Signal Classification Based on a Hybrid Approach of Supervised and Unsupervised Machine Learning

Gregory M. Vergilino - a61451

Thesis presented to the School of Technology and Management at the Polytechnic Institute of Bragança for the Master's Degree in **Eletrical and Computer Engineering** within the scope of the **Double Degree Program** with the Federal Technological University of Paraná in Control and Automation Engineering.

Work oriented by:
Prof. Ana I. Pereira
Prof. Uilian J. Dreyer
Prof. José Lima

Bragança
2024

Signal Classification Based on a Hybrid Approach of Supervised and Unsupervised Machine Learning

Gregory M. Vergilino - a61451

Thesis presented to the School of Technology and Management at the Polytechnic Institute of Bragança for the Master's Degree in **Eletrical and Computer Engineering** within the scope of the **Double Degree Program** with the Federal Technological University of Paraná in Control and Automation Engineering.

Work oriented by:
Prof. Ana I. Pereira
Prof. Uilian J. Dreyer
Prof. José Lima

Bragança
2024

Dedication

God blesses everyone. And I will always be grateful to Him for taking care of me in every step of my life. He is the Alpha, the Omega and Beyond. Amen!

To my beloved family, those who taught me the true meaning of love. But specially to my brother , I want to dedicate in order to inspire him to pursue his dreams as I did. I believe in all your efforts and know you can be even better me.

Acknowledgement

First, I want to acknowledge how important God is in my life. Without Him, I would not have achieved everything I have so far. He means everything to me — my past, my future, and my faith.

To the most important people in my life, my dad and mom, I want to express my deepest gratitude for showing me the path of honor and truth. I could not have asked for a better family, including my brother, for whom I have deep admiration.

To my friends, I do not have enough words to express my appreciation. Lucas Scriptore, my true friend, who supported me throughout this journey, studying with me both in person and on Discord, and always helping me through the chaos. Felipe Bueno, thank you very much for the time we spent together and for all the support, and I am thankful to Henrique as well. To my friends in Brazil, Luis, Maria and Samara, I express the same warmest and heartfelt thanks, always supporting my progress.

I would also like to thank both universities for the opportunity to complete this project, for all the financial support through the scholarship, and for the pedagogical guidance provided with excellence. Especially to Professor Ana Pereira, thank you for your patience, even when the unexpected happened. Your guidance was exceptional! Thank you so much.

Abstract

The main challenge in using optical sensors to collect data for automated prosthesis control lies in accurately predicting which finger is moving based solely on forearm signals. This is a complex task in the design of intelligent prosthetic systems, which must be both efficient and adaptive. However, improving the quality of life for individuals with motor disabilities demands reliable interpretation of such biosignals.

This work proposes the use of machine learning algorithms to address this problem. In this research was used a dataset of signal acquired with a Fiber Bragg Grating sensor positioned on the forearm, on a group of ten patients. The group were asked realize some finger movements in order to gather data. The main problem is to identify which movement is being realized without labeling the signal. In this research will be analyzed methods to apply label on the data and classify them. The focus was to get a precise hybrid approach of supervised and unsupervised methods. k-Means was used as an unsupervised machine learning method to group similar data into distinct clusters and label the data. Random Forest was used as supervised learning algorithms to classify the data after labeling.

Keywords: Clustering, Biosignals, k-Means, Random Forest.

Resumo

O principal desafio no uso de sensores ópticos para coletar dados para o controle automatizado de próteses reside na previsão precisa de qual dedo está se movendo com base apenas nos sinais do antebraço. Esta é uma tarefa complexa no projeto de sistemas protéticos inteligentes, que devem ser eficientes e adaptáveis. No entanto, melhorar a qualidade de vida de indivíduos com deficiências motoras exige uma interpretação confiável desses biossinais.

Este trabalho propõe o uso de algoritmos de aprendizado de máquina para abordar esse problema. Nesta pesquisa, foi utilizado um conjunto de dados de sinais adquiridos com sensor de Rede de Bragg em Fibra Óptica colocado no antebraço, em um grupo de dez pacientes. O grupo foi solicitado a realizar alguns movimentosde de dedos para coletar dados. O principal problema é identificar qual movimento está sendo realizado sem rotular o sinal. Nesta pesquisa, serão analisados métodos para aplicar rótulos aos dados e classificá-los. O foco foi obter uma abordagem híbrida precisa de métodos supervisionados e não supervisionados. O k-Means foi utilizado como um método de aprendizado de máquina não supervisionado para agrupar dados semelhantes em clusters distintos e rotulá-los. O algoritmo Random Forest foi utilizado como algoritmo de aprendizado supervisionado para classificar os dados depois de rotulados.

Palavras-chave: Clusterização, Biossinais, k-Means, Random Forest.

Contents

List of Tables

List of Figures

Acronyms

AI Artificial Intelligence.

BIRCH Balanced Iterative Reducing and Clustering using Hierarchies.

DBSCAN Density-Based Spatial Clustering of Applications with Noise.

DT Decision Tree.

FBG Fiber Bragg Grating.

KNN k-Nearest Neighbors.

L2OCV Leave-2-Out Cross Validation.

RF Random Forest.

SVM Support Vector Machine.

WCSS Within-Cluster Sum of Squares.

Chapter 1

Introduction

According to Mayer [1], the contemporary world is experiencing an overwhelming surge of data, representing several phenomena and aspects of society. Badman [2] defines Big Data as massive, complex data sets that traditional data management systems cannot handle. This concept is related to the collection, integration, management, storage, and analysis of large volumes of data from real-world signals and digital records, with the goal of transforming them into useful information for humanity.

As Big Data continues in exponential expansion, the answer to these limitations is the beginning of advanced computational methods. In this context, artificial intelligence has become essential. Among them , machine learning stands out as a powerful approach for uncovering patterns, making predictions, and enabling adaptive systems in data-intensive environments [3] [4].

In that scenario, machine learning algorithms focuses on building systems that learn and improve as they consume more data. It can extract, analyze, and identify patterns in large datasets faster than traditional statistical methods, enabling more accurate predictions and data-driven decision-making. Once the algorithm understands how to process the data, the machines could handle it autonomously [5].

Among the many domains that benefit from machine learning algorithms, the biomedical field has gained particular prominence due to the complexity and volume of physiological signals. Biosignals or biomechanical movement present a huge amount of information, such as those derived from muscle activity. By analyzing electrical signals related to muscle movements, it is possible to obtain detailed information about muscle activation [6]. Through pattern recognition, these signals can be used in a wide range of applications, including prosthetic control, rehabilitation monitoring, and assistive technology development [7].

Considering prosthetic devices, an efficient classification of biomedical signals is essential for the development of more functional prosthetic systems. The use of machine learning techniques has proven to be the most precise approach to reliably distinguishing and interpreting these signals, enabling a fast and adaptive response from the prosthesis [8], [9].

This research builds on a previous study by Júnior et al. [9], which developed an automated prosthesis using Fiber Bragg Grating (FBG) sensors. According to the authors [7], all current cases in this area use a lot of sensors. That is why they started the project studying the possibility of applying FBG sensors to acquire functional data of finger movements from forearm tension. FBG sensors are optical sensors with a very high sensitivity to strain and temperature variations, making them particularly suitable for capturing subtle biomechanical changes.

The previous study focused on labeling the acquired signals and applying supervised machine learning techniques to classify finger movements [7]. Gathering a great achievement for the project, allowing to apply on the automated prosthesis [9]. Building upon these findings, the present research proposes a hybrid methodology incorporating unsupervised learning, and applying it to unlabeled data, with supervised learning algorithms for classifying.

1.1 Objectives

In order to carry out this project and contribute with useful research. This study will seek to improve the precision and responsiveness of classification methods.

Analyzing a biosignal to apply in an integrated machine learning system is not that simple. Respecting the best manners to build an efficient deep learning and avoid future problems, it is extremely important to have a well parameterized aim.

The general objective of this research is to develop and evaluate a method for classifying hand movements based on signals acquired by FBG sensors. Building upon previous work conducted by [9], this study seeks to improve the segmentation and classification of finger movements, addressing issues related to potential overfitting observed in earlier experiments [7]. The goal is to process the available data effectively, ensuring that each movement pattern is correctly identified and distinguished using machine learning techniques.

The specific objectives outlined below were established to guide the development of this research and to ensure a structured approach to achieving the general objective. Each objective corresponds to a key step in the methodological process, from the initial exploration of the data to the evaluation and comparison of classification techniques. This structure allows for a comprehensive analysis of the signal behavior and the effectiveness of different machine learning approaches applied to FBG based dataset. They are as follows:

- Apply unsupervised clustering techniques to group patient signal data based on temporal and structural similarities;
- Compare several techniques of clustering;
- Analyze and characterize the extracted data in order to understand its statistical and dynamic properties;
- Implement a supervised learning algorithm to classify the movements based on the clustered and processed data;
- Compare the performance of the proposed method with that of a conventional neural network model.

1.2 Thesis Structure

This document has been structured with the aim of guiding the reader through a good reading process and to facilitate a comprehensive understanding of the research project discussed. For this reason, the document is divided into distinct chapters, each one focused on elucidating a specific aspect of the overall narrative of the study. In the currently section, a brief introduction to each of these chapters will be presented.

Chapter 1 Introduction, will lead to the purpose of this research. First, a major scenario is presented which involves a specific problem to be solved or questioned about. The justification that lead the project were settled. Then the main objective and specifics objectives were traced in order to follow a line logic. Finally, an overview of all the structure.

Chapter 2 State of the Art, will introduces the reader to the most recent projects in the area that the research is inserted. Also, it will be explained about all tools used during the project and some existent methods to compare.

Chapter 3 Methodology, will settle the path used during this study. It will be described the advantages and disadvantages in the process, this includes the major problems faced.

Chapter 4 Case Study, presents the database of raw signal captured by FBG sensor and identify the main characteristics this signal has.

Chapter 5 Results and Discussion, presents and analyzes the results obtained throughout the experimentation phase. It includes signal preprocessing steps, clustering and classification results, visualizations, and performance metrics. Special attention is paid to the interpretation of the findings, comparing different approaches and validating the effectiveness of the proposed methodology.

Chapter 6 Conclusions and Future Work, summarizes the key contributions of the research, highlights its limitations, and outlines directions for future work.

Chapter 2

State of the Art

This chapter presents an overview of the main theoretical concepts and technical tools that support the development of this research. Its objective is to provide the background necessary to facilitate a clear and comprehensive understanding of the topics discussed throughout the study.

The procedure performed during data acquisition was to ask the patient to move the finger up and down five times. It was repeated for the index, middle, ring, and little. In this section will be shown the muscles that are flexing in these gestures Section 2.1. And how the sensors will capture their signal Section 2.2, presents a review on FBG sensors, which are known for their high sensitivity to strain and temperature, and their growing application in biomedical signal acquisition and wearable technologies [10], [11].

2.1 Forearm Muscle Activation During Finger Movement

Firstly, this section introduces the importance of each muscle involved during the experimentation. Data were collected by instructing each participant to lift and lower the four fingers (index, middle, ring, and little), as mentioned before. While signals were gathered from forearm muscles. These movements were performed with the subject's forearm in a resting position and the sensors placed around the forearm like bracelets. The thumb was excluded from the analysis [8], [12].

Although the fingers themselves do not contain muscles [13], their movements are controlled by extrinsic muscles located in the forearm, as illustrated in Figure 2.1.

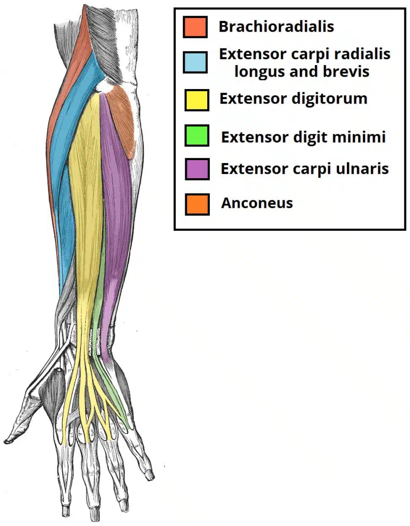

Figure 2.1: Forearm anatomy [13]

These muscles exert force through tendons that cross the wrist and insert into the fingers [13], [14]. During finger flexion, the primary muscles involved are [14]:

- **Flexor digitorum superficialis (FDS)** – flexes the proximal interphalangeal (PIP) joints of fingers index to little;
- **Flexor digitorum profundus (FDP)** – flexes the distal interphalangeal (DIP) joints, particularly active during full finger flexion;
- **Flexor carpi ulnaris and radialis** – assist with wrist stabilization during finger movement, especially relevant when fingers are moved individually.

When a subject individually lifts a finger, the corresponding extensor muscles located in the posterior compartment of the forearm become active [13]:

- ***Extensor digitorum*** – the main extensor of fingers (index to little);

- ***Extensor indicis*** – specifically assists in the extension of the index finger;
- ***Extensor digiti minimi*** – responsible for extending the little finger.

