophical magazine and journal of science* 2.11 (1901), pp. 559–572 (cit. on p. 13).

[24] Harold Hotelling. «Analysis of a complex of statistical variables into principal components.» In: *Journal of educational psychology* 24.6 (1933), p. 417 (cit. on p. 13).

[25] C Müller Andreas and Sarah Guido. *Introduction to machine learning with python: A guide for data scientists*. O'Reilly Media, Incorporated, 2016 (cit. on p. 13).

[26] Aurélien Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. " O'Reilly Media, Inc.", 2022 (cit. on p. 14).

[27] Ronald A Fisher. «The use of multiple measurements in taxonomic problems». In: *Annals of eugenics* 7.2 (1936), pp. 179–188 (cit. on p. 14).

[28] Evelyn Fix and Joseph Lawson Hodges. «Discriminatory analysis. Nonparametric discrimination: Consistency properties». In: *International Statistical Review/Revue Internationale de Statistique* 57.3 (1989), pp. 238–247 (cit. on p. 14).

[29] Thomas Cover and Peter Hart. «Nearest neighbor pattern classification». In: *IEEE transactions on information theory* 13.1 (1967), pp. 21–27 (cit. on p. 14).

[30] Gongde Guo, Hui Wang, David Bell, Yaxin Bi, and Kieran Greer. «KNN model-based approach in classification». In: *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2003, Catania, Sicily, Italy, November 3-7, 2003. Proceedings*. Springer. 2003, pp. 986–996 (cit. on p. 15).

[31] VB Prasath, Haneen Arafat Abu Alfeilat, Ahmad Hassanat, Omar Lasassmeh, Ahmad S Tarawneh, Mahmoud Bashir Alhasanat, and Hamzeh S Eyal Salman. «Distance and similarity measures effect on the performance of K-Nearest Neighbor classifier–A review». In: *arXiv preprint arXiv:1708.04321* (2017) (cit. on p. 15).

[32] Vladimir Vapnik and A Ya Chervonenkis. «A class of algorithms for pattern recognition learning». In: *Avtomat. i Telemekh* 25.6 (1964), pp. 937–945 (cit. on p. 15).

[33] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. «A training algorithm for optimal margin classifiers». In: *Proceedings of the fifth annual workshop on Computational learning theory*. 1992, pp. 144–152 (cit. on p. 15).

[34] Dustin Boswell. «Introduction to support vector machines». In: *Departement of Computer Science and Engineering University of California San Diego* 11 (2002) (cit. on pp. 16, 17).

[35] Tin Kam Ho. «Random decision forests». In: *Proceedings of 3rd international conference on document analysis and recognition*. Vol. 1. IEEE. 1995, pp. 278–282 (cit. on p. 18).

[36] Leo Breiman. «Random forests». In: *Machine learning* 45 (2001), pp. 5–32 (cit. on p. 18).

[37] Adele Cutler and Leo Breiman. «Archetypal analysis». In: *Technometrics* 36.4 (1994), pp. 338–347 (cit. on p. 18).

[38] Lior Rokach. *Pattern classification using ensemble methods*. Vol. 75. World Scientific, 2010 (cit. on p. 18).

[39] Muhammad Yaseen Khan, Abdul Qayoom, Muhammad Suffian Nizami, Muhammad Shoaib Siddiqui, Shaukat Wasi, and Syed Muhammad Khaliq-ur-Rahman Raazi. «Automated prediction of Good Dictionary EXamples (GDEX): a comprehensive experiment with distant supervision, machine learning, and word embedding-based deep learning techniques». In: *Complexity* 2021 (2021), pp. 1–18 (cit. on p. 18).

[40] Yoav Freund and Robert E Schapire. «A decision-theoretic generalization of on-line learning and an application to boosting». In: *Journal of computer and system sciences* 55.1 (1997), pp. 119–139 (cit. on p. 18).

[41] Yoav Freund, Robert E Schapire, et al. «Experiments with a new boosting algorithm». In: *icml*. Vol. 96. Citeseer. 1996, pp. 148–156 (cit. on p. 18).

[42] Tianqi Chen and Carlos Guestrin. «Xgboost: A scalable tree boosting system». In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 2016, pp. 785–794 (cit. on p. 19).

[43] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016 (cit. on p. 19).

[44] Michael A Arbib. *The handbook of brain theory and neural networks*. MIT press, 2003 (cit. on p. 19).

