DEPARTMENT OF MECHANICAL ENGINEERING
FACULTY OF ENGINEERING
UNIVERSITY OF MORATUWA

Course:	B.Sc. Eng. (Hons)	Session:	Semester 5
Module Code:	BM3500	Module Title:	Biomechanics
Intake:	22	Laboratory:	Die and Mould Center
Practical No.:	P05+P06	Practical Title:	Biosignal Processing I&II
Practical Coordinator:	Mr. Harindu Bandara	Facilitator(s):	Mr. Harindu Bandara

Biosignal Processing

1. Summary

The analysis of physiological signals is a fundamental component of modern biomedical engineering, providing valuable insights into human health and disease. This practical covers the complete workflow of biosignal analysis, starting with raw data that contains artifacts and progressing to the development of machine learning models for classification.

Part I focuses on the basic principles of Preprocessing and Filtering, addressing the essential need to maintain data quality and reduce noise. Part II expands on this foundation by exploring Feature Engineering and Machine Learning Classification to identify meaningful patterns and create predictive models. The complete protocol is designed for use within the MATLAB environment, utilizing its specialized toolboxes for signal processing and statistical learning

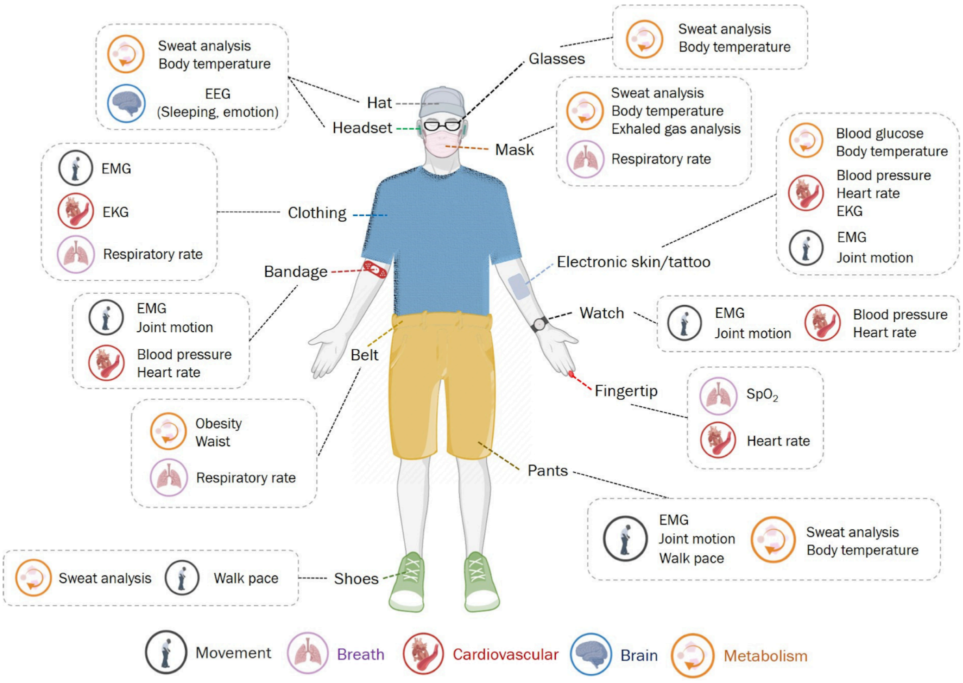


Figure 1: Diversity of Biosignals

2. Biosignal Preprocessing and Filtering

2.1. Introduction

The process of acquiring biosignals naturally exposes physiological measurements to various artifacts and noise. The accuracy of any later analysis depends greatly on applying a careful preprocessing pipeline that improves the signal-to-noise ratio (SNR) and preserves data quality. This first practical introduces the key principles and methods for systematically identifying and addressing common data quality problems, as well as designing digital filters to reduce specific types of noise.