Each of these muscle activations causes subtle changes in the shape and tension of the forearm surface. These mechanical deformations are captured by FBG sensors positioned around the forearm [9], [12]. Because each finger involves a distinct combination and intensity of muscle activation, the resulting deformation patterns are sufficiently unique to allow classification using machine learning models such as Random Forest [8]. Understanding the distribution of these muscles is which are fundamental for monitoring and analyzing the finger extension movements [7].

2.2 Fiber Bragg Grating Sensor

The FBG sensor is a microstructure that gathers information through the reflection of light beams on a small segment of optical fiber. Hottinger Bruel [15] explains that the process involves transversely illuminating the fiber with a UV laser beam through a phase mask, generating an interference pattern that induces a permanent modification in the physical properties of the silica matrix. Any interference in the microstructure will affect the signal wavelength, as can be seen in Figure 2.2:

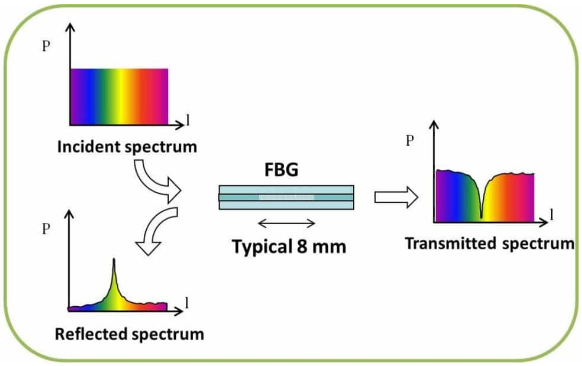

Figure 2.2: Working principle of FBG [16].

This exposure induces a permanent increase in the refractive index of the fiber's core, creating a fixed index modulation that follows the exposure pattern [16].

The Bragg wavelength, λ_B, is defined by $\lambda_B = 2n_{\text{eff}}\Lambda$, where n_{eff} is the effective refractive index of the fiber core, and Λ is the grating period. When the fiber undergoes mechanical strain or temperature changes, both n_{eff} and Λ are charged, obtaining in a measurable change in λ_B.

FBG sensors are widely used to monitor physical parameters such as strain and temperature. When broadband light travels through the fiber, the grating reflects a narrow wavelength band and transmits the remaining spectrum. This reflected wavelength is directly affected by changes in the fiber, making FBGs highly effective for structural health monitoring and other sensing applications.

The sensors used to measure the deformation of the forearm section were also applied in other works of this present project [7], [9], [12]. To obtain the dataset of patients, Pericles [9] and Kalinowski [12] proposed to use two elastic bracelets containing FBG sensors in the patient's forearm such as shown in Figure 2.3.

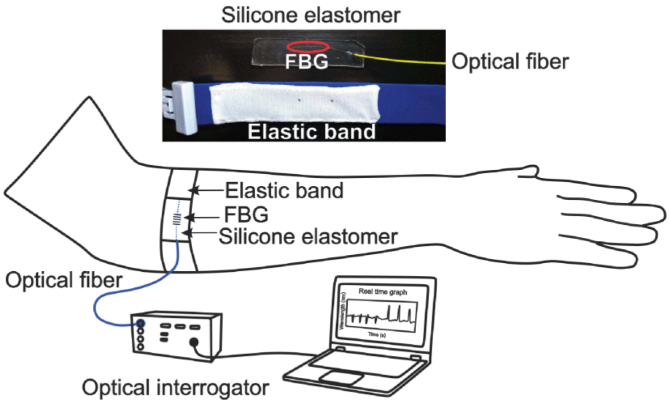

Figure 2.3: Schematic drawing [12].

FBG sensors offer several advantages, including immunity to electromagnetic interference, lightweight and compact form factor, high precision, and the ability to multiplex multiple sensors along a single optical fiber [7]. These features make them ideal for applications in aerospace, civil engineering, energy systems, and biomedical fields.

2.3 Machine Learning

Machine learning is a field in Artificial Intelligence (AI) that aims to teach computers how to learn from data. IBM [17] exploits it as a branch of AI focused on enabling computers and machines to imitate the way that humans learn, to perform tasks autonomously, and to improve their accuracy through experience and exposure to more data.

It has emerged as one of the most impactful technologies across various domains, from healthcare to finance and robotics [18]. Microsoft Azure [19] describes machine learning algorithms as pieces of code that help people explore, analyze, and find meaning in complex data sets. Each algorithm is a finite set of unambiguous step-by-step instructions that a machine can follow to achieve a certain goal. In a machine learning model, the goal is to establish or discover patterns that people can use to make predictions or categorize information.

Machine learning uses large datasets to train models. The more data used in training and with a good quality dataset, the more accurate will be the answer provided by the model [17]. The algorithm needs to receive meaningful information, that represents the reality. This information will lead the type of problem to inspect by the model, and next steps such as preparation and processing data [8].

It can be broken down into three main problems [17], [20]:

1. **Model Representation:** problems related to prediction or classification. It is based on database inputs, the algorithm will try to find a pattern in the data, which can be labeled or unlabeled, the algorithm will produce an estimate about a pattern in the data. This could be a linear function, a decision tree, a neural network, etc.

2. **Evaluation Function:** measures how well the model's predictions match the actual

results. It quantifies how far away a model's prediction is from the actual value, allowing the model to assess and improve its performance. If there are known examples, an error function can make a comparison to assess the accuracy of the model.This includes loss functions like mean squared error, cross-entropy, etc.

3. **A model optimization process:** problems related to estimation. If the model can fit better to the data points in the training set, then the weights are adjusted to reduce the discrepancy between the known example and the model estimate, usually through iterative methods like gradient descent.

After the type of problem is defined, the machine learning algorithm will be divided into methods to solve them. According to the University of California, Berkeley School of Information [20], there are four types of machine learning methods. They are separated as unsupervised learning, semi-supervised learning, supervised learning and reinforcement learning. Each is suited to different types of problem, as outlined below.

2.3.1 Unsupervised Learning

Unsupervised learning is a machine learning method that uses unlabeled data to train the model in order to find a pattern in them, using only input data without a target vector [8]. The purpose here is to uncover hidden structures, groups, or patterns that naturally emerge from the data itself, rather than relying on external supervision [21].

This type of machine learning focuses on a decision processes problems, where the amount labeled data are scarce or for exploratory insights [8]. In unsupervised learning, the datasets are typically analyzed using three main categories of algorithms [21]:

- **Association Rule Learning:** This type of unsupervised machine learning takes a rule-based approach to discovering interesting relationships between features in a given dataset. It works by using a measure of interest to identify strong rules found within a dataset.

- **Dimensionality Reduction:** These algorithms seek to transform data from high-dimensional spaces to low-dimensional spaces without compromising meaningful properties in the original data. These techniques are typically deployed during exploratory data analysis (EDA) or data processing to prepare the data for modeling.
- **Clustering:** It is a problem that the main objective is to group unlabeled data based on inherent similarities or differences among data points. The goal of clustering is to identify patterns and relationships in the data without any prior knowledge of the data's meaning.

2.3.2 Clustering algorithms

Following the clustering technique, its algorithms have some peculiarities that describes their approach. According to the Google Cloud, a platform developers and IA, it is broken in this types of algorithms:

- **Exclusive clustering:** each data point is assigned to one and only one cluster.
- **Overlapping clustering:** this technique allows a data point to belong to multiple clusters simultaneously, typically with varying degrees of membership. It reflects uncertainty or shared characteristics between groups.
- **Hierarchical clustering:** This method constructs a multi-level hierarchy of clusters by either successively merging smaller clusters (agglomerative) or dividing larger ones (divisive). Similar data points are grouped together at different levels of granularity, forming a tree-like structure.
- **Probabilistic clustering:** where each data point is associated with a probability of belonging to each cluster. This approach is different from other methods, which group data points based on similarities to others in a cluster.

.

As Ana [8] highlights, the preparation and processing of data are extremely important steps in building a good machine learning model. Then, more about the various methods that do clustering will be explored below.

k-Means

k-Means algorithm is the most widely used clustering method [22]. It partitions the data into k distinct clusters, where k defines the number of groups and the granularity of segmentation. As the number of k is already known it will simplify the classification [21].

First, it is defined the optimal number of clusters k for the model. This step is made using a technique called Elbow Method, a graphical method to select the number o cluster that will suit better for the model. After this, the processing is performed. The algorithm will set a random number as the first centroid of each cluster. The process repeated until the number of iterations chosen is completed.

Affinity Propagation

This method does not require a predefined number of clusters. It identifies exemplars by exchanging real-valued messages between data points, determining clusters based on data similarity. The measurement of similarity between two data points is computed using predefined formula and real-valued messages are transmitted among neighboring data points until a set of exemplars and respective clusters are obtained [23].

Mean Shift

A nonparametric clustering technique that does not assume prior knowledge of the number of clusters.This algorithm works by moving data points toward centroids to become the average of nearby points. It identifies dense areas of data points by shifting centroids toward areas of higher density, using a kernel-based approach. Mean-shift does not require specifying the number of clusters in advance. The number of clusters is determined by the algorithm with respect to the data.

The basic idea behind mean-shift clustering is to shift each data point towards the mode of the distribution of points within a certain radius. The algorithm iteratively performs these shifts until the points converge to a local maximum of the density function [24].

Spectral Clustering

It uses the connectivity between the data points to form the clustering. Uses the eigenvalues of a similarity matrix to reduce dimensionality before clustering. It is particularly effective in identifying non-convex clusters and can outperform k-means in complex cluster structures. It is based on the idea of a graph representation of data where the data point are represented as nodes and the similarity between the data points are represented by an edge [24].

Hierarchical Clustering

Builds a hierarchy of clusters either in a bottom-up (agglomerative) or top-down (divisive) manner. The results are typically represented using a dendrogram, allowing flexibility in choosing the number of clusters [24].

Density-Based Spatial Clustering of Applications with Noise

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is effective for detecting arbitrarily shaped clusters and outliers. It groups together points that are closely packed and labels points in low-density regions as noise [24].

Balanced Iterative Reducing and Clustering using Hierarchies

Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) is designed for large datasets. It incrementally and dynamically clusters incoming data using a tree structure to summarize cluster information efficiently [25].

It can cluster large datasets by first generating a small and compact summary of the large dataset that retains as much information as possible. This smaller summary is then clustered instead of clustering the larger dataset. BIRCH is often used to complement other clustering algorithms by creating a summary of the dataset that the other clustering algorithm can now use [24].

2.3.3 Semi-supervised Learning

Semi-supervised learning is a midterm between the supervised and unsupervised learning. It leverages a small amount of labeled data along with a larger set of unlabeled data during training. This approach is particularly advantageous when labeled data is expensive or difficult to obtain, while unlabeled data is abundant [26], [27].

The goal of semi-supervised learning is to learn a function that can accurately predict the output variable based on the input variables, similar to supervised learning. However, unlike supervised learning, the algorithm is trained on a dataset that contains both labeled and unlabeled data [24].

2.3.4 Supervised Learning

Supervised learning is a machine learning approach that is defined by its use of labeled data sets [21]. It is defined by its use of labeled datasets to train algorithms to classify data or predict outcomes accurately. As input data is fed into the model, the model adjusts its weights until it has been fitted appropriately. This will be better discussed below.

Regression

Regression is used when the output variable is continuous, meaning that result will predict a numerical value. It is often used in problems where the goal is to estimate or predict a quantity [28], [29].

Applications of Regression:

- **Price prediction:** Estimating the price of a house based on features like square footage, number of rooms, and location.
- **Stock market forecasting:** Predicting future stock prices based on historical data.
- **Temperature prediction:** Estimating daily temperatures based on factors like time of year and geographic location.

In regression, the model tries to find a function that best describes the relationship between the input features and the output variable.

Classification

Classification is used when the output variable is categorical, meaning that the result will assign data into one of several classes or categories. The goal is to predict the class or category that the input data belongs to [30].

Applications of Classification:

- **Spam detection:** Identifying whether an email is spam or not based on its content.
- **Image recognition:** Classifying images into categories, such as "cat," "dog," or "car."
- **Medical diagnosis:** Classifying patients based on medical test results to determine whether they have a certain disease or condition.

In classification, algorithms learn to distinguish between different classes by analyzing the patterns in the features. The objective here is to classify the data provided into four classes, because it was realized the movement for four fingers. Then it will be studied how to apply the classification method and its accuracy in detecting which finger moved. For example, logistic regression, decision trees, random forest, and support vector machine are popular classification techniques [29]. Even though Random Forest will be the main classification method used in this work, this section considers other relevant algorithms as well.

Logistic Regression

Logistic regression is a widely used algorithm for classification tasks. It is a probability-based classifier derived from linear regression models. While linear regression predicts continuous values using one or more independent variables, logistic regression applies a sigmoid function to generate probabilities and classify data points into discrete categories [28].

Decision Tree

Decision Tree (DT) are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. A tree can be seen as a piecewise constant approximation [25].

DTs are versatile models used for both classification and regression tasks. They recursively split the datasets into progressively smaller groups on the basis of binary classification judgments. The resulting structure resembles a tree, branching from an initial decision node into subsequent leaves or branches. Decision trees are known for their interpretability and ease of implementation.

