

# Enhanced Automatic Feature Extraction in Heterogeneous Biomedical Time Series for Optimization of Clinical Monitoring and Diagnosis

---

Erweiterte automatische Merkmalsextraktion in heterogenen  
biomedizinischen Zeitreihen zur Optimierung des klinischen  
Monitorings und der Diagnose

Von der Fakultät für Maschinenwesen der Rheinisch-Westfälischen  
Technischen Hochschule Aachen zur Erlangung des akademischen  
Grades eines Doktors der Naturwissenschaften genehmigte Dissertation

vorgelegt von

Pejman Farhadi Ghalati

Berichter:     Universitätsprofessor Dr.rer.nat. Andreas Schuppert  
                  Universitätsprofessor Dr.-Ing. Ulrich Steinseifer

Tag der mündlichen Prüfung: 17.10.2025

Diese Dissertation ist auf den Internetseiten der Universitätsbibliothek online verfügbar.

## Abstract

The prevalence of real-time data acquisition has increased significantly in recent years due to advancements in sensors, transmission, and storage technologies. This has led to the collection of large datasets in various fields, including engineering, medical, physical, and social sciences. In healthcare, dynamic monitoring data has become a crucial aspect for health diagnosis, treatment, and proactive health tracking. The emergence of wearable technology and monitoring systems has provided new avenues for understanding complex dynamic processes through longitudinal data, providing insight into an individual's health and comfort. However, it is important to note that data quality issues can significantly impact temporal data analysis.

Additionally, the analysis strategies required for monitoring data vary greatly between monitoring a few parameters at high sampling rates and monitoring several parameters over extended periods at low sampling frequencies. One approach for improving such data analysis is to represent the data as features, enhancing the interoperability of outcomes from statistical and machine learning analyses.

This thesis explores the domain knowledge feature engineering of monitoring time series data through a Python package, developed to provide a robust and systematic temporal data analysis platform. The package includes functional elements ranging from data reading to denoising and feature extraction, and its modular design allows for diverse applications and the incorporation of user-defined components. The findings of this thesis demonstrate the capability of temporal data abstraction using a methodical feature extraction scheme. The developed workflow provides a robust and user-friendly architecture for conducting time series analysis.

## Zusammenfassung

Die Verfügbarkeit von Echtzeit-Datenerfassung hat in den letzten Jahren aufgrund von Fortschritten in Sensor-, Übertragungs- und Speichertechnologien erheblich zugenommen. Dies führte zur Erhebung umfangreicher Datensätze in unterschiedlichen Bereichen wie Ingenieurwesen, Medizin, Natur- und Sozialwissenschaften. Im Gesundheitswesen ist die Erfassung dynamischer Überwachungsdaten zu einem zentralen Element für Gesundheitsdiagnosen, Therapien und proaktives Gesundheitsmonitoring geworden. Das Aufkommen tragbarer Technologien und Überwachungssysteme hat neue Möglichkeiten eröffnet, komplexe dynamische Prozesse anhand von Längsschnittdaten besser zu verstehen und Einblicke in den Gesundheitszustand sowie das Wohlbefinden Einzelner zu gewinnen. Es ist jedoch zu beachten, dass Probleme bei der Datenqualität die zeitliche Datenanalyse erheblich beeinflussen können.

Darüber hinaus unterscheiden sich die erforderlichen Analysestrategien für Überwachungsdaten je nach Einsatzgebiet stark: Sie reichen von der Beobachtung weniger Parameter mit hohen Abtastraten bis hin zur Überwachung mehrerer Parameter über längere Zeiträume bei niedrigen Abtastraten. Ein Ansatz zur Verbesserung solcher Datenanalysen besteht darin, die Daten in Form von Merkmalen darzustellen. Dies erhöht die Interoperabilität der Ergebnisse sowohl für statistische Verfahren als auch für Methoden des maschinellen Lernens.

Diese Dissertation untersucht das domänenspezifische Feature-Engineering für die Analyse von Zeitreihendaten im Überwachungskontext anhand eines Python-Pakets. Dieses Paket wurde entwickelt, um eine robuste und systematische Plattform für zeitliche Datenanalysen bereitzustellen. Es umfasst funktionale Elemente von der Dateneinlesung über die Rauschunterdrückung bis hin zur Merkmalsextraktion. Dank seiner modularen Architektur kann es für vielfältige Anwendungen genutzt und um benutzerdefinierte Komponenten erweitert werden. Die Ergebnisse dieser Arbeit demonstrieren die Fähigkeit zur Abstraktion zeitlicher Daten mithilfe eines methodischen Schemas zur Merkmalsextraktion. Der entwickelte Workflow bietet eine robuste und benutzerfreundliche Architektur, um Zeitreihenanalysen effizient durchzuführen.

## List of Publications

Parts of this thesis have been previously published in five research articles, with four of them listing me as the first author:

- **Pejman Farhadi Ghalati**, Satya S Samal, Jayesh S Bhat, Robert Deisz, Gernot Marx, Andreas Schuppert. Critical transitions in intensive care units: a sepsis case study. Scientific reports. 2019 Sep 9.  
<https://www.nature.com/articles/s41598-019-49006-2>
- **Pejman Farhadi Ghalati**, Moein E. Samadi, Marlo Verket, Paul Balfanz, Dirk Müller-Wieland, Stephan Jonas, Andreas Napp, Christoph Wanner, Markus Ketteler, Athina Vassiliadou, Stefan Heidenreich, Thomas Deserno, Gudrun Hetzel, Danilo Fliser, Malte Kelm, Jürgen Floege, Nikolaus Marx Andreas Schuppert. Monitoring individualized glucose levels predicts risk for bradycardia in type 2 diabetes patients with chronic kidney disease: a pilot study. Scientific reports. 2024 Dec 5.  
<https://www.nature.com/articles/s41598-024-81983-x>
- **Pejman Farhadi Ghalati** and Andreas Schuppert. CMDA: a tool for Continuous Monitoring Data Analysis. ArXive. 2023 Oct 22.  
<https://arxiv.org/abs/2310.14427>
- **Pejman Farhadi**, Konstantin Sharafutdinov, Jayesh Sudhir Bhat Andreas Schuppert. Big Data und künstliche Intelligenz in der Medizin. Telemedizin: Grundlagen und praktische Anwendung in stationären und ambulanten Einrichtungen. 2020 Dec 01.  
[https://link.springer.com/chapter/10.1007/978-3-662-60611-7\\_37](https://link.springer.com/chapter/10.1007/978-3-662-60611-7_37)
- Konstantin Sharafutdinov, Sebastian Johannes Fritsch, Mina Iravani, **Pejman Farhadi Ghalati**, Sina Saffaran, Declan G Bates, Jonathan G Hardman, Richard Polzin, Hannah Mayer, Gernot Marx, Johannes Bickenbach, Andreas Schuppert. Computational simulation of virtual patients reduces dataset bias and improves machine learning-based detection of ARDS from noisy heterogeneous ICU datasets. IEEE Open Journal of Engineering in Medicine and Biology. 2023 Feb 8.  
<https://ieeexplore.ieee.org/abstract/document/10040737>