2.2. Learning Objectives

Upon successful completion of this section, students will be able to:

- Implement statistical methods for the detection and handling of outliers and missing data segments.
- Design and apply digital filters for targeted noise removal.
- Analyze and interpret the spectral characteristics of biosignals using frequency-domain visualization.

2.3. Theoretical Background

2.3.1. Data Integrity and Cleaning

The reliability of findings based on biosignals depends directly on the quality of the data. Artifacts from physiological (intrinsic) sources and environmental (extrinsic) sources can either hide important signal features or create false patterns that resemble pathological events.

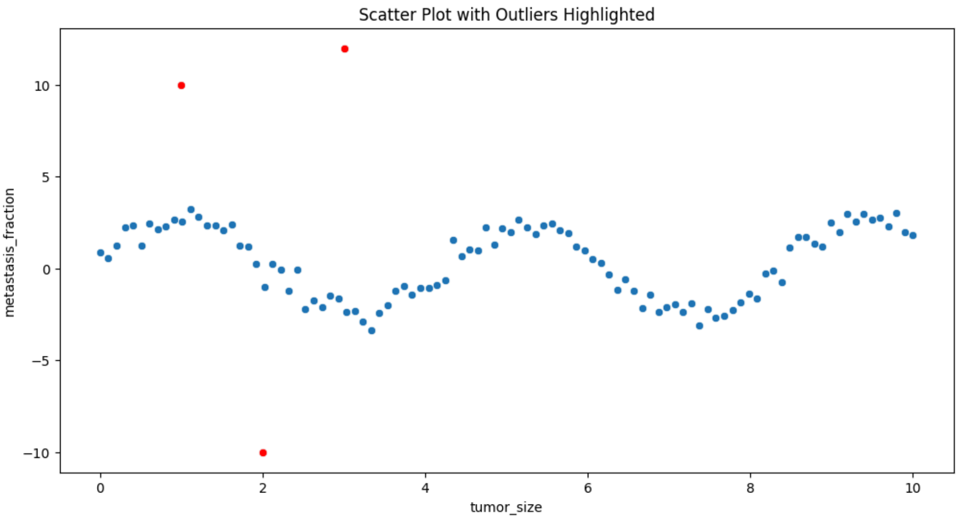


Figure 2: Visualization of outliers

- **Outliers**: Anomalous data points that deviate significantly from the expected statistical distribution. If unaddressed, they can disproportionately influence statistical moments, corrupt frequency-domain estimates, and degrade model performance. Common detection methods include the **Z-score**, the **Interquartile Range (IQR)** method, and the highly robust **Modified Z-score**, which utilizes the median and Median Absolute Deviation (MAD).

- **Missing Values**: Temporal gaps caused by intermittent electrode contact or data transmission errors. These disrupt time-series continuity and preclude the use of standard algorithms like the FFT. Strategies include **Deletion** (which can cause information loss) and **Imputation** via methods like **Linear Interpolation**, **Spline Interpolation**, or the shape-preserving **Piecewise Cubic Hermite Interpolating Polynomial (PCHIP)**.

2.3.2. Digital Filter Design

Digital filters are essential tools for reducing or removing unwanted frequency components from a signal.