[45] Frank Rosenblatt. «The perceptron: a probabilistic model for information storage and organization in the brain.» In: *Psychological review* 65.6 (1958), p. 386 (cit. on p. 20).

[46] Bohdan Macukow. «Neural networks–state of art, brief history, basic models and architecture». In: *Computer Information Systems and Industrial Management: 15th IFIP TC8 International Conference, CISIM 2016, Vilnius, Lithuania, September 14-16, 2016, Proceedings 15*. Springer. 2016, pp. 3–14 (cit. on p. 20).

[47] John J Hopfield. «Neural networks and physical systems with emergent collective computational abilities.» In: *Proceedings of the national academy of sciences* 79.8 (1982), pp. 2554–2558 (cit. on p. 20).

[48] David E Rumelhart, Geoffrey E Hinton, Ronald J Williams, et al. *Learning internal representations by error propagation*. 1985 (cit. on p. 20).

[49] Gail A Carpenter and Stephen Grossberg. «A massively parallel architecture for a self-organizing neural pattern recognition machine». In: *Computer vision, graphics, and image processing* 37.1 (1987), pp. 54–115 (cit. on p. 20).

[50] Pérez-Enciso and Laura Zingaretti. «A Guide for Using Deep Learning for Complex Trait Genomic Prediction». In: *Genes* 10 (July 2019), p. 553. DOI: 10.3390/genes10070553 (cit. on p. 20).

[51] Bo Zhang. «Stress recognition from heterogeneous data». PhD thesis. Université de Lorraine, 2017 (cit. on p. 21).

[52] Christos D Katsis, George Ganiatsas, and Dimitrios I Fotiadis. «An integrated telemedicine platform for the assessment of affective physiological states». In: *Diagnostic pathology* 1 (2006), pp. 1–9 (cit. on p. 21).

[53] Cornelia Setz, Bert Arnrich, Johannes Schumm, Roberto La Marca, Gerhard Tröster, and Ulrike Ehlert. «Discriminating stress from cognitive load using a wearable EDA device». In: *IEEE Transactions on information technology in biomedicine* 14.2 (2009), pp. 410–417 (cit. on pp. 21, 22, 24).

[54] Zongmin Wei, Damin Zhuang, Xiaoru Wanyan, Chen Liu, and Huan Zhuang. «A model for discrimination and prediction of mental workload of aircraft cockpit display interface». In: *Chinese Journal of Aeronautics* 27.5 (2014), pp. 1070–1077 (cit. on p. 21).

[55] Peyvand Ghaderyan, Ataollah Abbasi, and Afshin Ebrahimi. «Time-varying singular value decomposition analysis of electrodermal activity: A novel method of cognitive load estimation». In: *Measurement* 126 (2018), pp. 102–109 (cit. on p. 21).

[56] William L Romine, Noah L Schroeder, Josephine Graft, Fan Yang, Reza Sadeghi, Mahdieh Zabihimayvan, Dipesh Kadariya, and Tanvi Banerjee. «Using machine learning to train a wearable device for measuring students' cognitive load during problem-solving activities based on electrodermal activity, body temperature, and heart rate: Development of a cognitive load tracker for both personal and classroom use». In: *Sensors* 20.17 (2020), p. 4833 (cit. on pp. 21–24).

[57] Adnan Ghaderi, Javad Frounchi, and Alireza Farnam. «Machine learning-based signal processing using physiological signals for stress detection». In: *2015 22nd Iranian conference on biomedical engineering (ICBME)*. IEEE. 2015, pp. 93–98 (cit. on p. 21).

[58] Kyung Hwan Kim, Seok Won Bang, and Sang Ryong Kim. «Emotion recognition system using short-term monitoring of physiological signals». In: *Medical and biological engineering and computing* 42 (2004), pp. 419–427 (cit. on p. 21).

[59] Jing Zhai and Armando Barreto. «Stress detection in computer users based on digital signal processing of noninvasive physiological variables». In: *2006 international conference of the IEEE engineering in medicine and biology society*. IEEE. 2006, pp. 1355–1358 (cit. on p. 21).

[60] Christos D Katsis, Nikolaos Katertsidis, George Ganiatsas, and Dimitrios I Fotiadis. «Toward emotion recognition in car-racing drivers: A biosignal processing approach». In: *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* 38.3 (2008), pp. 502–512 (cit. on pp. 21, 23).

[61] Seyyed Abed Hosseini and Mohammad Bagher Naghibi-Sistani. «Classification of emotional stress using brain activity». In: *Applied Biomedical Engineering* 7 (2011), pp. 32–41 (cit. on pp. 21, 23).