Random Forest

Random Forest (RF) are ensemble learning methods that improve upon decision trees by combining multiple trees into a single predictive model. By aggregating the outputs of multiple decision trees, random forests improve the precision of prediction while reducing the risk of overfitting [31]. Like decision trees, random forests can be applied to both classification and regression problems. In this case of study, it will be used just as a classification.

Support Vector Machine

Support Vector Machine (SVM) are powerful classifiers that map data points in a multidimensional space, where the number of dimensions corresponds to the number of features in the dataset. The algorithm seeks to find the optimal decision boundary, known as a hyperplane, that best separates the data points into distinct classes [32].

The optimal hyperplane is the one that maximizes the margin, the distance between the hyperplane and the closest data points in each class, known as support vectors. Although SVM models that separate data using a hyperplane are linear, they can also handle nonlinear classification tasks using kernel functions that transform the feature space.

k-Nearest Neighbors

k-Nearest Neighbors (KNN) is a non-parametric classification algorithm that maps data points into a multidimensional space and classifies them based on the majority class among their closest neighbors [33]. To classify a new data sample, the algorithm examines the k nearest data points, determines the class distribution among those neighbors, and assigns the majority class to the new data point.

KNN models belong to the category of lazy learners, meaning that they do not build explicit models during the training phase. Instead, they store training data and perform computations at the time of prediction, which can make them slower than eager learners but more flexible in handling complex distributions.

Naive Bayes

Naïve Bayes classifiers are based on Bayes' theorem and use probability theory to predict class membership. These classifiers update the posterior probability of a class given new evidence, refining their predictions dynamically [34].

For example, in a diabetes prediction model, features such as blood pressure, age, and glucose levels serve as independent variables. A Naïve Bayes classifier combines prior probabilities (the general prevalence of diabetes) with the likelihood of the observed features in diabetic patients to compute a final probability score.

2.3.5 Reinforcement Learning

Reinforcement learning is a machine learning approach, similar to supervised learning. But instead of setting labeled data as target vector, this algorithm will teach using rewards. Unlike supervised learning, which requires labeled input-output pairs, or unsupervised learning, which seeks hidden patterns in unlabeled data. Reinforcement learning is centered around the concept of trial and error guided by feedback from the environment. A sequence of successful outcomes will be reinforced to develop the best recommendation or policy for a given problem [17], [35].

2.4 Performance measures

2.4.1 Silhouette Score

The Silhouette Score is an internal evaluation metric used to assess the quality of clustering in unsupervised learning [36]. It quantifies how well a data point fits within its assigned cluster compared to other clusters. The score ranges from -1 to 1, where values close to 1 indicate that the data points are well clustered, values near 0 suggest overlapping clusters, and negative values imply incorrect clustering. The silhouette coefficient $s(i)$ for each point i is defined as:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

where $a(i)$ is the mean intra-cluster distance (cohesion) and $b(i)$ is the mean nearest-cluster distance (separation). The overall score is the average of $s(i)$ over all points. In this project, it is used to evaluate the separation of movements identified by clustering algorithms applied to FBG signals.

2.4.2 Confusion Matrix

The confusion matrix is a fundamental evaluation tool in supervised classification tasks [37]. It is a square matrix that visualizes the performance of a classification algorithm by comparing the predicted classes against the true labels. Each element $C_{i,j}$ of the matrix represents the number of samples belonging to class i that were predicted as class j. The diagonal elements indicate correct predictions, while the off-diagonal elements represent misclassifications. The matrix is especially useful in multiclass problems like finger movement recognition, as it provides detailed insight into which movements are more likely to be confused.

2.4.3 Precision

Precision measures the proportion of true positive predictions among all positive predictions made by the model [38]. It reflects the ability of the classifier to avoid false positives. For a given class:

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

High precision is crucial in applications where the cost of false positives is high. In this context, precision helps evaluate how reliably the model detects a specific finger movement without wrongly classifying other gestures as that movement.

2.4.4 Recall

Recall, also known as sensitivity or true positive rate, measures the proportion of actual positive cases that were correctly predicted [38]. It indicates the model's ability to capture all relevant instances of a class. For a given class:

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

A high recall means that most instances of a class are correctly detected. This metric is especially important in biomedical applications, where missing a relevant signal (false negative) can be critical.

2.4.5 F1-Score

The F1-Score is the harmonic mean of precision and recall [38]. It provides a balanced measure that considers both false positives and false negatives. It is especially useful when dealing with imbalanced datasets. The F1-Score is defined as:

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

In the proposed methodology, the F1-Score is used to summarize the model's classification performance for each type of finger movement, offering a single metric that balances correctness and completeness.

2.4.6 Leave-Two-Out Cross-Validation

Leave-2-Out Cross Validation (L2OCV) is a less common yet effective variant of cross-validation, in which two subjects are excluded from the training set in each iteration and used exclusively for testing. This approach is particularly suitable for biomedical and user-specific datasets, where the main goal is to evaluate the model's ability to generalize to entirely unseen individuals. In this study, L2OCV was adopted as the primary validation strategy to assess the performance and generalization capability of the Random Forest classifier [39], [40].

2.5 Related Works

This section presents recent research related to the proposed project, which applies technology integrated with machine learning algorithms for signal classification.

2.5.1 Hand Movement Recognition

Barbosa [8] proposed a device control system based on classified EMG signals using supervised learning techniques. Her work involved collecting electromyographic signals during finger movements and applying machine learning classifiers to distinguish between different gestures. The study achieved promising results using algorithms such as k-Nearest Neighbors (k-NN), Decision Trees, and Random Forests. This research demonstrates the feasibility of using biosignals for gesture recognition and device control and serves as a methodological reference for the application of similar techniques to FBG signals.

2.5.2 Movement Recognition Using FBG

Recent developments in wearable sensor technology have led to the creation of innovative devices for hand posture sensing by Rao et al. [41]. One notable advancement is the integration of FBGs into wearable gloves. This study introduced a new wearable fiber optic sensor glove that uses flexible materials (Figure 2.4), such as polydimethylsiloxane (PDMS) and silicone tubes, to encapsulate FBGs.

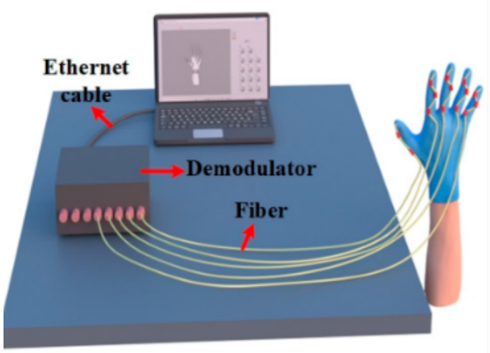

Figure 2.4: Schematic diagram of sensor integration [41]

The design enables the glove to perceive hand posture, recognize gestures, and predict grasping objects. The glove can concurrently monitor the motion of 14 hand joints, including metacarpophalangeal (MCP), proximal interphalangeal (PIP), and distal interphalangeal (DIP) joints. To enhance the measurement range of the sensors, a sinusoidal layout incorporates the FBG array into the glove. The study employed the Support Vector Machine (SVM) approach for predicting grasping objects, demonstrating the potential of combining advanced sensor technology with machine learning techniques for improved hand posture sensing. This research shows the possibility of applying optical sensors data in machine learning and achieving good results.

Chapter 3

Methodology

This chapter presents the methodology adopted during the research, which focuses on the analysis and processing of biosignals for the development of machine learning models. The study is based on a pre-existing dataset collected from forearm muscle activity. The signal presented is complex and not labeled. The methodology emphasizes on machine learning method and organize the signal, characterization, and preparatory steps for classification. Each stage is essential for building an accurate and robust system capable of recognizing the fingers movements, and for understanding the results presented in the next chapter.

3.1 Data Acquisition

The data used along this research was gathered by another team in Brazil, whose collected the signal from each patient's forearm and built an automated prosthesis [9], [12]. But it was faced some problems in classifying these data. Data acquisition started with patients being asked to wear a bracelet equipped with an FBG sensor on their forearm (as already shown in Figure 2.3 and instructed to perform one finger movement at a time, with five repetitions of moving up and down. The movements captured were from index, middle, ring and little finger. The tension made in the forearm muscle will disturb the FBG wavelength, which will be sent as data to a CSV file. All movements realized for each patient were stored as signal in each personal CSV file. The signal characteristics will be described in more detail in Chapter 4. All database received contains a total of ten CSV files, each file corresponding to a signal from a different patient.

3.2 Data Preparation

Since the raw data are unstructured and continuous, this step of the methodology consists in organizing this information into a usable format. Before any processing, the signals must be properly structured to reflect the experimental design. It means that all gaps, doubled lines or errors will be reviewed in order to have a good information. An essential part of the data preparation phase involves understanding the temporal structure of each signal and associating portions of the signal with the corresponding finger movements. This segmentation is initially guided by the known order of execution during the acquisition protocol. At this stage, no labels are assigned to specific time segments. Rather, the data are formatted to allow for further unsupervised learning analysis. Additionally, all signals are visually inspected and truncated if necessary to remove irrelevant sections before or after the active movement intervals. The result of this step is a clean, temporally consistent dataset where each patient's signal spans the same sequence of actions, enabling comparative analysis and downstream processing.

3.3 Clustering Techniques

Since all the data are unlabeled, this will difficult the accuracy of the model. As seen before, the best clustering would be k-Means.

Clustering not only enables visualization of when each movement occurs but also serves as a labeling strategy in the absence of predefined classes, which is common in signal-based biomedical datasets.

3.3.1 Methods Used

In this is step will be used as unsupervised machine learning algorithm:

- **k-Means**
- **Agglomerative Clustering**
- **Spectral Clustering**
- **BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies)**
- **DBSCAN (Density-Based Spatial Clustering of Applications with Noise)**

In order to compare which one will provide the best grouping. The database was tested for all of them, to proof the tool selection.

Elbow Method

The k-means algorithm is used to group similar signal patterns, facilitating the identification of different finger movements within the data.

In order to have an optimal number of clusters for k-Means to classify the dataset, it will be used the Elbow Method. It is a widely used heuristic to identify the appropriate number of clusters for a dataset. It consists in computing the Within-Cluster Sum of Squares (WCSS) for different numbers of clusters and plotting the results in a line graph [42], as follows in the formula Equation 3.1:

$$\text{WCSS} = \sum_{k=1}^{C_n} \left(\sum_{d_i \in C_k}^{d_m} \text{distance}(d_i, C_k)^2 \right) \tag{3.1}$$

Where C_n is the total number of clusters, C_k represents the centroid of the k-cluster, d_i is a data point belonging to cluster C_k, and d_m is the number of point in the cluster in C_k.

The inner summation calculates the squared Euclidean distance between each data point and its respective cluster centroid, while the outer summation aggregates this value across all clusters. The WCSS is a measure of cluster compactness: the lower the WCSS, the more tightly the data points are grouped around their centroids.

The point where the decrease in WCSS slows down significantly, forming an "elbow" shape, is considered the optimal number of clusters. This indicates the point at which adding more clusters provides diminishing returns in terms of explained variance.

This step is especially useful to ensure that the assumption of four clusters (based on the number of intended finger movements) is consistent with the actual structure of the data. While it is possible to define the number of clusters manually, applying the elbow method adds methodological rigor and helps confirm the separability of the movements in the signal space.

Silluete Score

After that, the unsupervised machine learning algorithm will provide the groups of each cluster made. An important metrics to use with these results is Silluete Score metrics. It quantifies how each data point fits within its assigned cluster compared to other clusters. The score combines two factors: cohesion (how close a data point is to other points in the same cluster) and separation (how far it is from points in the nearest different cluster).

In this study, the Silhouette Score is employed to compare different clustering methods and to support the selection of the most suitable algorithm for the structure of the data.

k-Means for Dataset Labeling

Once the signal is segmented into clusters, it becomes essential to understand the characteristics of each group. This step allows for an in-depth analysis of how the signals differ and provides insight into which features may be most relevant for machine learning models. The more distinguishable the clusters are in terms of statistical features, the more accurate the classification models are likely to be.

Since k-Means groups randomly, it is necessary to prepare the new labeled data again. This means that when data are clustered, the first cluster of one patient will not represent the same to other patients. But in fact, the procedure was the same for all of them. So, each cluster was reordered according to temporal data from cluster centroids.

3.4 Classification technique

After that all the data are labeled, the supervised learning part can be started for classifying. Random Forest will be used as the classifier, due to its precision mentioned in a previous research [7].

Considering [9], Random Forest provides more accurate results for this application compared to other supervised learning methods, such as SVM, KNN, and Decision Trees. Therefore, it will be used to classify the labeled dataset generated by the clustering step.

It will be selected the most useful features extracted for random forest and applied in the process.

3.5 Data Validation

Cross-validation is a well-established strategy for estimating the generalization performance of machine learning models, especially when the dataset is limited in size. In this project, considering the reduced number of subjects, the method adopted wasL2OCV, which offers a balanced evaluation across different subject combinations.