The first article serves as the foundation for Chapter 8. The second article contributed to Chapter 7. The third article provides a brief introduction and instructions for using the software described in 2. The fourth article is a review on big data and AI in medicine, parts of which are incorporated into the introduction chapter of this

thesis. Finally, the clustering methodology from the fifth article is the basis of the clustering approach in Chapter [6](#).

## Acknowledgments

First and foremost, I would like to express my deepest gratitude to my supervisor, Professor Andreas Schuppert, for his invaluable guidance, support, and encouragement throughout my PhD journey. His kindness, patience and insightful advice not only shaped my research, but also made this journey an enriching and enjoyable experience. I am truly grateful for his mentorship and constant support.

I would also like to extend my heartfelt appreciation to my colleagues and friends, Ali Hadizadeh, Maryam Montazeri, Moein Samadi, and Lisa Schätzle, for their support, collaboration, and friendship. Their helpful discussions, shared experiences, and encouragement have played a significant role in making this journey more fulfilling.

Last but certainly not least, I am deeply grateful to my parents, sisters, and my wife, Sanaz, for their unconditional love, patience, and support. Their strong belief in me has been my greatest source of strength and motivation.

This journey would not have been possible without the support of these incredible individuals, and I am truly fortunate to have had them by my side.

# List of Tables

|     |                                                                                                                                                 |     |
|-----|-------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| 4.1 | A sample signal with 128 samples, frequency range 0 to $f_n$ and 4 levels of decomposition and 5 output scales . . . . .                        | 31  |
| 5.1 | A 2-by-2 contingency table, where X and Y are sample groups, and A and B are the categories . . . . .                                           | 34  |
| 6.1 | Numbers of ECG records of individual diagnosis classes . . . . .                                                                                | 48  |
| 6.2 | Numbers of ECG records of individual diagnosis subclasses . . . . .                                                                             | 49  |
| 6.3 | ECG spectral features classification results . . . . .                                                                                          | 52  |
| 6.4 | ECG SWT features classification results . . . . .                                                                                               | 53  |
| 6.5 | ECG data clusters population ratio, give K=2 . . . . .                                                                                          | 56  |
| 6.6 | ECG data clusters population ratio, give K=3 . . . . .                                                                                          | 56  |
| 6.7 | ECG data clusters population ratio, give K=5 . . . . .                                                                                          | 57  |
| 6.8 | Classification metrics of CD prediction in individual clusters . . . . .                                                                        | 57  |
| 7.1 | HRV time-domain features . . . . .                                                                                                              | 69  |
| 7.2 | HRV frequency domain features . . . . .                                                                                                         | 70  |
| 7.3 | HRV nonlinear features . . . . .                                                                                                                | 71  |
| 7.4 | Bradycardia risk prediction scores . . . . .                                                                                                    | 81  |
| 7.5 | The mean and standard deviation of performance measures of the Random Forests classifier trained on normalized data and non-normalized. . . . . | 83  |
| 8.1 | Mean sampling rate of the clinical variables . . . . .                                                                                          | 90  |
| 8.2 | $SL$ , $AC1$ and $VAR$ statistical test bootstrap frequencies . . . . .                                                                         | 103 |
| A.1 | Built-in preprocessing features . . . . .                                                                                                       | 110 |
| A.2 | Built-in time-domain features . . . . .                                                                                                         | 111 |
| A.3 | Built-in frequency-domain features . . . . .                                                                                                    | 111 |
| A.4 | Built-in entropy features . . . . .                                                                                                             | 112 |

---

|     |                                                                         |     |
|-----|-------------------------------------------------------------------------|-----|
| C.1 | [Classification metrics of NORM prediction in individual clusters . . . | 123 |
| C.2 | [Classification metrics of CD prediction in individual clusters . . . . | 124 |
| C.3 | Classification metrics of HYP prediction in individual clusters . . . . | 124 |
| C.4 | Classification metrics of MI prediction in individual clusters . . . .  | 124 |
| C.5 | [Classification metrics of STTC prediction in individual clusters . . . | 125 |
| D.1 | Demographic data of our septic shock and non-sepsis cohorts . . . .     | 127 |

# List of Figures

|      |                                                                                                                                                                                              |    |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| 2-1  | CMDA workflow . . . . .                                                                                                                                                                      | 13 |
| 4-1  | SWT filter bank . . . . .                                                                                                                                                                    | 32 |
| 5-1  | SVM kernels example . . . . .                                                                                                                                                                | 38 |
| 5-2  | Random Forests schema . . . . .                                                                                                                                                              | 38 |
| 5-3  | K-fold cross-validation schema . . . . .                                                                                                                                                     | 39 |
| 5-4  | Shuffled Stratified cross-validation . . . . .                                                                                                                                               | 40 |
| 6-1  | ECG typical 10 electrodes configuration . . . . .                                                                                                                                            | 46 |
| 6-2  | Heart conduction system . . . . .                                                                                                                                                            | 47 |
| 6-3  | QRS complex frequency range of a healthy ECG signal . . . . .                                                                                                                                | 50 |
| 6-4  | Flowchart of ECG signal decomposition using 9 scale SWT . . . . .                                                                                                                            | 52 |
| 6-5  | ECG classification AUCs of diagnosis subclasses . . . . .                                                                                                                                    | 53 |
| 6-6  | Normal class most significantly expressed features . . . . .                                                                                                                                 | 54 |
| 6-7  | Diagnosis classes significantly expressed features . . . . .                                                                                                                                 | 59 |
| 6-8  | Important ECG-leads . . . . .                                                                                                                                                                | 60 |
| 6-9  | ECG Data clusters size . . . . .                                                                                                                                                             | 61 |
| 6-10 | Enrichment scores of individual diagnosis classes in each cluster. high positive values (red color) indicates enrichment and low negative values (blue color) shows low presence. . . . .    | 62 |
| 6-11 | Enrichment scores of individual diagnosis subclasses in each cluster. high positive values (red color) indicates enrichment and low negative values (blue color) shows low presence. . . . . | 62 |
| 6-12 | CD diagnosis classification results of individual clusters . . . . .                                                                                                                         | 63 |
| 7-1  | Distribution of bradycardia episodes per patients . . . . .                                                                                                                                  | 66 |
| 7-2  | Number of bradycardia events in all patients at every hour of the day . . . . .                                                                                                              | 67 |
| 7-3  | Percentage of bradycardia events at absolute glucose levels . . . . .                                                                                                                        | 72 |