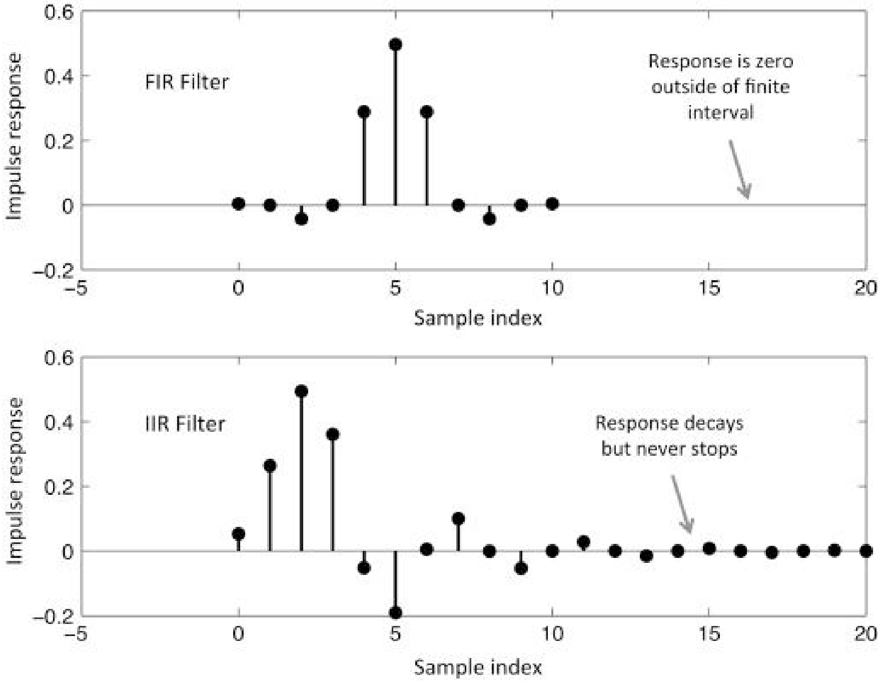


Figure 3: Comparison of FIR and IRR filters

- **Filter Topologies**:
  - **Finite Impulse Response (FIR) Filters**: Non-recursive, inherently stable filters that can be designed to have a perfectly **linear phase response**, which is critical for preserving waveform morphology.
  - **Infinite Impulse Response (IIR) Filters**: Recursive filters that are computationally efficient but exhibit a **non-linear phase response**, which can distort the waveform's shape.

- **Common Filter Approximations**: These include the **Butterworth** (maximally flat passband), **Chebyshev** (steeper roll-off with ripple), and **Elliptic** (sharpest transition with ripple in both bands) filters.

- **Design Process**: An effective filter design follows three steps: 1) **Specification** of filter requirements (e.g., cutoff frequencies, attenuation levels); 2) **Selection and Synthesis** of filter type and order; and 3) **Validation** of the filter's performance in both the frequency and time domains.

2.4. Practical Protocol

2.4.1. Pre-Laboratory Requirements

- **Software**: MATLAB with the **Signal Processing Toolbox** and **DSP System Toolbox**.
- **Dataset**: A raw biosignal dataset with a sampling frequency (f_s) ≥ 250 Hz and a minimum of 1,000 samples.
- **Conceptual Review**: Familiarity with fundamental MATLAB operations and basic signal processing concepts.

2.4.2. Laboratory Procedure

1. **Data Import and Verification**: Load the dataset and verify its integrity, dimensions, and sampling frequency.

2. **Initial Visual and Statistical Inspection**: Plot the raw signals to visually identify gross artifacts and compute descriptive statistics.

3. **Missing Data Handling**: Programmatically detect and impute missing data using an appropriate interpolation method, justifying the choice.
4. **Outlier Detection and Mitigation**: Employ a robust statistical method to identify and replace outliers, justifying the defined threshold.
5. **Frequency-Domain Analysis**: Compute and plot the Power Spectral Density (PSD) of the cleaned signal to identify signal and noise components, which will inform filter design.
6. **Digital Filter Design and Application**: Based on the PSD analysis, specify, design, and apply a digital filter, justifying the choice of FIR vs. IIR.
7. **Post-Filtering Validation**: Generate time-domain overlay plots and a post-filtering PSD to assess noise reduction and quantitatively confirm the attenuation of targeted frequencies.

3. Feature Engineering and Machine Learning Classification

3.1. Introduction

With clean, preprocessed data as a starting point, this second practical covers the next stages of the analysis pipeline: feature engineering and machine learning classification. The goal is to extract quantitative, distinctive information from the processed signals and use it to build predictive models for automated classification.