3.5.1 Leave Two Out Cross Validation

L2OCV consists in training the model using data from $N - 2$ patients and testing it on the remaining 2, iteratively cycling through all possible unique pairs of patients. The total number of iterations is given by:

$$\text{Iterations}_{L2O} = \binom{N}{2} = \frac{N(N-1)}{2} \tag{3.2}$$

In this study, with $N = 10$ subjects, the total number of iterations is:

$$\text{Iterations}_{L2O} = \binom{10}{2} = 45$$

Each iteration respects the approximate split between 80% training and 20% testing data. Specifically, in every fold, the model is trained with 8 patients (80%) and evaluated with 2 patients (20%). This setting is particularly relevant in biomedical signal processing, where inter-subject variability is significant, and testing on unseen subjects is essential to validate generalization [39].

The use of leave-pair-out validation is supported in the literature as a reliable alternative to traditional k-fold and leave-one-out methods. According [39], leave-p-out cross-validation methods tend to offer better performance estimates in scenarios with small sample sizes and high variability . Blanchard and Massart [43] further demonstrated that such approaches can adapt better to margin conditions in classification problems.

3.5.2 Confusion Matrix

To evaluate the classification performance of the model at each iteration of L2OCV, a confusion matrix is computed. The confusion matrix is a fundamental tool in supervised learning for analyzing the quality of predictions in multi-class problems.

It is structured as an $n \times n$ matrix, where n is the number of movement classes considered. The rows correspond to the true class labels, and the columns represent the predicted class labels. Each element C_{ij} in the matrix indicates the number of times a sample from class i was predicted as class j.

From the confusion matrix, several performance metrics are derived:

- **Precision:** the ratio between true positives and total predicted positives for a given class.
- **Recall (Sensitivity):** the ratio between true positives and total actual instances of the class.
- **F1-score:** the harmonic mean of precision and recall, offering a balance between them.
- **Macro average:** Calculates the average of the metric (Precision, Recall and F1-score) for each class equally, regardless of the number of samples in each class.

Useful to treat all classes with equal importance.

- **Weighted average:** It is calculated the average of the metric weighted by the number of samples in each class. Useful when the account for class imbalance.

These metrics are calculated for each fold and then averaged across all 45 iterations to obtain a final performance estimation for each test pairs. This evaluation enables a detailed understanding of the model's strengths and weaknesses in classifying specific finger movements from FBG signal data. In this model will be classified four clusters. It will be used L2OCV, meaning that two patients will be used as a test, resulting in eight classifications in each interaction. So for this model will be presented a total of 360 classifications.

3.6 Python

Python is a high-level and easy-to-learn programming language that offers a wide range of libraries, making it easier to access various functionalities and implement complex tasks efficiently.

For the use of machine learning, Scikit-Learn was selected [25]. It provides support for each clustering method in unsupervised learning. For supervised learning it offers all methods, but it was chosen just random forest to classify the clusters.

To analyze the signal features used, Pandas library was employed [44]. This library is widely used for data manipulation and statistical analysis. Through the use of the `.describe()` function, a variety of statistical descriptors are extracted for each cluster.

These statistics are crucial for evaluating signal stability, variability, and distribution patterns. For instance, high variability may indicate noisy segments, while tightly grouped values may suggest consistent muscle activation. Such information helps in selecting features for classification and in designing filters or normalization procedures.

Chapter 4

Case Study

This chapter presents a detailed characterization of the dataset used in this study, which was provided by a research group in Brazil [7]. The dataset comprises forearm biosignals acquired using FBG sensors during repeated finger movements. Analyzing and understanding these signals is a critical step toward building a robust unsupervised clustering and supervised classification framework.

Since the dataset lacks labeled events or movement annotations, it poses significant challenges in interpretation and model training. Therefore, this chapter focuses on describing the structure of the dataset, the characteristics of the signals, and the practical issues that must be addressed before applying machine learning techniques.

4.1 Dataset Description

The dataset consists of ten independent recordings one for each subject captured during controlled hand movement tasks. Each file is stored in '.txt' format and contains six columns of time-series and diagnostic information, as follows:

- `Time 1` – Sampling time in seconds;
- `NTP_TIME` – Absolute time in seconds using the Network Time Protocol;
- `DIAG_SEQ_NUMBER` – Diagnostic sequence identifier;

- `DIAG_BUF_FREE` – Buffer availability in percentage;
- `DIAG_SAMPLES_PER_CYCLE` – Number of samples per acquisition cycle;
- `CH_1 Sensor_1` – Sensor reading (in nanometers) from the FBG channel.

An excerpt of one signal file is shown in Table 4.1.

Time [s]	**NTP TIME [s]**	**DIAG SEQ NUMBER**	**DIAG BUF FREE [%]**	**DIAG SAMPLES PER CYCLE**	**CH1 Sensor_1 [nm]**
0.000	1655222038.225	24904000.000	99	1	1543.466
0.020	1655222038.245	24904020.000	100	1	1543.466
0.040	1655222038.265	24904040.000	100	1	1543.466
0.060	1655222038.285	24904060.000	100	1	1543.465
0.080	1655222038.305	24904160.000	100	1	1543.466
0.180	1655222038.405	24904080.000	99	1	1543.465

Table 4.1: Excerpt of FBG signal log with diagnostic fields.

Each recording contains approximately 9500 samples. During acquisition, each subject performed four distinct finger movements: index, middle, ring, and little fingers. Thumb movement was not included in the protocol. These gestures were performed in sequence, but were not annotated with time markers in the signal (Figure 4.1), resulting in an unlabeled dataset.

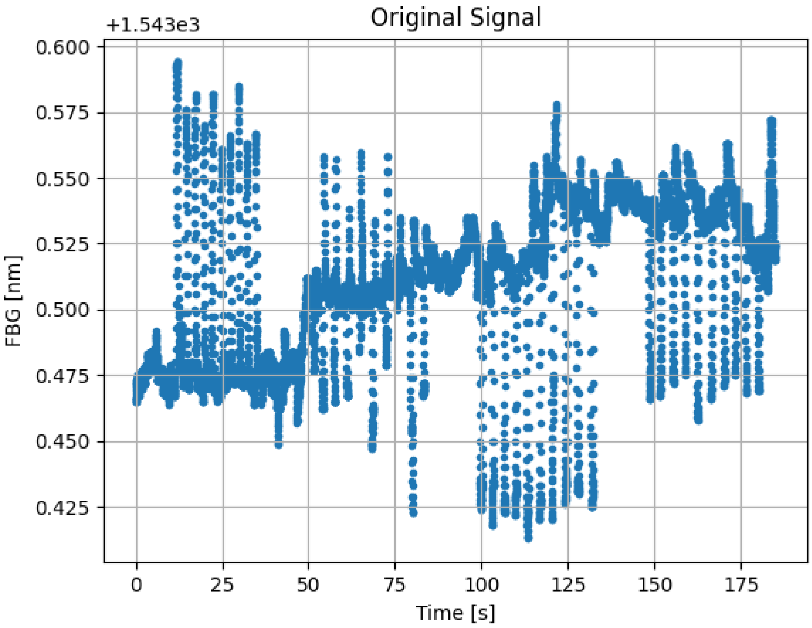

Figure 4.1: Original Signal

4.2 Data Interpretation Challenges

Although the signal data is continuous and appears stable, the lack of movement labels introduces a significant interpretation challenge. Without markers to indicate the start and end of each gesture, traditional supervised learning approaches cannot be applied directly. Furthermore, subtle variations in signal amplitude and pattern may differ across subjects due to anatomical and behavioral differences, requiring adaptive or individualized processing.

Noise is minimal due to the quality of the FBG sensors, but the narrow range of signal variation reduces the contrast between movements. This makes clustering and classification more sensitive to minor fluctuations, highlighting the need for accurate preprocessing and feature selection.

4.3 Signal Characteristics

To investigate the structure of the signal, a descriptive statistical analysis was performed using the `pandas` Python library [44]. The `.describe()` method was used to extract the following statistical indicators:

- **Count:** Total number of signal samples;
- **Mean:** Average amplitude value;
- **Standard deviation:** Amplitude variability;
- **Min/Max:** Range of signal values;
- **25%, 50%, 75% quantiles:** Distribution shape indicators.

A representative summary of one signal recording is shown in Table 4.2.

Statistic	**Time [s]**	**FBG Signal [nm]**
Count	9502	9502
Mean	95.010	1543.722
Std	54.863	0.080
Min	0.000	1543.435
25%	47.505	1543.663
50% (Median)	95.010	1543.747
75%	142.515	1543.785
Max	190.020	1543.904

Table 4.2: Descriptive statistics of a sample signal.

These values indicate that sensor readings are consistent and exhibit low variance, which is favorable for noise reduction. However, the subtlety in variation between samples reinforces the importance of feature extraction techniques capable of detecting small but relevant differences associated with each movement type.

4.4 Overview

In summary, this chapter provided an overview of the dataset structure, explored the statistical properties of the FBG signal, and discussed the main challenges of working with unlabeled biosignals. These insights justify the use of unsupervised clustering to segment the signal and highlight the need for robust classification strategies.

The next chapter presents the application of these techniques and evaluates their performance in distinguishing finger movements from the raw forearm signals.

Chapter 5

Results and Discussion

This chapter presents and discusses the results obtained throughout the research. It will disposed tables and graphics to facilitate the understanding of the development. This section will follow the sequence of events set during Methodology (Chapter 3).

5.1 Clustering

After the data preparation phase (Chapter 3.2) and the analysis of the raw signal (Chapter 4), the next step was to divide the signal into parts referring to the number of fingers' movements captured. Since the raw signal was not labeled, it would not be possible to determine which movement was being performed. The focus will be to group the signal's data in four groups, representing each finger. It was used Python's library for that objective, obtaining the optimal number of clusters and clustering raw signal into labeled data. k-Means was chosen to cluster the data from each patient. Considering that the process was the same for each patient and they were oriented moving five times the finger using the same order, it can be divided into:

- Cluster 0 represents the index movements;
- Cluster 1 represents the middle finger movements;
- Cluster 2 represents the ring finger movements;

- Cluster 3 represents the little finger movements;

The optimal number of clusters for the use of k-Means method is calculated by the Elbow Method (Figure 5.1). The goal is to identify the point where the rate of decrease in WCSS (Equation 3.1) sharply changes, indicating that adding more clusters (beyond this point) produces diminishing returns [42], [45]. If it matches to the moves captured, then the software will be able to separate each move in all file provided, Figure 5.1:

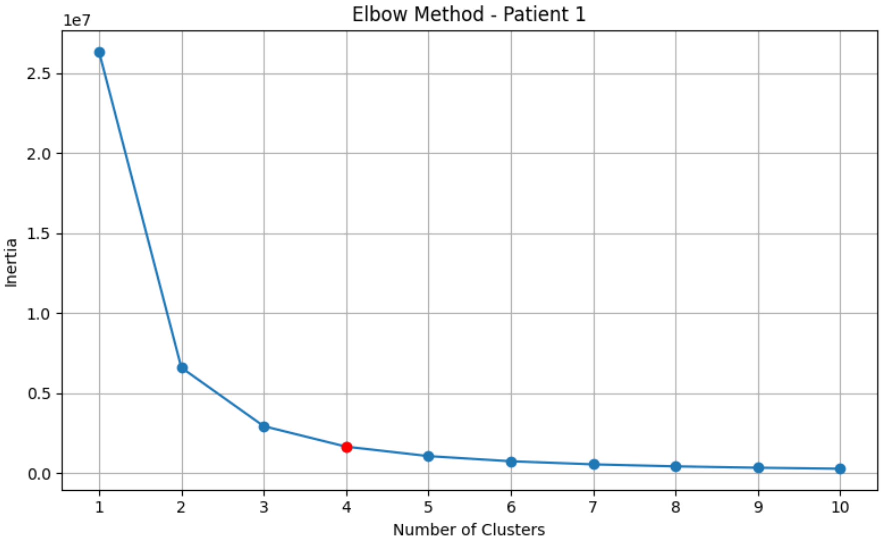

Figure 5.1: Elbow method.

As can be seen in the image above (Figure 5.1), the approximate optimal number of clusters (between 3 and 4 clusters), after the red point the graphic starts to stay steady. It matches the number of movements realized during the test (4 movements). The same repeated in each patient elbow method verification.