|      |                                                                                                                                 |     |
|------|---------------------------------------------------------------------------------------------------------------------------------|-----|
| 7-4  | Glucose range and profile in patients with bradyarrhythmias . . . . .                                                           | 73  |
| 7-5  | Distribution of glucose levels . . . . .                                                                                        | 74  |
| 7-6  | Percentage of bradycardia events at different glucose tertiles . . . . .                                                        | 74  |
| 7-7  | Relative glucose tertiles enrichment . . . . .                                                                                  | 75  |
| 7-8  | Heart rate and glucose average over 24 hours . . . . .                                                                          | 75  |
| 7-9  | Percentage of data points at different glucose tertiles in 3 hours' time intervals . . . . .                                    | 76  |
| 7-10 | Percentage of bradycardia events at different glucose tertiles in 3 hours time intervals . . . . .                              | 76  |
| 7-11 | Distribution of p-values at low and high relative glucose tertiles in 3 hours time intervals . . . . .                          | 77  |
| 7-12 | Heatmap of HRV extracted features pairwise correlation . . . . .                                                                | 78  |
| 7-13 | HRV features significance test p-values . . . . .                                                                               | 79  |
| 7-14 | Bradycardia risk zone and normal zone identification . . . . .                                                                  | 80  |
| 7-15 | Bradycardia risk prediction ROC curves . . . . .                                                                                | 80  |
| 7-16 | Bradycardia risk prediction scores . . . . .                                                                                    | 81  |
| 7-17 | The recall-precision curve of Random Forests classifier trained on normalized features . . . . .                                | 82  |
| 7-18 | The recall-precision curve of Random Forests classifier . . . . .                                                               | 83  |
| 7-19 | Important features in bradycardia risk prediction . . . . .                                                                     | 84  |
| 8-1  | Sepsis-3 definition . . . . .                                                                                                   | 89  |
| 8-2  | A schematic diagram for the calculation of $SL$ , $SLMean$ , and $t_{max}$ . . . . .                                            | 93  |
| 8-3  | An example of $SLMean$ calculation from a synthetic dataset . . . . .                                                           | 95  |
| 8-4  | A schematic representing our data-sampling strategy using $SLMean$ . . . . .                                                    | 96  |
| 8-5  | The changes over time in a set of variables and the corresponding calculated $SLMean^{ic}$ of a sample septic patient . . . . . | 98  |
| 8-6  | Componentwise mean and standard deviation of $SLMean$ . . . . .                                                                 | 99  |
| 8-7  | Componentwise mean and standard deviation of $AC1$ and $VAR$ . . . . .                                                          | 100 |
| 8-8  | Distribution of the critical transitions times from the onset time of septic shock . . . . .                                    | 101 |
| 8-9  | Robustness of $SL$ procedure . . . . .                                                                                          | 102 |
| 8-10 | Statistical test bootstrap frequencies of of each variable . . . . .                                                            | 104 |
| B-1  | Imported ECG and ABP sample signals . . . . .                                                                                   | 114 |
| D-1  | Variables profiles in septic and non-septic cohorts . . . . .                                                                   | 128 |

|     |                                                               |     |
|-----|---------------------------------------------------------------|-----|
| D-2 | Variables profiles in septic and non-septic cohorts . . . . . | 129 |
| D-3 | Variables profiles in septic and non-septic cohorts . . . . . | 130 |
| D-4 | Variables profiles in septic and non-septic cohorts . . . . . | 131 |
| D-5 | Variables profiles in septic and non-septic cohorts . . . . . | 132 |



# Contents

|          |                                           |           |
|----------|-------------------------------------------|-----------|
| <b>1</b> | <b>Introduction</b>                       | <b>1</b>  |
| 1.1      | Big Data in Healthcare . . . . .          | 1         |
| 1.2      | RWE . . . . .                             | 2         |
| 1.3      | Biosignals . . . . .                      | 3         |
| 1.4      | Time-Series Data Analysis . . . . .       | 5         |
| 1.5      | Time-Series Feature Engineering . . . . . | 7         |
| 1.6      | Time-Series Analysis Tools . . . . .      | 8         |
| 1.7      | Objective of the work . . . . .           | 9         |
| 1.8      | Outline . . . . .                         | 11        |
| <b>2</b> | <b>Concept of the software Platform</b>   | <b>13</b> |
| 2.1      | Structure . . . . .                       | 13        |
| 2.2      | Data Importer Module . . . . .            | 13        |
| 2.3      | Preprocessing Module . . . . .            | 14        |
| 2.4      | Feature Extraction Module . . . . .       | 14        |
| 2.5      | Pipeline . . . . .                        | 14        |
| <b>3</b> | <b>Preprocessing Methods</b>              | <b>17</b> |
| 3.1      | Normalization . . . . .                   | 17        |
| 3.2      | Missing Value Imputation . . . . .        | 18        |
| 3.3      | Windowing . . . . .                       | 18        |
| <b>4</b> | <b>Features</b>                           | <b>21</b> |
| 4.1      | Time-Domain Features . . . . .            | 21        |
| 4.1.1    | Main Statistical Features . . . . .       | 21        |
| 4.1.2    | Peak to Peak distance . . . . .           | 22        |
| 4.1.3    | Root Mean Square . . . . .                | 22        |
| 4.1.4    | Zero Crossing Rate . . . . .              | 22        |