3.2. Learning Objectives

Upon successful completion of this section, students will be able to:

- Implement a diverse set of feature extraction techniques from the time, frequency, and time-frequency domains.
- Apply Principal Component Analysis (PCA) for dimensionality reduction and data visualization.
- Systematically evaluate and compare classifier performance using standard validation protocols and metrics.

3.3. Theoretical Background

3.3.1. Feature Engineering

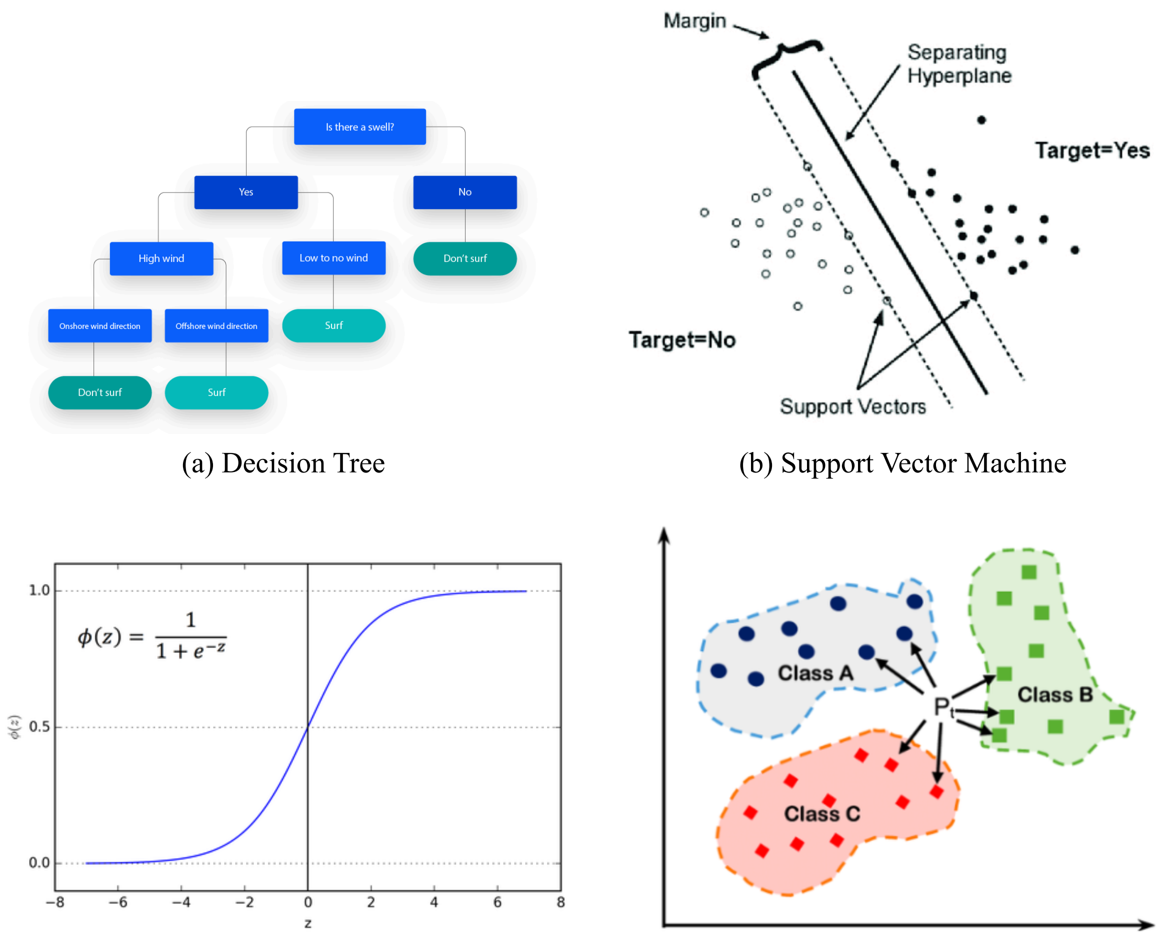
Feature engineering reduces high-dimensional data into a concise but informative set of descriptive features.