This means that k-Means will be able to divide the data as better as possible into four groups. All clustered data are shown bellow, Figure 5.2:

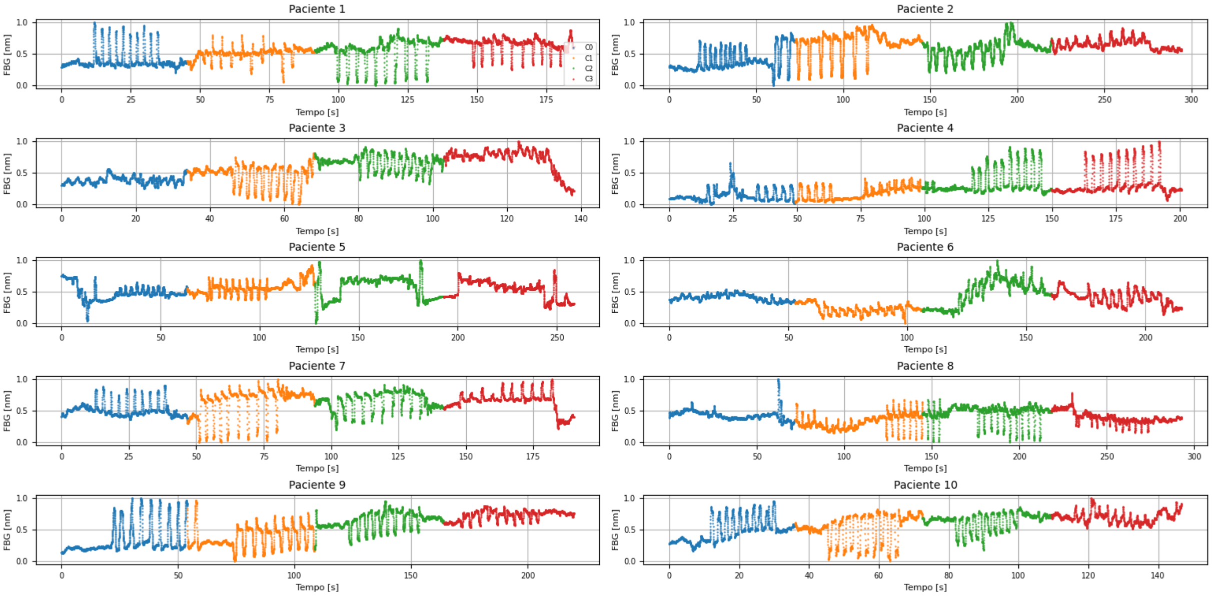

Figure 5.2: Clustered Data.

k-Means method worked with good precision, but not in all cases. As can be seen above in Figure 5.2, some patients such as 5 and 9 had part of the cluster covering data that should be considered on the previous one. That could be one of the disadvantages of this method.

It was tested the signal normalized (by Standard Scaler) using the same process. And k-Means ends messing up in clustering (Figure 5.4).

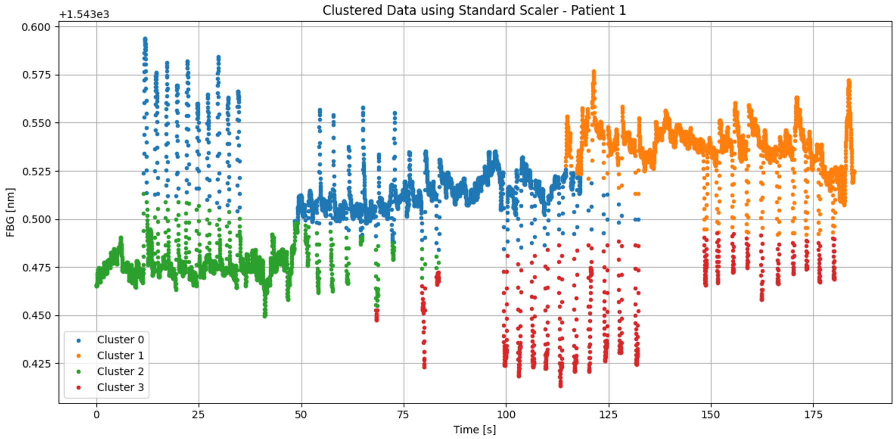

Figure 5.3: Clustered Data using Standard Scaler.

It happens because the tool uses Euclidean distance, using normalized data (Standard

Scaler) in a very low number difference affects the tool process. But applying it on Min-Max Scaler the graphics stayed the same:

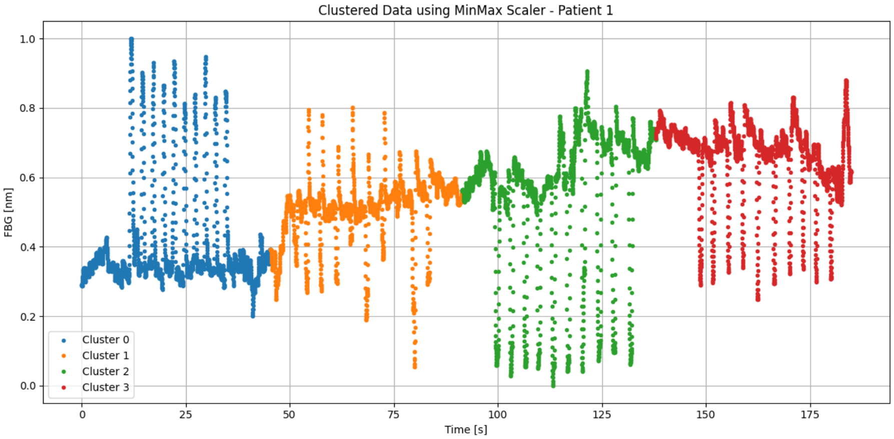

Figure 5.4: Clustered Data using MinMax Scaler.

There was no such difference compared to Clustered Data using the raw signal (Figure 5.2, so it was not used in the main program.

All clusters were reordered according to temporal position,where cluster 0 will always be the first movement and so on. Once k-Means is a centroid-based algorithm, that searches for groups choosing centroids randomly. That is why reordering is crucial in this phase, for example, cluster 0 from patient 1 might not represent the same finger for others patients' cluster 0.

Next, all files were labeled with each cluster representing one finger. Just like the Table 5.1:

Table 5.1: Labeled Data - Patient 1

Time [s]	FBG [nm]	Cluster
0.00	0.29	0
0.02	0.29	0
0.04	0.29	0
0.06	0.29	0
0.08	0.29	0
...	...	0
45.40	0.39	1
45.42	0.39	1
45.44	0.39	1
45.46	0.39	1
45.48	0.39	1
...	...	1
91.38	0.56	2
91.40	0.56	2
91.42	0.56	2
91.44	0.56	2
91.46	0.56	2
...	...	2
138.00	0.71	3
138.02	0.72	3
138.04	0.72	3
138.06	0.73	3
138.08	0.73	3
...	...	3

5.2 Comparison of Clustering Algorithms

To validate the choice of clustering algorithm used in the segmentation step, five widely used methods were compared:

- **k-Means**
- **Agglomerative Clustering**
- **Spectral Clustering**
- **BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies)**

- **DBSCAN (Density-Based Spatial Clustering of Applications with Noise)**

Each algorithm was applied independently to the raw FBG signal of each patient and the clustering quality was evaluated using the Silhouette Score . The number of clusters was set to 4, corresponding to the number of finger movements in the acquisition protocol. Table 5.2 shows the s Silhouette Scores obtained per patient.

Patient	**k-Means**	**Agglom.**	**Spectral**	**BIRCH**	**DBSCAN**
1	0.572	0.561	0.560	0.553	-0.05
2	0.572	0.540	0.560	0.554	0.607
3	0.572	0.533	0.560	0.564	0.513
4	0.572	0.508	0.560	0.511	0.502
5	0.572	0.537	0.560	0.544	0.182
6	0.572	0.527	0.560	0.564	0.457
7	0.572	0.550	0.560	0.549	0.359
8	0.572	0.514	0.560	0.537	0.086
9	0.572	0.558	0.559	0.557	0.499
10	0.572	0.559	0.560	0.558	0.242

Table 5.2: Silhouette scores per patient for each clustering algorithm. DBSCAN values include number of clusters found in parentheses.

Across all patients, k-Means consistently achieved the highest silhouette score (0.572). Spectral clustering yielded similar results but did not surpass k-Means in any case. Agglomerative and BIRCH also performed reasonably well, though slightly below. The difference were very tight as can be seen in Figure 5.5, the first patient was chosen for comparison.

In contrast, DBSCAN produced inconsistent results, with highly variable silhouette scores and widely varying cluster counts, often far from the expected number of 4. This is expected, as DBSCAN is sensitive to its parameters and may interpret minor signal fluctuations as new clusters.

These findings support the use of k-Means as the primary clustering method in this study. Its stability, simplicity, and alignment with the expected number of finger movements make it an effective tool for segmenting unlabeled signal data.

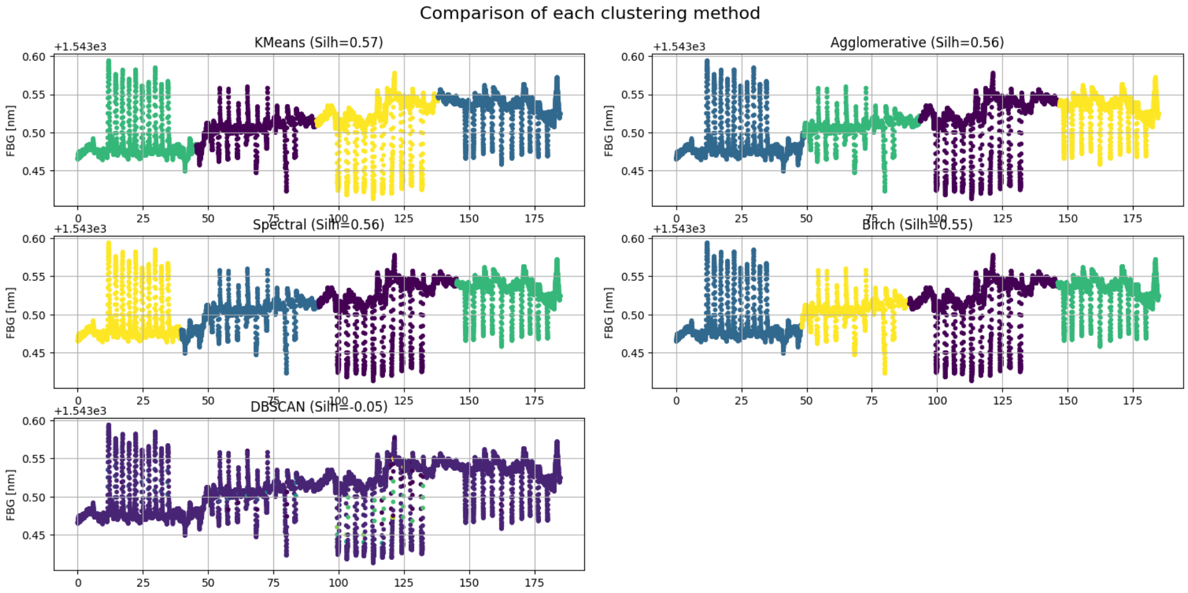

Figure 5.5: Comparison of each cluster method

5.3 Random Forest Classification

The Random Forest algorithm was selected for its robustness, interpretability, and strong performance with structured, tabular data.

The classifier was trained using statistical features [25] extracted from each segment. Among these, the most informative features were:

- **Min,Mean,Max and dispersion percentiles** — already discussed in Chapter 4;
- **Entropy and kurtosis** — capturing the complexity and peakedness of the signal, respectively;
- **Skewness** — indicates de asymetry of the signal distribution;
- **Dominant frequency** — the most representative frequency component within the signal component.

For each patient, there are 4 clusters and their features, in a total of 40 dataset. Resulting in a brand new dataset, Table 5.3 (the table below is part of all dataset, the rest of data is presented on Appendix A.1:

Table 5.3: Dataset of features extracted from patient 1 for random forest

Patient	**Cluster**	**mean**	**std**	**min**	**max**	**skewness**	**kurtosis**
1	0	0.3986	0.1517	0.2008	1.0000	2.3699	4.5380
1	1	0.5043	0.0960	0.0535	0.8012	-1.1628	2.5645
1	2	0.5555	0.2030	0.0000	0.9062	-1.4220	0.9582
1	3	0.6593	0.1124	0.2486	0.8797	-1.6748	2.7199

Training and testing were performed using a leave-2-patients-out cross-validation strategy.

5.3.1 Accuracy per Patient Pair

It was made a table with each pair test in L2OCV , Apenddix B.1. The Figure 5.6 shows the classification accuracy for each test combination.

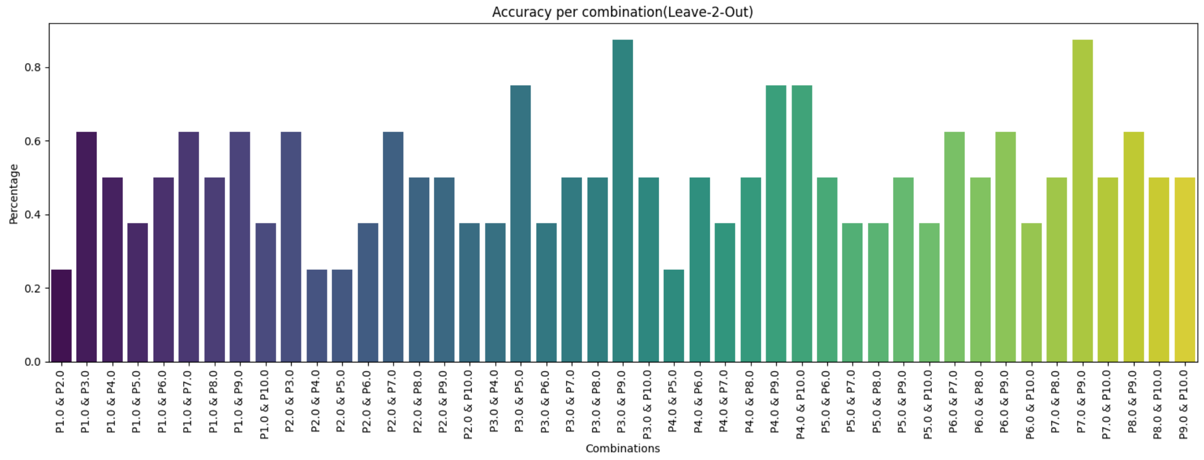

Figure 5.6: Histogram of each accuracy obtained.