|          |                                                                        |           |
|----------|------------------------------------------------------------------------|-----------|
| 4.1.5    | Auto-correlation . . . . .                                             | 22        |
| 4.2      | Spectral Features . . . . .                                            | 23        |
| 4.2.1    | Power Spectrum Energy . . . . .                                        | 23        |
| 4.2.2    | Mean Frequency . . . . .                                               | 23        |
| 4.2.3    | Median Frequency . . . . .                                             | 24        |
| 4.2.4    | Variance of Central Frequency . . . . .                                | 24        |
| 4.2.5    | Peak Frequency . . . . .                                               | 25        |
| 4.2.6    | Power Spectrum Ratio . . . . .                                         | 25        |
| 4.2.7    | Frequency Band Features . . . . .                                      | 25        |
| 4.3      | Entropy Features . . . . .                                             | 26        |
| 4.3.1    | Sample Entropy . . . . .                                               | 26        |
| 4.3.2    | Permutation Entropy . . . . .                                          | 27        |
| 4.3.3    | Spectral Entropy . . . . .                                             | 28        |
| 4.4      | Wavelet Features . . . . .                                             | 29        |
| 4.4.1    | Discrete Wavelet Transform . . . . .                                   | 29        |
| 4.4.2    | SWT . . . . .                                                          | 31        |
| <b>5</b> | <b>Statistical and Machine Learning Models</b>                         | <b>33</b> |
| 5.1      | Statistical Tests . . . . .                                            | 33        |
| 5.1.1    | Wlicoxon Ranksum Test . . . . .                                        | 33        |
| 5.1.2    | Fisher's Exact Test . . . . .                                          | 34        |
| 5.2      | Classification . . . . .                                               | 34        |
| 5.2.1    | Logistic Regression . . . . .                                          | 35        |
| 5.2.2    | Support Vector Machines . . . . .                                      | 37        |
| 5.2.3    | Random Forests . . . . .                                               | 37        |
| 5.3      | Cross Validation . . . . .                                             | 39        |
| 5.4      | Clustering . . . . .                                                   | 41        |
| 5.4.1    | Consensus Clustering . . . . .                                         | 41        |
| 5.4.2    | Clusters Enrichment Analysis . . . . .                                 | 43        |
| <b>6</b> | <b>Application Showcase: ECG Signals Classification and Clustering</b> | <b>45</b> |
| 6.1      | Introduction . . . . .                                                 | 45        |
| 6.2      | ECG Signal . . . . .                                                   | 45        |
| 6.3      | Data . . . . .                                                         | 47        |
| 6.4      | Feature Extraction . . . . .                                           | 48        |
| 6.4.1    | Spectral Features . . . . .                                            | 49        |
| 6.4.2    | SWT Features . . . . .                                                 | 50        |

|          |                                                                                                       |           |
|----------|-------------------------------------------------------------------------------------------------------|-----------|
| 6.5      | Results . . . . .                                                                                     | 51        |
| 6.5.1    | Classification . . . . .                                                                              | 51        |
| 6.5.2    | Clustering . . . . .                                                                                  | 55        |
| 6.6      | Discussions . . . . .                                                                                 | 57        |
| <b>7</b> | <b>Application Showcase: Risk of bradycardia during hypoglycemia in patients with type 2 diabetes</b> | <b>65</b> |
| 7.1      | Introduction . . . . .                                                                                | 65        |
| 7.2      | Data . . . . .                                                                                        | 66        |
| 7.3      | Heart Rate Variability Analysis . . . . .                                                             | 67        |
| 7.3.1    | HRV Time domain features . . . . .                                                                    | 68        |
| 7.3.2    | HRV Spectral domain features . . . . .                                                                | 68        |
| 7.3.3    | HRV Nonlinear features . . . . .                                                                      | 68        |
| 7.4      | Data Preprocessing and Feature Extraction . . . . .                                                   | 70        |
| 7.4.1    | Windowing . . . . .                                                                                   | 70        |
| 7.4.2    | Filtering . . . . .                                                                                   | 71        |
| 7.4.3    | Missing value imputation . . . . .                                                                    | 72        |
| 7.5      | Results . . . . .                                                                                     | 72        |
| 7.5.1    | Association between glucose level and bradycardia events . . .                                        | 72        |
| 7.5.2    | Association between HRV features and Bradycardia events . .                                           | 77        |
| 7.5.3    | Forecasting the Bradycardia events . . . . .                                                          | 78        |
| 7.6      | Discussion . . . . .                                                                                  | 83        |
| <b>8</b> | <b>Application Showcase: Critical Transition in Septic Patients</b>                                   | <b>87</b> |
| 8.1      | Introduction . . . . .                                                                                | 87        |
| 8.2      | Data . . . . .                                                                                        | 88        |
| 8.3      | Methods . . . . .                                                                                     | 91        |
| 8.3.1    | Time Binning . . . . .                                                                                | 91        |
| 8.3.2    | Missing Value Imputation . . . . .                                                                    | 91        |
| 8.3.3    | Data Normalization . . . . .                                                                          | 91        |
| 8.3.4    | Surprise Loss . . . . .                                                                               | 91        |
| 8.3.5    | Data-sampling strategy with SLMean . . . . .                                                          | 94        |
| 8.3.6    | Bootstrapping . . . . .                                                                               | 95        |
| 8.3.7    | Autocorrelation and variance as early-warning signals . . . .                                         | 96        |
| 8.4      | Results . . . . .                                                                                     | 97        |
| 8.4.1    | SLMean as an early-warning indicator . . . . .                                                        | 97        |
| 8.4.2    | Robustness of the SL calculation . . . . .                                                            | 99        |

|          |                                               |            |
|----------|-----------------------------------------------|------------|
| 8.4.3    | SLMean-based data-sampling strategy . . . . . | 99         |
| 8.5      | Discussions . . . . .                         | 101        |
| <b>9</b> | <b>Summary and Outlooks</b>                   | <b>105</b> |
| <b>A</b> | <b>Appendix A</b>                             | <b>109</b> |
| A.1      | Data Importer Module . . . . .                | 109        |
| A.2      | Preprocessing Module . . . . .                | 109        |
| A.3      | Feature Extraction Module . . . . .           | 110        |
| A.3.1    | Time-domain Features . . . . .                | 110        |
| A.3.2    | Frequency-domain Features . . . . .           | 111        |
| A.4      | Entropy Features . . . . .                    | 112        |
| A.5      | Wavelet Features . . . . .                    | 112        |
| A.6      | Feature Engineering Pipeline . . . . .        | 112        |
| <b>B</b> | <b>Appendix B</b>                             | <b>113</b> |
| B.1      | Feature Engineering . . . . .                 | 113        |
| B.1.1    | Feature Object . . . . .                      | 113        |
| B.1.2    | Multivariate Feature Extraction . . . . .     | 117        |
| B.1.3    | Feature Extraction Pipeline . . . . .         | 120        |
| <b>C</b> | <b>Appendix C</b>                             | <b>123</b> |
| C.1      | NORM classification . . . . .                 | 123        |
| C.2      | HYP classification . . . . .                  | 123        |
| <b>D</b> | <b>Appendix D</b>                             | <b>127</b> |
|          | <b>Bibliography</b>                           | <b>147</b> |