[62] Hindra Kurniawan, Alexandr V Maslov, and Mykola Pechenizkiy. «Stress detection from speech and galvanic skin response signals». In: *Proceedings of the 26th IEEE international symposium on computer-based medical systems*. IEEE. 2013, pp. 209–214 (cit. on pp. 21, 23, 24).

[63] Daniel McDuff, Sarah Gontarek, and Rosalind Picard. «Remote measurement of cognitive stress via heart rate variability». In: *2014 36th annual international conference of the IEEE engineering in medicine and biology society*. IEEE. 2014, pp. 2957–2960 (cit. on p. 21).

[64] Nandita Sharma and Tom Gedeon. «Modeling a stress signal». In: *Applied Soft Computing* 14 (2014), pp. 53–61 (cit. on pp. 21, 22, 24).

[65] Xiyuan Hou, Yisi Liu, Olga Sourina, Yun Rui Eileen Tan, Lipo Wang, and Wolfgang Mueller-Wittig. «EEG based stress monitoring». In: *2015 IEEE international conference on systems, man, and cybernetics*. IEEE. 2015, pp. 3110–3115 (cit. on pp. 21, 22).

[66] Fares Al-Shargie, Masashi Kiguchi, Nasreen Badruddin, Sarat C Dass, Ahmad Fadzil Mohammad Hani, and Tong Boon Tang. «Mental stress assessment using simultaneous measurement of EEG and fNIRS». In: *Biomedical optics express* 7.10 (2016), pp. 3882–3898 (cit. on p. 21).

[67] Choubeila Maaoui, Frédéric Bousefsaf, and Alain Pruski. «Automatic human stress detection based on webcam photoplethysmographic signals». In: *Journal of Mechanics in Medicine and Biology* 16.04 (2016), p. 1650039 (cit. on p. 21).

[68] Giorgos Giannakakis, Matthew Pediaditis, Dimitris Manousos, Eleni Kazantzaki, Franco Chiarugi, Panagiotis G Simos, Kostas Marias, and Manolis Tsiknakis. «Stress and anxiety detection using facial cues from videos». In: *Biomedical Signal Processing and Control* 31 (2017), pp. 89–101 (cit. on pp. 22, 23).

[69] Reza Khosrowabadi. «Stress and perception of emotional stimuli: Long-term stress rewiring the brain». In: *Basic and clinical neuroscience* 9.2 (2018), p. 107 (cit. on p. 22).

[70] Amandeep Cheema and Mandeep Singh. «An application of phonocardiography signals for psychological stress detection using non-linear entropy based features in empirical mode decomposition domain». In: *Applied Soft Computing* 77 (2019), pp. 24–33 (cit. on p. 22).

[71] Likun Xia, Aamir Saeed Malik, and Ahmad Rauf Subhani. «A physiological signal-based method for early mental-stress detection». In: *Cyber-Enabled Intelligence*. Taylor & Francis, 2019, pp. 259–289 (cit. on p. 22).

[72] Nargess Nourbakhsh, Yang Wang, and Fang Chen. «GSR and blink features for cognitive load classification». In: *Human-Computer Interaction–INTERACT 2013: 14th IFIP TC 13 International Conference, Cape Town, South Africa, September 2-6, 2013, Proceedings, Part I 14*. Springer. 2013, pp. 159–166 (cit. on pp. 22, 23, 25).

[73] Igor Mijić, Marko Šarlija, and Davor Petrinović. «MMOD-COG: A database for multimodal cognitive load classification». In: *2019 11th international symposium on image and signal processing and analysis (ispa)*. IEEE. 2019, pp. 15–20 (cit. on pp. 22, 25).

[74] Shaibal Barua, Mobyen Uddin Ahmed, and Shahina Begum. «Towards intelligent data analytics: A case study in driver cognitive load classification». In: *Brain sciences* 10.8 (2020), p. 526 (cit. on pp. 22, 23).

[75] Martin Gjoreski, Bhargavi Mahesh, Tine Kolenik, Jens Uwe-Garbas, Dominik Seuss, Hristijan Gjoreski, Mitja Luštrek, Matjaž Gams, and Veljko Pejović. «Cognitive load monitoring with wearables–lessons learned from a machine learning challenge». In: *IEEE Access* 9 (2021), pp. 103325–103336 (cit. on pp. 22, 23, 25).