These variations reflect differences in signal quality and cluster separability among subjects. Some combinations (e.g., the pair P3 & P9 got approximately 88%) resulted in high classification performance, while others (e.g., P1 & P2 got just 25%) performed poorly due to overlapping clusters or weak transitions. There was a good part rated between 50% and 60% .

5.3.2 Confusion Matrix

The general confusion matrix in all test cases is presented in Figure 5.7. It is presented all classification per pair of patients tested that the model had during L2OCV and added in this matrix. Since there were 45 iterations (as mentioned before in Chapter 3), this chart will show all 360 classifications. This means that all classifications made were divided into what it was supposed to be (True Label) and how the model classified it (Predicted Label), according to the four clusters. As can be seen in Figure 5.7:

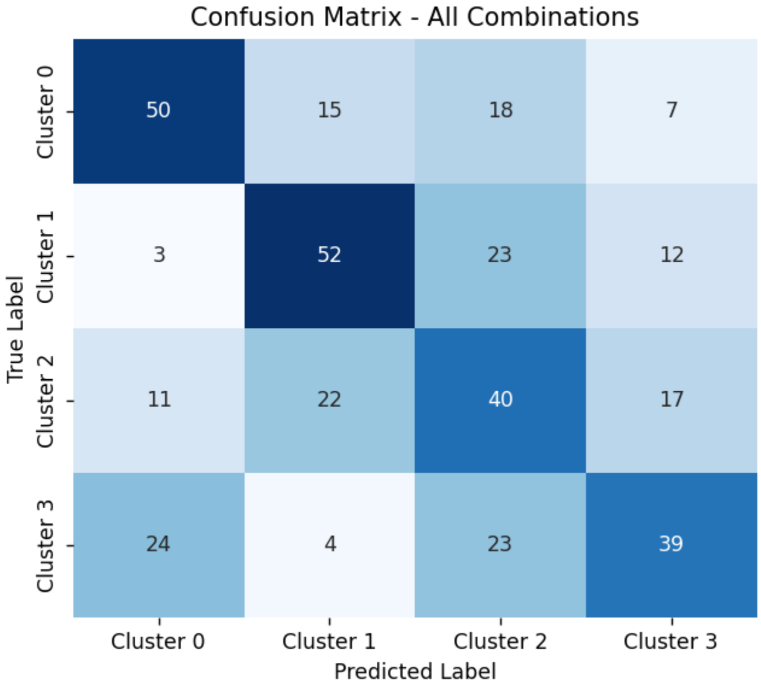

Figure 5.7: General confusion matrix.

It reveals that cluster 1 (typically the middle portion of the signal sequence) was the most accurately predicted, while cluster 3 (final segment) suffered from higher misclassification rates.

The confusion between clusters 2 and 3 is particularly evident, due to the similarity in their signal shapes or positions within the sequence. These two ha less amount of true label per predicted label. This may also result from inconsistencies in how movements were performed or how clusters were aligned between patients.

A Classification Report was made such as Table 5.4. It summarizes the precision, recall, and F1-score for each class across the entire dataset.

Class	**Precision**	**Recall**	**F1-score**
0	0.50	0.50	0.50
1	0.60	0.52	0.56
2	0.83	0.84	0.83
3	0.32	0.35	0.33
Macro avg	0.56	0.55	0.56
Weighted avg	0.56	0.55	0.56

Table 5.4: Classification report aggregated over all test folds.

Despite the modest overall accuracy, this study demonstrates that a hybrid clustering-classification pipeline can extract useful structure from fully unlabeled biosignals. The use of time as a primary feature highlights the regularity of the gesture sequence in the dataset. However, this also raises concerns about overreliance on time-based separation, which may not generalize well to asynchronous or real-world scenarios.

Misclassifications may arise from two main sources:

- Imperfect clustering boundaries, leading to ambiguous training labels;
- Inter-patient variability in signal patterns and gesture timing.

These results validate the pipeline's potential while also exposing its limitations. Future work may incorporate more dynamic segmentation, sensor fusion, or data augmentation to improve generalization.

These results confirm that the time feature plays a central role in segmenting and identifying movement phases in the dataset. Its removal significantly impaired the classifier's ability to generalize across subjects. Although this indicates a reliance on signal position within the sequence, it also highlights the challenge of real-time gesture recognition when time structure is not explicitly defined.

Future research should explore time-invariant features or adaptive sequence detection methods to overcome this limitation.

Chapter 6

Conclusion and Future Work

This work explored a hybrid methodology that combines unsupervised clustering with supervised classification for the analysis of forearm biosignals collected through FBG sensors. The goal was to identify patterns related to finger movements without relying on labeled datasets based on time feature.

The main contributions of this research are:

- A preprocessing pipeline capable of filtering and organizing continuous FBG signals from multiple patients;
- A comparative evaluation of clustering methods, showing that k-Means offers the best balance between stability and alignment with expected movement segmentation;
- The application of statistical descriptors for segment characterization;
- A leave-2-patients-out validation (L2OCV) framework to assess model generalization across different subjects;

The results showed that while classification performance was modest (55% average accuracy with Random Forest). But considering the highest accuracy obtained (88%), it can be affirmed that the model is not overfitting, as expected[7]. Since it was acquired the same number as labeled data. The system was able to extract meaningful structure from unlabeled data. Removing the time feature significantly degraded performance, confirming its importance in organizing movement sequences. Comparing to forearm muscle motion, it is comprehensive that some clusters were too close, affecting later in random forest decision. Additionally, the comparative study of clustering algorithms highlighted the reliability of k-Means when using time-aligned repetitions of gestures.

Several limitations were encountered. The original dataset lacked movement labels, requiring indirect validation strategies. Most models relies on time-structured gestures, reducing the applicability to real-time or asynchronous use. Only simple features (mean, std, skewness, kurtosis, etc.) were used, limiting model expressiveness. There was also variability between patient, which affected generalization. The mos recommend suggestion for avoiding these errors is make a new dataset with longer time between each finger, in a longer period of test. Gather more data than ten patients.

According to the problems faced during this research and some perspectives realizes while analysing the results that it led to, it is suggested as future works and points that should be considered in next steps of this project. First, amplifying the database would be the most advantage in order to advance the theme. Since the major problem arisen during the survey was the doubt if the 10 files would be enough, including standardize the capturing method with more time between each finger. It would help the clustering and return a better precision on random forest for non labeled data. Future research directions include:

- Incorporating additional sensor modalities or using more complex features (e.g., frequency domain descriptors);
- Developing real-time gesture segmentation algorithms to replace time-based assumptions;
- Applying dynamic time warping or deep unsupervised methods (autoencoders, contrastive learning) for better representation;
- Creating annotated datasets to support more supervised benchmarking;

- Extending the method to embedded applications for real-time prosthetic control.

Despite its limitations, this work provides a foundation for future exploration of unlabeled biosignal classification and highlights the challenges of building intelligent prosthetic control systems from real-world data.

Bibliography

[1] V. Mayer-Schönberger and K. Cukier, *Big Data: A Revolution That Will Transform How We Live, Work, and Think*. Houghton Mifflin Harcourt, 2013.

[2] A. Badman and M. Kosinski, *What is big data?* `https://www.ibm.com/think/topics/big-data`, Nov. 2024.

[3] A. Oussous, F.-Z. Benjelloun, A. A. Lahcen, and S. Belfkih, "Big data technologies: A survey," *Journal of King Saud University - Computer and Information Sciences*, vol. 30, 2018. DOI: `10.1016/j.jksuci.2017.06.001`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S1319157817300034`.

[4] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, 2015.

[5] V. Gupta, V. K. Mishra, P. Singhal, and A. Kumar, "An overview of supervised machine learning algorithm," in *11th International Conference on System Modeling Advancement in Research Trends (SMART)*, 2022, pp. 87–92. DOI: `10.1109/SMART55829.2022.10047618`.

[6] M. A. Oskoei and H. Hu, "Support vector machine-based classification scheme for myoelectric control applied to upper limb," *IEEE Transactions on Biomedical Engineering*, vol. 55, no. 8, pp. 1956–1965, 2008.

[7] P. V. R. Júnior, "Pattern recognition in hand finger extension movements using data collected by bragg gratings in optical fibers," M.S. thesis, UTFPR, 2024.

[8] A. C. Barbosa, "Device control system based on classified emg signals: A machine learning approach," M.S. thesis, IPB, 2024.

[9] P. V. R. Júnior, E. H. Dureck, A. Kalinowski, D. G. Fernandes, J. C. C. d. Silva, and U. J. Dreyer, "Automated prosthesis control by using fiber bragg grating forearm sensor," in *2021 SBMO/IEEE MTT-S International Microwave and Optoelectronics Conference (IMOC)*, 2021, pp. 1–3. DOI: `10.1109/IMOC53012.2021.9624845`.

[10] Y. Guo, X. Tong, Y. Shen, and H. Wu, "Wearable optical fiber beat frequency digital sensing system for real-time non-invasive multiple human physiological parameters monitoring," *Journal of Lightwave Technology*, vol. 41, no. 9, pp. 2911–2920, 2023. DOI: `10.1109/JLT.2023.3238476`.

[11] K. S. C. Kuang, R. Kenny, M. P. Whelan, W. J. Cantwell, and P. R. Chalker, "Optical fiber bragg grating sensor for dynamic strain measurement," *Measurement Science and Technology*, vol. 13, no. 10, pp. 1523–1529, 2002.

[12] A. Kalinowski, E. H. Dureck, U. J. Dreyer, *et al.*, "Recognition of fingers movement using fiber bragg gratings in silicon elastomer packing," in *SBMO/IEEE MTT-S International Microwave and Optoelectronics Conference (IMOC)*, 2019. DOI: `10.1109/IMOC43827.2019.9317561`.

[13] TeachMeAnatomy. "Muscles of the posterior forearm." (2024), [Online]. Available: `https://teachmeanatomy.info/upper-limb/muscles/posterior-forearm/`.

[14] Kenhub. "Flexores superficiais do antebraço." (2024), [Online]. Available: `https://www.kenhub.com/pt/library/anatomia/flexores-superficiais-do-antebraco`.

[15] HBM Test and Measurement, *Dicas e informações: O que é sensor fbg?* 2024. [Online]. Available: `https://www.hbm.com/pt/4596/dicas-e-informacoes-o-que-e-sensor-fbg/`.

[16] FBGS Technologies GmbH, *An introduction to the fbg principle*, `https://fbgs.com/technology/fbg-principle/`, 2024.

[17] IBM Data and AI Team, *What is machine learning (ml)?* `https://www.ibm.com/think/topics/machine-learning`, Sep. 2021.

[18] Y. Weng, J. Wu, T. Kelly, and W. Johnson, "Comprehensive overview of artificial intelligence applications in modern industries," *arXiv*, 2024.

[19] Microsoft Azure, *What are machine learning algorithms?* `https://azure.microsoft.com/en-us/resources/cloud-computing-dictionary/what-are-machine-learning-algorithms/`, 2024.

[20] UC Berkeley, *What is machine learning (ml)?* UC Berkeley Online Blog, `https://ischoolonline.berkeley.edu/blog/what-is-machine-learning/`, 2020.

[21] IBM Cloud Education. "Supervised vs. unsupervised learning: What's the difference?" (2023), [Online]. Available: `https://www.ibm.com/think/topics/supervised-vs-unsupervised-learning`.

[22] Google Cloud, *What is unsupervised learning?* 2024. [Online]. Available: `https://cloud.google.com/discover/what-is-unsupervised-learning`.

[23] M. Mallik, A. K. Panja, and C. Chowdhury, "Paving the way with machine learning for seamless indoor–outdoor positioning: A survey," *Information Fusion*, 2023. DOI: `https://doi.org/10.1016/j.inffus.2023.01.023`. [Online]. Available: `https://www.sciencedirect.com/science/article/pii/S1566253523000313`.

[24] GeeksforGeeks, *Unsupervised learning*, `https://www.geeksforgeeks.org/machine-learning/unsupervised-learning/`, 2025.

[25] F. Pedregosa, G. Varoquaux, A. Gramfort, *et al.*, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[26] X. Zhu, "Semi-supervised learning literature survey," 2005.

[27] J. E. van Engelen and H. H. Hoos, "A survey on semi-supervised learning," *Machine Learning*, vol. 109, no. 2, pp. 373–440, 2020. DOI: `10.1007/s10994-019-05855-6`.

[28] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006, ISBN: 9780387310732.

[29] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*, 2nd. O'Reilly Media, 2019, ISBN: 9781492032649.

[30] K. P. Murphy, *Machine Learning: A Probabilistic Perspective.* MIT Press, 2012, ISBN: 9780262018029.