# Introduction

## 1.1 Big Data in Healthcare

One may call the 21st century the era of data and digitalization. We can observe the rapidly increasing data availability in all aspects of our lives. From our social interaction to public policy and safety, from travel and transportation to news and information, and from education and employment to cutting-edge sciences.

Due to the exponential growth in the amount and variety of generated data, also referred to as Big Data, and their management complexity, we require new strategies and resources for governing these data. New technologies, accompanied by advancements in building high capacity physical storage, have already demonstrated promising solutions for Big data collection and storing [1]. Nevertheless, today's challenge is no longer just generating and collecting more data but transforming this data into real value by extracting hidden and complex relationships from the data, yielding new information [2, 3].

The vision of Big Data shifts the paradigm of data analysis from testing determined hypotheses to hypothesis-generating data analysis, achieved by machine learning (ML) techniques. ML, a subcategory of artificial intelligence, is based on the concept that systems can learn from data, recognize patterns and make decisions in an automated approach. In other words, ML models learn quantitative patterns from a number of observations without understanding the underlying mechanisms. ML allows the exploitation of Big Data at multiple scales to abandon a priori hypotheses and represent the real world in a practically unbiased way[4, 5].

One of the areas that has also been flooded by the explosion of data is the healthcare system and medical services [6]. The expectations for Big Data in medicine are based on the immense complexity of the mechanisms that initiate the onset and development of diseases in individual patients. The progression of diseases is not only controlled by molecular level factors but by a multitude of heterogeneous elements,

e.g., genetic background, lifestyle, stress factors, and comorbidities. Elaborated and established workflows drive the development of therapies targeting core molecular processes in laboratory research and clinical trials. In particular, clinical trials are based on sophisticated protocols, designed by requirements arising from statistical needs, with few patients compared to the overall patient population. Therefore, data gathered in clinical trials are selected concerning a detailed hypothesis and rarely represent the total cohort of patients in actual clinical practice [7]. Therefore, by analyzing data from clinical practice, so-called real-world evidence (RWE) analysis, which reflects the actual and unbiased patient population, using ML-based tools for data analysis, there is hope to overcome the limitations of established research and development workflows to boost precision medicine. Precision medicine, also known as personalized medicine, is a medical approach that utilizes an individual's unique characteristics, such as genetics and lifestyle, to develop tailored treatment strategies with the goal of improving health outcomes. This approach to healthcare utilizes large-scale data analysis to identify specific biomarkers that can inform treatment decisions. By leveraging this information, precision medicine enables the creation of more targeted, effective therapies and can potentially reduce the need for hospitalization by facilitating more effective healthcare management at home [8].

## 1.2 RWE

As explained in the previous section, Real World Evidence (RWE) in medicine implies evidence conveyed from real-world data (RWD).

RWD are collected and documented data from several sources of heterogeneous patient populations in real-world environments during clinical practice rather than data generated in the context of randomized controlled trials (RCTs) [9].

The data structures of Real World Evidence datasets are very heterogeneous. They can contain various data types, as follows:

- Static and single recorded data (e.g., patient demographic, biometric data, admission date, medical images, free text forms and diagnosis)
- Low sampling rate time series (e.g., laboratory parameters, medication prescriptions)
- High sampling rate time series (e.g., ECG, EEG, Respiratory rate, Blood pressure)

Such data multiformity results in a combination of static and temporal data with very different sampling densities. In terms of volume, RWE- databases may include

from hundreds to millions of patients and from a few to thousands of parameters per patient [10]. More detailed and small databases are usually related to specific indications or medical areas such as intensive care. In contrast, gigantic databases with several million patients are mostly limited to electronic medical records collected by healthcare providers and health insurers. For instance, the IQVIA EMR US database comprises records of 30 million active patients treated by more than 30,000 healthcare providers in the United States. The database includes medical and pharmacy service data files that contain information on patient demographics, diagnoses, prescription medications and comorbidities [11].

Since RWE datasets cover broad patient cohorts, they have potential applications in rare diseases and syndromes discovery [12]. Besides, such data can contribute to the identification of more minor but homogeneous groups of patients who may benefit from new treatment opportunities [6].

Despite all the potential of RWE, they have limitations and suffer a few shortcomings. Contrary to well-defined protocols for data collection in clinical trials, RWE databases may not have consistent and adequate data quality, structure, and standardization. For instance, unstructured data such as medications and diagnoses are considered high uncertainty records, as the data documentation depends heavily on the hospital guidelines and physician or clinical staff situation during data documentation. In addition, absolute values of a diagnostic parameter may depend on the type of monitoring equipment and the respective calibration parameters, even for structured data. Such tremendous heterogeneity of data types requires sophisticated approaches to make these data usable and translatable. In practice, a parameter is only taken into assessments, when clinicians are convinced it is necessary, which lead to severe data gaps in analyses [10].

Dynamic observable data, such as vital signs (e.g., heart rate, blood pressure, body temperature) or routine lab parameters collection (e.g., blood and urine test results), are less sensitive to human errors, so they are advantageous for RWE data analysis. However, the monitoring rate of temporal data may vary from sparse to dense sampling frequency during hospitalization, which must be adequately tackled during analysis.

## 1.3 Biosignals

Biosignals are a branch of RWE time-series data types and have developed into a vital indicator for health diagnosis, successive treatment, and proactive health tracking.

They deliver data on physiological conditions that are indicative of an individual's health and comfort [13, 14, 15].