[76] Aamir Arsalan, Muhammad Majid, Syed Muhammad Anwar, and Ulas Bagci. «Classification of perceived human stress using physiological signals». In: *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE. 2019, pp. 1247–1250 (cit. on pp. 22, 23, 26).

[77] Sawon Pratiher, Ananth Radhakrishnan, Karuna P Sahoo, Sazedul Alam, Scott E Kerick, Nirmalya Ghosh, et al. «Classification of VR-gaming difficulty induced stress levels using physiological (EEG & ECG) signals and machine learning». In: *Authorea Preprints* (2023) (cit. on pp. 22–24, 26).

[78] Christian Herff, Dominic Heger, Ole Fortmann, Johannes Hennrich, Felix Putze, and Tanja Schultz. «Mental workload during n-back task—quantified in the prefrontal cortex using fNIRS». In: *Frontiers in human neuroscience* 7 (2014), p. 935 (cit. on p. 22).

[79] Jennifer A Healey and Rosalind W Picard. «Detecting stress during real-world driving tasks using physiological sensors». In: *IEEE Transactions on intelligent transportation systems* 6.2 (2005), pp. 156–166 (cit. on p. 22).

[80] Brian R Nhan and Tom Chau. «Classifying affective states using thermal infrared imaging of the human face». In: *IEEE Transactions on Biomedical Engineering* 57.4 (2009), pp. 979–987 (cit. on pp. 22, 24).

[81] Dimitris Giakoumis, Anastasios Drosou, Pietro Cipresso, Dimitrios Tzovaras, George Hassapis, Andrea Gaggioli, and Giuseppe Riva. «Using activity-related behavioural features towards more effective automatic stress detection». In: (2012) (cit. on p. 22).

[82] Paolo Melillo, Marcello Bracale, and Leandro Pecchia. «Nonlinear Heart Rate Variability features for real-life stress detection. Case study: students under stress due to university examination». In: *Biomedical engineering online* 10 (2011), pp. 1–13 (cit. on p. 22).

[83] Jesus Minguillon, Eduardo Perez, Miguel Angel Lopez-Gordo, Francisco Pelayo, and Maria Jose Sanchez-Carrion. «Portable system for real-time detection of stress level». In: *Sensors* 18.8 (2018), p. 2504 (cit. on p. 22).

[84] P Karthikeyan, M Murugappan, and Sazali Yaacob. «EMG signal based human stress level classification using wavelet packet transform». In: *Trends in Intelligent Robotics, Automation, and Manufacturing: First International Conference, IRAM 2012, Kuala Lumpur, Malaysia, November 28-30, 2012. Proceedings*. Springer. 2012, pp. 236–243 (cit. on p. 22).

[85] Jacqueline Wijsman, Bernard Grundlehner, Hao Liu, Hermie Hermens, and Julien Penders. «Towards mental stress detection using wearable physiological sensors». In: *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE. 2011, pp. 1798–1801 (cit. on pp. 22, 24).

[86] AS Anusha, Joy Jose, SP Preejith, Joseph Jayaraj, and Sivaprakasam Mohanasankar. «Physiological signal based work stress detection using unobtrusive sensors». In: *Biomedical Physics & Engineering Express* 4.6 (2018), p. 065001 (cit. on pp. 22, 24).

[87] Awais Gul Airij, Rubita Sudirman, and Usman Ullah Sheikh. «GSM and GPS based real-time remote physiological signals monitoring and stress levels classification». In: *2018 2nd International Conference on BioSignal Analysis, Processing and Systems (ICBAPS)*. IEEE. 2018, pp. 130–135 (cit. on pp. 22, 23).

[88] Marco Pedrotti, Mohammad Ali Mirzaei, Adrien Tedesco, Jean-Rémy Chardonnet, Frédéric Mérienne, Simone Benedetto, and Thierry Baccino. «Automatic stress classification with pupil diameter analysis». In: *International Journal of Human-Computer Interaction* 30.3 (2014), pp. 220–236 (cit. on p. 22).

[89] Dorien Huysmans, Elena Smets, Walter De Raedt, Chris Van Hoof, Katleen Bogaerts, Ilse Van Diest, and Denis Helic. «Unsupervised learning for mental stress detection-exploration of self-organizing maps». In: *Proc. of Biosignals 2018* 4 (2018), pp. 26–35 (cit. on pp. 22, 24).