[31] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001. DOI: `10.1023/A:1010933404324`.

[32] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995. DOI: `10.1007/BF00994018`.

[33] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *The American Statistician*, vol. 46, no. 3, pp. 175–185, 1992. DOI: `10.1080/00031305.1992.10475879`.

[34] I. Rish, "An empirical study of the naive bayes classifier," IBM Research Division, Yorktown Heights, NY, Tech. Rep. RC 22230 (W0110-022), 2001.

[35] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction.* Cambridge, MA, USA: A Bradford Book, The MIT Press, 2018, ISBN: 0262039249. [Online]. Available: `https://web.stanford.edu/class/psych209/Readings/SuttonBartoIPRLBook2ndEd.pdf`.

[36] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, 1987. DOI: `10.1016/0377-0427(87)90125-7`.

[37] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing & Management*, vol. 45, no. 4, pp. 427–437, 2009. DOI: `10.1016/j.ipm.2009.03.002`.

[38] D. M. W. Powers, "Evaluation: From precision, recall and f-measure to roc, informedness, markedness & correlation," *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37–63, 2011.

[39] S. Arlot and A. Celisse, "A survey of cross-validation procedures for model selection," *Statistics surveys*, vol. 4, pp. 40–79, 2010.

[40] G. Varoquaux, "Cross-validation failure: Small sample sizes lead to large error bars," *NeuroImage*, vol. 180, pp. 68–77, 2018. DOI: `10.1016/j.neuroimage.2017.06.061`.

[41] H. Rao, B. Luo, D. Wu, *et al.*, "Study on the design and performance of a glove based on the fbg array for hand posture sensing," *Sensors (Basel)*, vol. 23, no. 20, p. 8495, 2023. DOI: `10.3390/s23208495`.

[42] V. Salunkhe, *K-means clustering*, `https://medium.com/@viveksalunkhe80/k-means-clustering-b8e4ca9a75bb`, 2020.

[43] G. Blanchard and P. Massart, "Concentration inequalities and model selection," in *Lecture Notes in Mathematics*, vol. 1908, Springer, 2006, pp. 1–145. DOI: `10.1007/11752787_1`.

[44] Pandas Development Team, *Pandas - python data analysis library*, `https://pandas.pydata.org`, 2025.

[45] GeeksforGeeks, *Elbow method for optimal value of k in kmeans*, 2021. [Online]. Available: `https://www.geeksforgeeks.org/machine-learning/elbow-method-for-optimal-value-of-k-in-kmeans/`.

Appendix A

Dataset for Random Forest

Table A.1: Features extracted by cluster and patient for Random Forest

Patient	Cluster	mean	std	min	max	skewness	kurtosis	entropy	peak_count	energy	mean_derivative	std_derivative
1	0	0.398560	0.151744	0.200832	1.000000	2.369858	4.538041	1.201491	87	412.858644	0.002181	0.695997
1	1	0.504344	0.095986	0.053473	0.801240	-1.162824	2.564515	1.480741	93	605.962592	0.003631	0.558744
1	2	0.555489	0.203015	0.000000	0.906170	-1.421985	0.958161	1.755489	95	815.344685	0.003239	0.795943
1	3	0.659272	0.112425	0.248627	0.879664	-1.674797	2.719933	1.715686	81	1049.763280	-0.001928	0.516891
2	0	0.393176	0.169769	0.000000	0.836035	0.937623	-0.034002	1.843374	149	662.838198	0.005264	0.362735
2	1	0.679391	0.191383	0.094395	0.966273	-1.659923	2.287747	1.754492	158	1823.910774	-0.000218	0.374437
2	2	0.569591	0.151523	0.205617	1.000000	-0.089934	0.555010	1.904115	150	1289.521978	0.000908	0.258607
2	3	0.651584	0.077362	0.506110	0.908406	0.646108	0.195501	2.017260	183	1608.954308	-0.002472	0.157505
3	0	0.383899	0.059562	0.249776	0.562606	0.623511	0.375978	2.017586	71	255.518324	0.007322	0.263485
3	1	0.468189	0.170993	0.000000	0.810684	-1.116444	0.273983	1.829688	64	426.321575	0.005893	0.721142
3	2	0.663492	0.117850	0.319088	0.909676	-0.545501	-0.265216	2.027765	69	789.698627	-0.000317	0.616829
3	3	0.715708	0.186213	0.147310	1.000000	-1.583182	1.466657	1.701850	70	958.192345	-0.014776	0.337266
4	0	0.138854	0.092014	0.000000	0.658747	1.760426	4.348258	1.494344	105	68.368694	-0.001300	0.326283
4	1	0.173725	0.095309	0.024484	0.428194	0.307104	-1.197550	2.031739	109	97.964402	0.004571	0.385822
4	2	0.333265	0.183883	0.146060	0.912118	1.692310	1.489135	1.490970	105	366.398216	-0.000137	0.615618
4	3	0.323552	0.196252	0.061748	1.000000	1.888681	2.238698	1.457995	106	364.876477	-0.000054	0.743835
5	0	0.483928	0.119898	0.036049	0.769418	0.378306	0.834396	1.703194	131	788.935592	-0.002960	0.237668
5	1	0.566096	0.103706	0.360882	0.921547	0.732531	0.911329	1.893989	115	1064.871760	0.001265	0.277239
5	2	0.568690	0.179427	0.000000	1.000000	-0.478371	-0.580100	1.682783	138	1159.262498	-0.003335	0.331014
5	3	0.530254	0.135473	0.222320	0.843929	-0.430065	-0.816354	2.004931	145	983.032287	-0.001600	0.234125
6	0	0.405957	0.051193	0.299292	0.536126	-0.041315	-0.842668	2.137117	100	442.663235	-0.000672	0.132910
6	1	0.229260	0.073825	0.000000	0.388171	0.066285	-0.407614	1.982214	105	155.293095	-0.002245	0.225875
6	2	0.479448	0.206758	0.103510	1.000000	-0.127321	-1.204750	2.036367	116	739.616120	0.004516	0.407858
6	3	0.400757	0.113240	0.135357	0.674604	0.052651	-0.597758	2.111099	108	473.810117	-0.004318	0.267689
7	0	0.506760	0.108196	0.351074	0.898891	1.723882	2.658349	1.747152	109	626.438803	-0.001010	0.506144
7	1	0.645899	0.221552	0.000000	1.000000	-1.562288	1.323681	1.634713	121	1101.330761	0.004952	1.105208
7	2	0.649808	0.145052	0.194974	0.912076	-0.770242	-0.202586	1.994333	114	1061.679787	-0.001069	0.641310
7	3	0.642120	0.146245	0.213929	0.993928	-0.794022	0.924841	1.764604	109	1046.097851	-0.002891	0.306538
8	0	0.425111	0.076393	0.209565	1.000000	2.441969	15.590390	1.231104	206	670.667353	-0.000709	0.272740
8	1	0.319789	0.105803	0.033415	0.675182	0.179887	-0.206531	1.872991	174	413.104776	0.001590	0.550795
8	2	0.474417	0.115669	0.000000	0.682022	-2.137756	4.643016	1.558278	195	880.120880	0.000441	0.695207
8	3	0.392571	0.087044	0.151210	0.778521	0.708901	0.824474	1.644960	195	600.836502	-0.001564	0.259542
9	0	0.330023	0.230356	0.117898	1.000000	1.671440	1.328033	1.647857	122	436.858043	0.013412	0.556101
9	1	0.371271	0.169166	0.000000	0.959416	0.397883	0.057725	1.880881	106	454.766775	-0.012293	0.505256
9	2	0.628674	0.132345	0.185474	0.944676	-0.564630	0.122171	1.936180	130	1143.307521	0.007540	0.441600
9	3	0.721887	0.078447	0.533766	0.871462	-0.619352	-0.688107	2.062053	127	1470.041980	0.002913	0.253152
10	0	0.478827	0.183434	0.156349	0.946031	0.659624	-0.485586	2.066546	84	473.524675	0.008951	0.592908
10	1	0.524515	0.215222	0.000000	0.817860	-0.820117	-0.563345	2.017323	92	586.299886	0.002415	1.005530
10	2	0.636709	0.133799	0.174030	0.852173	-1.258546	0.960535	1.827697	94	782.683730	0.001388	0.574217
10	3	0.695106	0.090461	0.507751	1.000000	0.516861	0.135972	1.997953	86	914.904458	0.004327	0.370064

Appendix B

Table of Accuracies Per Pair Tested

Table B.1: Accuracy results for Leave-2-Out validation

Test_1	**Test_2**	**Accuracy**
1	2	0.25
1	3	0.625
1	4	0.5
1	5	0.375
1	6	0.5
1	7	0.625
1	8	0.5
1	9	0.625
1	10	0.375
2	3	0.625
2	4	0.25
2	5	0.25
2	6	0.375
2	7	0.625
2	8	0.5
2	9	0.5

Test 1- Patient	Test 2 - Patient	Accuracy
2	10	0.375
3	4	0.375
3	5	0.75
3	6	0.375
3	7	0.5
3	8	0.5
3	9	0.875
3	10	0.5
4	5	0.25
4	6	0.5
4	7	0.375
4	8	0.5
4	9	0.75
4	10	0.75
5	6	0.5
5	7	0.375
5	8	0.375
5	9	0.5
5	10	0.375
6	7	0.625
6	8	0.5
6	9	0.625
6	10	0.375
7	8	0.5
7	9	0.875
7	10	0.5
8	9	0.625
8	10	0.5

Test 1- Patient	**Test 2 - Patient**	**Accuracy**
9	10	0.5

Appendix C

Tables of Classification Report per Pair Tested

Table C.1: Classification Report for Pair (1,10)

	precision	recall	f1-score	support
0	0.00	0.00	0.00	2.00
1	0.50	0.50	0.50	2.00
2	0.20	0.50	0.29	2.00
3	1.00	0.50	0.67	2.00
macro avg	0.42	0.38	0.36	8.00
weighted avg	0.42	0.38	0.36	8.00

Table C.2: Classification Report for Pair (1,2)

	precision	recall	f1-score	support
0	1.00	0.50	0.67	2.00
1	0.00	0.00	0.00	2.00
2	0.00	0.00	0.00	2.00
3	0.50	0.50	0.50	2.00
macro avg	0.38	0.25	0.29	8.00
weighted avg	0.38	0.25	0.29	8.00

Table C.3: Classification Report for Pair (1,3)

	precision	recall	f1-score	support
0	1.00	1.00	1.00	2.00
1	0.50	0.50	0.50	2.00
2	0.33	0.50	0.40	2.00
3	1.00	0.50	0.67	2.00
macro avg	0.71	0.62	0.64	8.00
weighted avg	0.71	0.62	0.64	8.00

Table C.4: Classification Report for Pair (1,4)

	precision	recall	f1-score	support
0	0.33	0.50	0.40	2.00
1	0.50	1.00	0.67	2.00
2	0.00	0.00	0.00	2.00
3	1.00	0.50	0.67	2.00
macro avg	0.46	0.50	0.43	8.00
weighted avg	0.46	0.50	0.43	8.00

Table C.5: Classification Report for Pair (1,5)

	precision	recall	f1-score	support
0	1.00	0.50	0.67	2.00
1	0.25	0.50	0.33	2.00
2	0.00	0.00	0.00	2.00
3	0.50	0.50	0.50	2.00
macro avg	0.44	0.38	0.38	8.00
weighted avg	0.44	0.38	0.38	8.00

Table C.6: Classification Report for Pair (1,6)

	precision	recall	f1-score	support
0	0.50	0.50	0.50	2.00
1	0.50	1.00	0.67	2.00
2	0.00	0.00	0.00	2.00
3	1.00	0.50	0.67	2.00
macro avg	0.50	0.50	0.46	8.00
weighted avg	0.50	0.50	0.46	8.00

Table C.7: Classification Report for Pair (1,7)

	precision	recall	f1-score	support
0	1.00	0.50	0.67	2.00
1	1.00	0.50	0.67	2.00
2	0.40	1.00	0.57	2.00
3	1.00	0.50	0.67	2.00
macro avg	0.85	0.62	0.64	8.00
weighted avg	0.85	0.62	0.64	8.00

Table C.8: Classification Report for Pair (1,8)

	precision	recall	f1-score	support
0	0.50	0.50	0.50	2.00
1	0.50	1.00	0.67	2.00
2	0.00	0.00	0.00	2.00
3	1.00	0.50	0.67	2.00
macro avg	0.50	0.50	0.46	8.00
weighted avg	0.50	0.50	0.46	8.00

Table C.9: Classification Report for Pair (1,9)

	precision	recall	f1-score	support
0	1.00	0.50	0.67	2.00
1	0.67	1.00	0.80	2.00
2	0.33	0.50	0.40	2.00
3	1.00	0.50	0.67	2.00
macro avg	0.75	0.62	0.63	8.00
weighted avg	0.75	0.62	0.63	8.00