Biosignals are driven by the development of monitoring technologies, so-called wearables (portable devices) or implantable sensors. Such devices are an arising innovation in the healthcare system that enables remote and real-time monitoring, tracking, and analysis of human health states [16], including daily steps, heart rate, brain activity, and glucose level.

The concept of wearable devices was introduced in 1990 to empower patients to monitor their health conditions outside the clinic or hospital [15]. These clinical signals can eventually support disease prevention or be used as diagnostic tools for quick treatment and recovery from diseases or injuries [17]. For example, such biosignals have been extensively applied to various applications, including detecting cardiac arrhythmias, epileptic seizures, sleep phase evaluation, sentiment analysis, and blood pressure detection. Although data collected at clinics is always considered the gold standard compared to data monitored from wearable devices, these devices are now gaining popularity due to their capability to continuously monitor a small set of health-related signals in very dense sampling rates over very long periods. These monitored patient data can be shared with health professionals to establish a monitoring framework through the cloud.

Despite the fact that wearables are mainly used in fitness applications today, they are anticipated to be the first choice in the future for various medical applications. For example, The concept of clinical trials can be improved by monitoring patients continuously throughout the day rather than just during clinic visits. Incorporating wearable technology and remote monitoring methods in multi-institutional trials can expand the sample size and diversity of participants. Furthermore, the utilization of wearables to collect more detailed and specific health outcomes data can aid in demonstrating the clinical benefit of a new treatment, increasing the probability of proceeding to a large-scale randomized trial [18].

The wearable sensors integrated with smart devices offer continuous monitoring, wireless transmission, low memory and power consumption, cost-effectiveness and user-friendliness. In contrast, traditional data monitoring methods cannot be performed remotely, and data acquisition, transformation, and processing require a significant amount of time and effort [16]. Therefore, the data structures generated by wearables are very different from the telemetric RWE data structures discussed above. Wearables deliver long time series data with high sampling frequency and high availability, which leads to a low number of missing values compared to RWE

datasets. Data from wearables is consistently structured and clearly defined; however, in contrast to RWE data, wearables can only record a few signals that corresponding sensors can assess. Additional data types such as diets, activities, etc., have to be collected manually by patients and then added into the data, which is highly error-prone process, as patients might not record such data types consistently and accurately, particularly in elderly groups.

Another issue with wearables is that laypersons must operate and handle them, which in turn can induce systematic measurement errors if the sensors are not positioned correctly, and the displacement is not noticed. Although sensors used in wearables are primarily non-invasive, complex signal processing techniques are required to extract a high-quality signal. Therefore, sophisticated algorithms for pattern detection with a low signal-to-noise ratio are essential for a prospering health monitoring analysis. In spite of this fact, new developments in embedded sensors software showed promising results in real-time filtering of data [19, 20].

Today, wearables collect large datasets for each individual, which facilitates the development of personalized diagnostic tools. Nonetheless, identifying data patterns related to diseases, in any case, requires data from larger cohorts of patients, which are not yet widely available. Therefore, ML-based pattern identification is focused on patterns that can be easily identified by ML tools or that are at least qualitatively known from previous analysis, e.g., arrhythmia detection in ECG data [21]. However, this situation could change qualitatively if commercial wearable providers could access consumer data, creating databases of monitoring signals and accordingly opening up new prospects.

## 1.4 Time-Series Data Analysis

Time-series data is a sequence of observations collected over a period of time. We discussed in section 1.1, in the era of Big Data, as a result of the healthcare sector's transition from manual to automated data collection processes, there has been an explosion in massive and complex temporal clinical data generation. Clinical time series enclose a wide range of data types that can be captured in a variety of forms and structures, from electronic health records (EHRs) and clinical trials to biosignals collected by commercial monitoring wearables [22, 23].

Although collecting and storing such data types have become more effortless, their methodical use demands the development of specific algorithms and mathematical models. For instance, two typical applications are detecting hidden patterns of

recurring time series segments and identifying surprisingly irregular patterns. These patterns can be used, for example, to identify high-risk patients at an early stage or to predict the occurrence of sudden critical disease processes at an earlier phase compared to the state of the art of medical therapy protocols [24].

Because of the high heterogeneity, high noise level, and incompleteness of clinical time series, data preprocessing is the first step to identifying and handling these issues. Noise reduction and detection of unwanted outliers is the first stage of time series preprocessing. Noise reduction can be performed using various techniques, e.g., spectral filters, rule-based approaches, and autoregression [23]. Missing values are another difficulty typically experienced in acquiring sequential time series data. Missing observations can be caused by missing or incorrect data collection, lack of documentation, or failure of recording equipment. Missing data values can significantly affect the accuracy of further analyses. Therefore, it is essential to determine the underlying cause of missing data to select the appropriate method for handling missing values. There are two approaches to dealing with missing data: conventional methods such as ignore, delete, and mean/mode imputation on the one hand, and imputation methods such as interpolation or implementing the Gaussian process on the other [25, 26]. Time series data analysis presents a significant challenge due to its high dimensionality. Analyzing temporal data point by point is often inadequate when dealing with time series data. However, a favorable aspect of time series data is that consecutive values are often highly correlated, leading to significant redundancy [27]. To address this issue, feature extraction techniques should be employed to compress the time series by retaining only the important information and eliminating noise and correlations. By selecting the appropriate features, the analysis will be accelerated and produce more accurate results compared to analyzing the original data [27, 28]. In general, individuals use this method to develop frameworks for many time-series data applications, including disease classification and symptom prediction. This can significantly lower the amount of data required to develop models for tasks such as disease classification and symptom prediction. The vector space values of extracted features can be readily understood and interpreted because the technique explains the steps required to convert data into features. The following section explains feature engineering as the core concept of this thesis. The feature space data can be used later for different ML tasks. Two specific applications are pattern recognition and anomaly detection in time series. Time series data may vary from one patient to another, or time sequences may comprise different segments in the same patient. Comparing different time series or segments of a series and identifying

similar patterns in the data is called time series clustering. Time-series clustering can be used for patient stratification in large populations and describe the similarity between the patients of each cluster by means of the weight of extracted features [29]. Time-series segments clustering can be employed to identify and characterize different states in the dynamics of a biological system (e.g., in disease progression) and discover the relationships between the segments and phenotypes [23]. The use of classification learning algorithms can enable pattern recognition and prediction when labels are present in the data. Once the learning process is completed, the system can automatically classify new sequences into their respective classes when processing them. Anomaly detection refers to techniques used to detect, identify, and predict anomalous patterns and observations in data. Such anomalies may be unwanted errors that need to be removed from the data during the denoising process. However, anomalous patterns may also indicate the occurrence of biologically relevant events or transitions from one clinical condition to another. For example, a sudden shift from a healthy condition to a disease condition might be an anomaly in the data. The cardiac arrhythmia events are examples of events found as abnormal patterns in time series. Mathematical models that can detect and indicate transitions before these events can be operated as early warning signals (EWS)[24, 30].