[90] Zhong Yin and Jianhua Zhang. «Cross-session classification of mental workload levels using EEG and an adaptive deep learning model». In: *Biomedical Signal Processing and Control* 33 (2017), pp. 30–47 (cit. on pp. 22, 26).

[91] Soo-Yeon Han, No-Sang Kwak, Taegeun Oh, and Seong-Whan Lee. «Classification of pilots' mental states using a multimodal deep learning network». In: *Biocybernetics and Biomedical Engineering* 40.1 (2020), pp. 324–336 (cit. on pp. 22, 26).

[92] Berina Alić, Dijana Sejdinović, Lejla Gurbeta, and Almir Badnjevic. «Classification of stress recognition using artificial neural network». In: *2016 5th Mediterranean Conference on Embedded Computing (MECO)*. IEEE. 2016, pp. 297–300 (cit. on p. 22).

[93] Anum Asif, Muhammad Majid, and Syed Muhammad Anwar. «Human stress classification using EEG signals in response to music tracks». In: *Computers in biology and medicine* 107 (2019), pp. 182–196 (cit. on pp. 22, 23).

[94] Jing Huang, Yu Liu, and Xiaoyan Peng. «Recognition of driver's mental workload based on physiological signals, a comparative study». In: *Biomedical Signal Processing and Control* 71 (2022), p. 103094 (cit. on pp. 23, 24, 26).

[95] Wenhui Liao, Weihong Zhang, Zhiwei Zhu, and Qiang Ji. «A real-time human stress monitoring system using dynamic bayesian network». In: *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)-workshops*. IEEE. 2005, pp. 70–70 (cit. on p. 23).

[96] Mi-hee Lee, Gyunghye Yang, H-K Lee, and Seokwon Bang. «Development stress monitoring system based on personal digital assistant (PDA)». In: *The 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. Vol. 1. IEEE. 2004, pp. 2364–2367 (cit. on p. 23).

[97] Nichakorn Pongsakornsathien, Yixiang Lim, Alessandro Gardi, Samuel Hilton, Lars Planke, Roberto Sabatini, Trevor Kistan, and Neta Ezer. «Sensor networks for aerospace human-machine systems». In: *Sensors* 19.16 (2019), p. 3465 (cit. on p. 23).

[98] Mohammad Ali Khalilzadeh, Seyyed Mehran Homam, SEYED ABED HOSSEINI, and Vahid Niazmand. «Qualitative and quantitative evaluation of brain activity in emotional stress». In: (2010) (cit. on p. 23).

[99] Palanisamy Karthikeyan, Murugappan Murugappan, and Sazali Yaacob. «Descriptive analysis of skin temperature variability of sympathetic nervous system activity in stress». In: *Journal of Physical Therapy Science* 24.12 (2012), pp. 1341–1344 (cit. on p. 23).

[100] Olympia Simantiraki, Giorgos Giannakakis, Anastasia Pampouchidou, and Manolis Tsiknakis. «Stress detection from speech using spectral slope measurements». In: *Pervasive Computing Paradigms for Mental Health: Selected Papers from MindCare 2016, Fabulous 2016, and IIoT 2015 3*. Springer. 2018, pp. 41–50 (cit. on p. 23).

[101] Ryan McKendrick, Bradley Feest, Amanda Harwood, and Brian Falcone. «Theories and methods for labeling cognitive workload: Classification and transfer learning». In: *Frontiers in human neuroscience* 13 (2019), p. 295 (cit. on pp. 23, 24).

[102] Peng Ren, Armando Barreto, Ying Gao, and Malek Adjouadi. «Affective assessment by digital processing of the pupil diameter». In: *IEEE Transactions on Affective computing* 4.1 (2012), pp. 2–14 (cit. on p. 23).

[103] Daniel J McDuff, Javier Hernandez, Sarah Gontarek, and Rosalind W Picard. «Cogcam: Contact-free measurement of cognitive stress during computer tasks with a digital camera». In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 2016, pp. 4000–4004 (cit. on p. 23).

[104] Serdar Baltaci and Didem Gokcay. «Stress detection in human–computer interaction: Fusion of pupil dilation and facial temperature features». In: *International Journal of Human–Computer Interaction* 32.12 (2016), pp. 956–966 (cit. on pp. 23, 24).

[105] Gaël Vila, Christelle Godin, Sylvie Charbonnier, Etienne Labyt, Oumayma Sakri, and Aurélie Campagne. «Pressure-specific feature selection for acute stress detection from physiological recordings». In: *2018 IEEE international conference on systems, man, and cybernetics (SMC)*. IEEE. 2018, pp. 2341–2346 (cit. on p. 24).