- **Time-Domain Features**: Derived from the signal's time-series. Examples include **Statistical Moments** (RMS, variance, skewness, kurtosis) and measures of **Temporal Dynamics** (Zero Crossing Rate, Waveform Length).
- **Frequency-Domain Features**: Derived from the signal's Power Spectral Density (PSD). Examples include **Band Power** in specific physiological bands and **Spectral Descriptors** like Spectral Centroid and Spectral Flatness.
- **Advanced Features**: For non-stationary signals, **Time-Frequency** representations (from STFT or Wavelets) and **Nonlinear Features** like **Entropy Measures** (Shannon, Approximate, Sample) are used to quantify signal complexity and predictability.

3.3.2. Dimensionality Reduction: Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is an orthogonal linear transformation that maps a high-dimensional feature set onto a new, lower-dimensional coordinate system composed of principal components (PCs). These components are arranged according to the amount of variance they explain in the data, enabling dimensionality reduction while preserving as much information as possible.

3.3.3. Classification Algorithms



(a) Decision Tree

(b) Support Vector Machine

(c) Logistic Regression

(d) k-Nearest Neighbors

Figure 4: Classification Models

- **Decision Trees (DT)**: Interpretable hierarchical models that classify data via recursive partitioning of the feature space.
- **Support Vector Machines (SVM)**: Algorithms that find an optimal hyperplane maximizing the margin between classes, using the "kernel trick" for non-linear problems.
- **Logistic Regression (LR)**: A generalized linear model that models the posterior probability of a class using the logistic function.
- **k-Nearest Neighbors (k-NN)**: A non-parametric method that classifies data based on the majority class of its k nearest neighbors in the feature space.

3.3.4. Performance Evaluation

Classifier performance is evaluated using metrics derived from a **confusion matrix**. Key metrics include **Accuracy, Sensitivity (Recall), Specificity, Precision,** and the **F1-Score**. For binary tasks, **Receiver Operating Characteristic (ROC)** curve analysis provides a threshold-independent measure of performance, quantified by the **Area Under the Curve (AUC)**.

3.4. Practical Protocol

1. **Data Loading and Structuring**: Load the preprocessed data and corresponding class labels, verifying the data structure.

2. **Comprehensive Feature Extraction**: Systematically extract a multi-domain feature set (time, frequency, etc.) for each signal segment.
3. **Feature Matrix Consolidation and Normalization**: Assemble features into a single matrix and normalize it using Z-score standardization.
4. **Dimensionality Reduction via PCA**: Apply PCA to the normalized feature matrix. Analyze the explained variance profile to select an appropriate number of principal components.
5. **Classifier Training and Validation**: Import the reduced-dimension data and configure a cross-validation scheme. Train a suite of classifiers (DT, SVM, LR, k-NN).
6. **Performance Evaluation and Model Selection**: Compare the cross-validated accuracy of all models. Export the best-performing model for detailed analysis. Generate its confusion matrix and calculate key performance metrics (Accuracy, Sensitivity, F1-Score, etc.). Generate and interpret the ROC curve and AUC.
7. **Feature Importance Analysis**: For applicable models, extract and analyze predictor importance scores to identify the most discriminative features and relate them to the underlying physiology.

4. Final Deliverables and Assessment

Upon completion of both practicals, students are required to submit a single report (PDF format) containing the following items, representing the entire process:

- **Methodology**: A detailed description of the entire pipeline, from preprocessing choices (e.g., filter design) to feature engineering, PCA parameter selection, and the classification scheme.
- **Results**: A clear presentation of all key findings. This includes figures from the preprocessing stage (e.g., pre/post PSD plots) and the classification stage (e.g., confusion matrix, ROC curve, feature importance plot), all labeled. A summary table of classifier performance metrics is required.
- **Discussion**: A holistic, critical interpretation of the results. This section should evaluate the effectiveness of the full pipeline, discuss the trade-offs between different models, and identify the limitations and potential future work.