## 1.5 Time-Series Feature Engineering

The ultimate aim of time series analysis tools is to process data efficiently within a reasonable time frame. Nevertheless, direct application of an analysis method to unprocessed time series, especially to high-resolution biosignals, might be too costly and computationally intensive, e.g., for rapid assessment of risks. In this context, representation, also called feature extraction, refers to the transformation of the time series into another data space formed from derived extracted features [23]. A good representation should have a lower computational cost, retain the elemental information of the original sequence, allow reconstruction of the data from the reduced representation with sound quality, and be insensitive to noise. Therefore, developing algorithms for extracting features from time-series data is a key area of research in time series analysis that can effectively generate information related to specific concerns [31]. There are two fundamental methods for extracting relevant feature descriptors from biosignals, including hand-crafted and semi/unsupervised learning features. The engineered descriptors exploit domain-expert knowledge to develop methods for deriving relevant features from data. On the other side, generic feature

extraction is a technique that employs numerous feature functions to extract significant descriptors from signal data. The integration of domain-expert knowledge into machine learning (ML) applications has been shown to be a valuable technique for improving the performance of the ML models [32]. In the context of time series feature engineering, domain knowledge can be used to identify relevant features and to guide the selection of appropriate feature extraction methods [33]. For example, a domain expert in the field of cardiology may be able to identify specific cardiac biosignals, such as heart rate variability, that are relevant to predicting cardiovascular disease [34]. Similarly, a domain expert in the field of neurology may be able to identify specific skeletal muscles signals, such as electromyography (EMG) patterns, that are relevant to predicting neurological disorders [35]. Using domain knowledge in clinical time series and biosignals feature engineering can also help to improve the interpretability of the models. By understanding the features and the feature extraction methods used, domain experts can better interpret the models and make more informed decisions about patient care. Additionally, application-based features, such as those derived from domain-specific knowledge, can capture essential information that is not captured by generic features. This can lead to improved performance in tasks such as diagnosis, prognosis, and monitoring of disease progression [36]. Incorporating such features can improve the performance of the models and make them more robust to outliers and noise in the data. The advantages of using domain-expert knowledge in time series feature engineering motivated us to explore the potential benefits of utilizing and adapting such features from other scientific fields into healthcare data analysis. We showed how a latent temporal feature could be used to identify critical transitions in biological systems [37]. Our findings, which have been presented in this thesis too (See Chapter. 8), demonstrated the potential of such features to gain a deeper understanding of the underlying physiological processes. Encouraged by these results, we have also investigated the potential of domain-knowledge features in other healthcare temporal data analysis applications using a robust time series analysis tool.

## 1.6 Time-Series Analysis Tools

To achieve a concise time-series data analysis with reliable outcomes, one should perform successive steps, including data wrangling, feature extraction, and modeling. In this sense, depending on the application, design, and implementation of the described steps can be elaborative tasks and require comprehensive domain knowledge as well

as proficient programming skills. Therefore, in recent years, we have witnessed the development of several platforms that tried to construct and facilitate the analysis workflow of all the temporal data analysis steps or at least one step. Such platforms aim to provide software and tools which provides scientists to perform all or part of their time-series analysis in an established and standardized way. As an example, Christ et al.[38] developed a Python-based package to support a fast and automated time series feature extraction and selection. *Tsfresh* provides an enormous number of built-in features that can be picked by the user, where its feature selection algorithm can filter the extracted features. In the context of prediction schemes, imbalanced data and a heterogeneous set of features can lead to biased and unreliable outcomes. To handle these issues, Lee et al.[39] proposed a computationally inexpensive and robust predictive tool named *SoFt*, which applies a selected learning model and utilizes a filtering-based feature selection algorithm for addressing imbalanced datasets. Another Python library, called *Tslearn* [40], tries to cover all the time-series analysis steps and deliver preprocessing tools, feature extractors, and ML algorithms to classify and cluster temporal data. *Tslearn* utilizes time-series distinctive metrics (e.g., DWT) to quantify the similarity between univariate or multivariate time series. The *pyts* Python package [41] is another example of open-source software developed for time series classification, offering implementations of several classification algorithms. The described packages are only some examples of published toolboxes, and there exist numerous other libraries with more or less similar capabilities and functionality (e.g., *Seglearn* [42], *Cesium* [43], *Featuretools* [44], *Pastas* [45], *sktime* [46]).

## 1.7 Objective of the work

In section 1.5, the significance of domain knowledge in the extraction of informative features from time series data and its application in machine learning tasks was discussed. The process of extracting information from raw dynamic data and making informed decisions requires multiple steps, which may include complex implementation and programming.

Due to the rapid accumulation and storage of time-series data over the past few years, there has been a growing demand for tools and software to simplify the process mentioned above. As noted in the previous section, various libraries and packages have been developed to address this demand, covering different aspects of time series analysis, from data preprocessing and feature engineering to operating statistical and artificial intelligence models.

The analytical part of the time series analysis, which includes data mining and the development of predictive models, is highly dependent on the specific application and requires specialized tools. Nevertheless, it is crucial to acknowledge that data preparation and feature extraction are fundamental aspects of any time series analysis implementation. These stages play an important role in pattern recognition during the analysis phase, as the quality of data preparation can significantly impact the results of machine learning tasks [47].

Therefore, it is necessary and feasible to develop a generic pipeline that produces ready-to-use data for analysis, as this will ensure consistent and accurate results in the time series analysis. The importance of a well-structured data preparation stage cannot be overstated, as it lays the foundation for a successful time series analysis implementation.