[106] Niloofar Momeni, Fabio Dell'Agnola, Adriana Arza, and David Atienza. «Real-time cognitive workload monitoring based on machine learning using physiological signals in rescue missions». In: *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE. 2019, pp. 3779–3785 (cit. on pp. 24, 26).

[107] Talha Iqbal, Adnan Elahi, William Wijns, and Atif Shahzad. «Exploring unsupervised machine learning classification methods for physiological stress detection». In: *Frontiers in Medical Technology* 4 (2022), p. 782756 (cit. on pp. 24, 26).

[108] Brian D Womack and John HL Hansen. «Classification of speech under stress using target driven features». In: *Speech Communication* 20.1-2 (1996), pp. 131–150 (cit. on p. 24).

[109] Guojun Zhou, John HL Hansen, and James F Kaiser. «Nonlinear feature based classification of speech under stress». In: *IEEE Transactions on speech and audio processing* 9.3 (2001), pp. 201–216 (cit. on p. 24).

[110] Sumitra Shukla, Samarendra Dandapat, and SRM Prasanna. «Spectral slope based analysis and classification of stressed speech». In: *International Journal of Speech Technology* 14 (2011), pp. 245–258 (cit. on p. 24).

[111] Jacqueline Wijsman, Bernard Grundlehner, Hao Liu, Julien Penders, and Hermie Hermens. «Wearable physiological sensors reflect mental stress state in office-like situations». In: *2013 humaine association conference on affective computing and intelligent interaction*. IEEE. 2013, pp. 600–605 (cit. on p. 24).

[112] Zhong Yin, Mengyuan Zhao, Wei Zhang, Yongxiong Wang, Yagang Wang, and Jianhua Zhang. «Physiological-signal-based mental workload estimation via transfer dynamical autoencoders in a deep learning framework». In: *Neurocomputing* 347 (2019), pp. 212–229 (cit. on p. 26).

[113] Nandita Sharma and Tom Gedeon. «Objective measures, sensors and computational techniques for stress recognition and classification: A survey». In: *Computer methods and programs in biomedicine* 108.3 (2012), pp. 1287–1301 (cit. on p. 26).

[114] J Ridley Stroop. «Studies of interference in serial verbal reactions.» In: *Journal of experimental psychology* 18.6 (1935), p. 643 (cit. on p. 27).

[115] JHM Tulen, P Moleman, HG Van Steenis, and F Boomsma. «Characterization of stress reactions to the Stroop Color Word Test». In: *Pharmacology Biochemistry and Behavior* 32.1 (1989), pp. 9–15 (cit. on p. 27).

[116] Michael J Kane, Andrew RA Conway, Timothy K Miura, and Gregory JH Colflesh. «Working memory, attention control, and the N-back task: a question of construct validity.» In: *Journal of Experimental psychology: learning, memory, and cognition* 33.3 (2007), p. 615 (cit. on p. 27).

[117] Lukas Arts and Egon L Van Den Broek. «Biosignal Quality Control in Real-World Intelligent Environments». In: *Workshop Proceedings of the 19th International Conference on Intelligent Environments (IE2023)*. IOS Press. 2023, pp. 24–33 (cit. on p. 28).

[118] Hamed Rahimi Nasrabadi and Jose-Manuel Alonso. «Modular streaming pipeline of eye/head tracking data using Tobii Pro Glasses 3». In: *bioRxiv* (2022), pp. 2022–09 (cit. on p. 28).

[119] Laura Bajardi. «Multisignal approach for stress and workload analysis». PhD thesis. Politecnico di Torino, 2022 (cit. on p. 29).

[120] Sebastian Raschka. *Python machine learning*. Packt publishing ltd, 2015 (cit. on p. 31).

[121] Roi Yehoshua. *Pipelines in Scikit-Learn*. `https://medium.com/ai-made-simple/pipelines-in-scikit-learn-46c61c5c60b2`. Accessed: 01/03/2024. Mar. 2023 (cit. on p. 37).

[122] F. Pedregosa et al. «Scikit-learn: Machine Learning in Python». In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830 (cit. on pp. 38, 39).

[123] Ilias Tougui, Abdelilah Jilbab, and Jamal El Mhamdi. «Impact of the choice of cross-validation techniques on the results of machine learning-based diagnostic applications». In: *Healthcare informatics research* 27.3 (2021), p. 189 (cit. on p. 40).