Table C.10: Classification Report for Pair (2,10)

	precision	recall	f1-score	support
0	0.00	0.00	0.00	2.00
1	0.50	0.50	0.50	2.00
2	0.25	0.50	0.33	2.00
3	0.50	0.50	0.50	2.00
macro avg	0.31	0.38	0.33	8.00
weighted avg	0.31	0.38	0.33	8.00

Table C.11: Classification Report for Pair (2,3)

	precision	recall	f1-score	support
0	1.00	0.50	0.67	2.00
1	0.50	0.50	0.50	2.00
2	0.50	1.00	0.67	2.00
3	1.00	0.50	0.67	2.00
macro avg	0.08	0.25	0.12	8.00
weighted avg	0.75	0.62	0.62	8.00

Table C.12: Classification Report for Pair (2,4)

	precision	recall	f1-score	support
0	0.00	0.00	0.00	2.00
1	0.33	0.50	0.40	2.00
2	0.00	0.00	0.00	2.00
3	0.33	0.50	0.40	2.00
macro avg	0.17	0.25	0.20	8.00
weighted avg	0.17	0.25	0.20	8.00

Table C.13: Classification Report for Pair (2,5)

	precision	recall	f1-score	support
0	0.00	0.00	0.00	2.00
1	0.00	0.00	0.00	2.00
2	0.00	0.00	0.00	2.00
3	0.33	1.00	0.50	2.00
macro avg	0.08	0.25	0.12	8.00
weighted avg	0.08	0.25	0.12	8.00

Table C.14: Classification Report for Pair (2,6)

	precision	recall	f1-score	support
0	0.33	0.50	0.40	2.00
1	0.50	0.50	0.50	2.00
2	0.00	0.00	0.00	2.00
3	0.33	0.50	0.40	2.00
macro avg	0.29	0.38	0.33	8.00
weighted avg	0.29	0.38	0.33	8.00

Table C.15: Classification Report for Pair (2,7)

	precision	recall	f1-score	support
0	1.00	0.50	0.67	2.00
1	0.00	0.00	0.00	2.00
2	0.50	1.00	0.67	2.00
3	1.00	1.00	1.00	2.00
macro avg	0.62	0.62	0.58	8.00
weighted avg	0.62	0.62	0.58	8.00

Table C.16: Classification Report for Pair (2,8)

	precision	recall	f1-score	support
0	0.50	0.50	0.50	2.00
1	0.50	0.50	0.50	2.00
2	1.00	0.50	0.67	2.00
3	0.33	0.50	0.40	2.00
macro avg	0.58	0.50	0.52	8.00
weighted avg	0.58	0.50	0.52	8.00

Table C.17: Classification Report for Pair (2,9)

	precision	recall	f1-score	support
0	0.00	0.00	0.00	2.00
1	0.50	0.50	0.50	2.00
2	0.33	0.50	0.40	2.00
3	0.67	1.00	0.80	2.00
macro avg	0.38	0.50	0.42	8.00
weighted avg	0.38	0.50	0.42	8.00

Table C.18: Classification Report for Pair (3,10)

	precision	recall	f1-score	support
0	1.00	0.50	0.67	2.00
1	0.00	0.00	0.00	2.00
2	0.33	1.00	0.50	2.00
3	1.00	0.50	0.67	2.00
macro avg	0.58	0.50	0.46	8.00
weighted avg	0.58	0.50	0.46	8.00

Table C.19: Classification Report for Pair (3,4)

	precision	recall	f1-score	support
0	0.33	0.50	0.40	2.00
1	0.50	1.00	0.67	2.00
2	0.00	0.00	0.00	2.00
3	0.00	0.00	0.00	2.00
macro avg	0.21	0.38	0.27	8.00
weighted avg	0.21	0.38	0.27	8.00

Table C.20: Classification Report for Pair (3,5)

	precision	recall	f1-score	support
0	1.00	0.50	0.67	2.00
1	1.00	0.50	0.67	2.00
2	1.00	1.00	1.00	2.00
3	0.50	1.00	0.67	2.00
macro avg	0.88	0.75	0.75	8.00
weighted avg	0.88	0.75	0.75	8.00

Table C.21: Classification Report for Pair (3,6)

	precision	recall	f1-score	support
0	0.00	0.00	0.00	2.00
1	0.50	1.00	0.67	2.00
2	0.00	0.00	0.00	2.00
3	0.25	0.50	0.33	2.00
macro avg	0.19	0.38	0.25	8.00
weighted avg	0.19	0.38	0.25	8.00

Table C.22: Classification Report for Pair (3,7)

	precision	recall	f1-score	support
0	1.00	1.00	1.00	2.00
1	1.00	0.50	0.67	2.00
2	0.25	0.50	0.33	2.00
3	0.00	0.00	0.00	2.00
macro avg	0.56	0.50	0.50	8.00
weighted avg	0.56	0.50	0.50	8.00

Table C.23: Classification Report for Pair (3,8)

	precision	recall	f1-score	support
0	0.67	1.00	0.80	2.00
1	0.67	1.00	0.80	2.00
2	0.00	0.00	0.00	2.00
3	0.00	0.00	0.00	2.00
macro avg	0.33	0.50	0.40	8.00
weighted avg	0.33	0.50	0.40	8.00

Table C.24: Classification Report for Pair (3,9)

	precision	recall	f1-score	support
0	1.00	1.00	1.00	2.00
1	1.00	1.00	1.00	2.00
2	1.00	0.50	0.67	2.00
3	0.67	1.00	0.80	2.00
macro avg	0.92	0.88	0.87	8.00
weighted avg	0.92	0.88	0.87	8.00

Table C.25: Classification Report for Pair (4,10)

	precision	recall	f1-score	support
0	0.50	1.00	0.67	2.00
1	1.00	1.00	1.00	2.00
2	1.00	0.50	0.67	2.00
3	1.00	0.50	0.67	2.00
macro avg	0.88	0.75	0.75	8.00
weighted avg	0.88	0.75	0.75	8.00

Table C.26: Classification Report for Pair (4,5)

	precision	recall	f1-score	support
0	0.33	0.50	0.40	2.00
1	1.00	0.50	0.67	2.00
2	0.00	0.00	0.00	2.00
3	0.00	0.00	0.00	2.00
macro avg	0.33	0.25	0.27	8.00
weighted avg	0.33	0.25	0.27	8.00

Table C.27: Classification Report for Pair (4,6)

	precision	recall	f1-score	support
0	0.33	1.00	0.50	2.00
1	1.00	0.50	0.67	2.00
2	1.00	0.50	0.67	2.00
3	0.00	0.00	0.00	2.00
macro avg	0.58	0.50	0.46	8.00
weighted avg	0.58	0.50	0.46	8.00

Table C.28: Classification Report for Pair (4,7)

	precision	recall	f1-score	support
0	0.33	0.50	0.40	2.00
1	0.50	0.50	0.50	2.00
2	0.33	0.50	0.40	2.00
3	0.00	0.00	0.00	2.00
macro avg	0.29	0.38	0.33	8.00
weighted avg	0.29	0.38	0.33	8.00

Table C.29: Classification Report for Pair (4,8)

	precision	recall	f1-score	support
0	0.40	1.00	0.57	2.00
1	0.67	1.00	0.80	2.00
2	0.00	0.00	0.00	2.00
3	0.00	0.00	0.00	2.00
macro avg	0.27	0.50	0.34	8.00
weighted avg	0.27	0.50	0.34	8.00

Table C.30: Classification Report for Pair (4,9)

	precision	recall	f1-score	support
0	0.50	1.00	0.67	2.00
1	1.00	1.00	1.00	2.00
2	1.00	0.50	0.67	2.00
3	1.00	0.50	0.67	2.00
macro avg	0.88	0.75	0.75	8.00
weighted avg	0.88	0.75	0.75	8.00

Table C.31: Classification Report for Pair (5,10)

	precision	recall	f1-score	support
0	0.00	0.00	0.00	2.00
1	1.00	0.50	0.67	2.00
2	0.25	0.50	0.33	2.00
3	0.33	0.50	0.40	2.00
macro avg	0.40	0.38	0.35	8.00
weighted avg	0.40	0.38	0.35	8.00

Table C.32: Classification Report for Pair (5,6)

	precision	recall	f1-score	support
0	1.00	0.50	0.67	2.00
1	0.50	0.50	0.50	2.00
2	0.00	0.00	0.00	2.00
3	0.50	1.00	0.67	2.00
macro avg	0.50	0.50	0.46	8.00
weighted avg	0.50	0.50	0.46	8.00

Table C.33: Classification Report for Pair (5,7)

	precision	recall	f1-score	support
0	1.00	0.50	0.67	2.00
1	0.00	0.00	0.00	2.00
2	0.33	1.00	0.50	2.00
3	0.00	0.00	0.00	2.00
macro avg	0.33	0.38	0.29	8.00
weighted avg	0.33	0.38	0.29	8.00

Table C.34: Classification Report for Pair (5,8)

	precision	recall	f1-score	support
0	0.50	0.50	0.50	2.00
1	0.33	0.50	0.40	2.00
2	0.33	0.50	0.40	2.00
3	0.00	0.00	0.00	2.00
macro avg	0.29	0.38	0.33	8.00
weighted avg	0.29	0.38	0.33	8.00

Table C.35: Classification Report for Pair (5,9)

	precision	recall	f1-score	support
0	1.00	0.50	0.67	2.00
1	1.00	0.50	0.67	2.00
2	0.50	0.50	0.50	2.00
3	0.25	0.50	0.33	2.00
macro avg	0.69	0.50	0.54	8.00
weighted avg	0.69	0.50	0.54	8.00

Table C.36: Classification Report for Pair (6,10)

	precision	recall	f1-score	support
0	0.00	0.00	0.00	2.00
1	0.50	1.00	0.67	2.00
2	0.33	0.50	0.40	2.00
3	0.00	0.00	0.00	2.00
macro avg	0.21	0.38	0.27	8.00
weighted avg	0.21	0.38	0.27	8.00

Table C.37: Classification Report for Pair (6,7)

	precision	recall	f1-score	support
0	0.67	1.00	0.80	2.00
1	1.00	0.50	0.67	2.00
2	0.50	1.00	0.67	2.00
3	0.00	0.00	0.00	2.00
macro avg	0.54	0.62	0.53	8.00
weighted avg	0.54	0.62	0.53	8.00

Table C.38: Classification Report for Pair (6,8)

	precision	recall	f1-score	support
0	0.40	1.00	0.57	2.00
1	0.67	1.00	0.80	2.00
2	0.00	0.00	0.00	2.00
3	0.00	0.00	0.00	2.00
macro avg	0.27	0.50	0.34	8.00
weighted avg	0.27	0.50	0.34	8.00

Table C.39: Classification Report for Pair (6,9)

	precision	recall	f1-score	support
0	0.50	1.00	0.67	2.00
1	0.50	0.50	0.50	2.00
2	1.00	0.50	0.67	2.00
3	1.00	0.50	0.67	2.00
macro avg	0.75	0.62	0.62	8.00
weighted avg	0.75	0.62	0.62	8.00

Table C.40: Classification Report for Pair (7,10)

	precision	recall	f1-score	support
0	1.00	0.50	0.67	2.00
1	0.00	0.00	0.00	2.00
2	0.33	1.00	0.50	2.00
3	1.00	0.50	0.67	2.00
macro avg	0.58	0.50	0.46	8.00
weighted avg	0.58	0.50	0.46	8.00

Table C.41: Classification Report for Pair (7,8)

	precision	recall	f1-score	support
0	0.67	1.00	0.80	2.00
1	0.50	0.50	0.50	2.00
2	0.33	0.50	0.40	2.00
3	0.00	0.00	0.00	2.00
macro avg	0.37	0.50	0.42	8.00
weighted avg	0.37	0.50	0.42	8.00

Table C.42: Classification Report for Pair (7,9)

	precision	recall	f1-score	support
0	1.00	1.00	1.00	2.00
1	1.00	0.50	0.67	2.00
2	0.67	1.00	0.80	2.00
3	1.00	1.00	1.00	2.00
macro avg	0.92	0.88	0.87	8.00
weighted avg	0.92	0.88	0.87	8.00

Table C.43: Classification Report for Pair (8,10)

	precision	recall	f1-score	support
0	0.50	0.50	0.50	2.00
1	0.50	0.50	0.50	2.00
2	0.33	0.50	0.40	2.00
3	1.00	0.50	0.67	2.00
macro avg	0.58	0.50	0.52	8.00
weighted avg	0.58	0.50	0.52	8.00

Table C.44: Classification Report for Pair (8,9)

	precision	recall	f1-score	support
0	0.50	1.00	0.67	2.00
1	0.50	0.50	0.50	2.00
2	1.00	0.50	0.67	2.00
3	1.00	0.50	0.67	2.00
macro avg	0.75	0.62	0.62	8.00
weighted avg	0.75	0.62	0.62	8.00

Table C.45: Classification Report for Pair (9,10)

	precision	recall	f1-score	support
0	0.00	0.00	0.00	2.00
1	0.50	0.50	0.50	2.00
2	0.50	1.00	0.67	2.00
3	0.50	0.50	0.50	2.00
macro avg	0.38	0.50	0.42	8.00
weighted avg	0.38	0.50	0.42	8.00