Section 1.6 highlighted libraries that were developed for this purpose. However, their limitations restrict their utilization in every time series application or necessitate additional, sophisticated coding. These tools concentrate on data preprocessing and representation, whereas one of the most significant issues, particularly with large time-series data, is the importation into the workflow environment. Loading and retrieving thousands to millions of data files or importing dense and lengthy signals can impact the overall pipeline efficiency. Another issue is that packages with data feature extraction functionality limit users by providing pre-defined features. A major challenge in healthcare data analytics is the high heterogeneity of data assessment protocols, which can cause problems when built-in functions do not meet the specific application requirements.

The present study aims to investigate the domain knowledge features present in various types of clinical time series data by implementing a standardized workflow. A generic workflow for time series data analysis was developed, and its technical capabilities and applicability were explored. The central objective of this thesis is to establish a standardized framework for time series data analysis, incorporating the necessary steps of temporal data importation, windowing (if necessary), preprocessing (such as outlier removal, denoising, and missing value imputation), and feature extraction for subsequent analysis. This framework has been implemented as a Python package with specified attributes:

- Robust and easy-to-use with reliable machinery that aims to facilitate exploring and study of any temporal data structure from sparse longitudinal data to continuous high-resolution signals for scientists as well as non-experts (e.g., clinicians).

- Modular design that allows users to select proper tools based on the application.
- Having ready-to-use time-domain, frequency-domain, and nonlinear feature functions can be utilized for data representation.
- User-defined functions can be easily embedded into the workflow for data cleaning or feature engineering.
- Efficient and fast implementation of the pipeline multi-processing mode, that makes it a suitable tool for cloud-based computation.

We demonstrated these functionalities through three showcases, covering high-resolution biosignals classification, statistical analysis and classification of lengthy multivariate clinical temporal data with different sampling rates, and critical transition detection in complex heterogeneous multivariate longitudinal data.

## 1.8 Outline

This thesis is structured as follows:

Chapter 2 provides an in-depth analysis of the structure of the developed Python package, which consists of three main components - data importation, data preprocessing, and feature extraction. The design and architecture of the package are discussed in detail, focusing on the specific purposes served by each component.

Chapter 3 focuses on the data preprocessing methods integrated into the pipeline of the package. It provides a comprehensive description of the various methods used to clean, normalize, and prepare raw data for further analysis, as well as a discussion of the rationale behind the use of each method.

Chapter 4 provides an overview of the built-in features incorporated into the developed package, including time domain, frequency domain, and wavelet features. The methods used to extract these features and their utility in the analysis of clinical time series data are described in detail.

Chapter 5 provides an overview of the Statistical and Machine Learning Models used in the thesis.

Chapters 6, 7, and 8 present case studies that illustrate the application of features in clinical time series data analysis. Chapter 6 demonstrates the use of time and frequency domain features to classify high-resolution ECG signals. Chapter 7 explores the relationship between Bradyarrhythmias and extreme glucose levels in patients with type-2 diabetes and the development of a predictive model for bradycardia events. Chapter 8 proposes a temporal feature based on surprise loss to uncover data-driven indicators of transitions in a multivariate time series dataset of septic

shock and non-sepsis patient cohorts.

Finally, Chapter 9 summarizes the key findings of the thesis and provides recommendations for future research in the field of clinical time series data analysis. The chapter reflects on the results obtained in the thesis and identifies opportunities for future work in this field, including areas where further research is needed.

In conclusion, this thesis comprehensively examines the structure, methods, and applications of incorporating feature engineering in the analysis of clinical time series data. The case studies presented in Chapters 6, 7, and 8 highlight the utility of the features incorporated into a workflow and demonstrate the potential of this tool in the analysis of clinical time series data.

# Concept of the software Platform

## 2.1 Structure

The developed Python package, called Continuous Monitoring Data Analysis (CMDA), contains three major modules that can be employed a workflow delivering the raw data to ready-to-use data. Figure 2-1 shows the schematic of the workflow. A technical documentation of the package and implementation in Python are provided in appendices. [A](#) and [B](#).

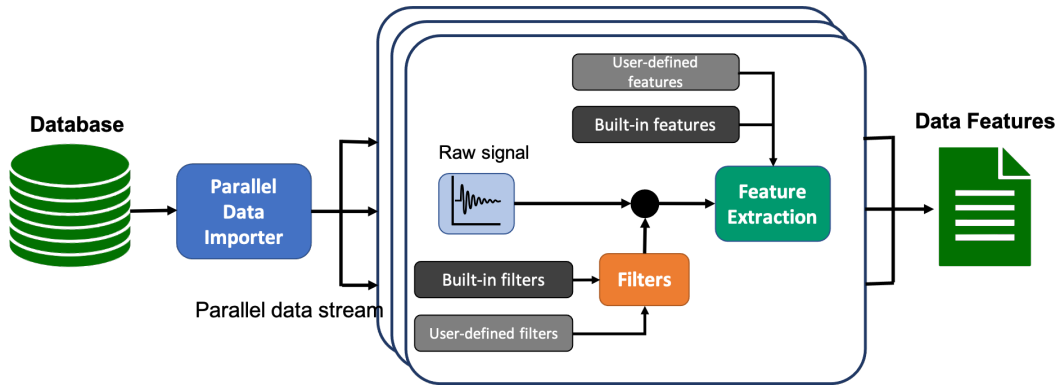


Figure 2-1: **CMDA** workflow; The structure contains three components: 1)Data importer, where multiple records and files can be imported to the workflow from various sources, 2) Preprocessing unit, where built-in functions or user-defined functions can be employed for outlier removal, denoising and missing value imputation, 3) Feature extraction unit, where numerous built-in features or user-defined features can be used to extract features from the imported records.

## 2.2 Data Importer Module

Data importer, the first component in the pipeline, offers tools for loading a large number of CSV files from a server or local disk into the python coding environment. A comma-separated values (CSV) file is a comma-separated values file that allows

data to be stored in a tabular format, making it a standard and popular format for saving time series data. The importer module also supports importing data files directly from Physionet databases. Physionet, funded by the National Institutes of Health, is a vast archive of hundreds of digital recordings of physiologic signals and related data for medical research [48] (See sect. A.1). By utilizing importer functions, computation time can be reduced during the loading process by processing multiple data files at once. Additionally, for extremely lengthy time series, a rolling window importer function can be used to perform time binning and distribute the importing process of each window across multiple nodes in parallel. A detailed explanation of windowing can be found in Section 3.3.

## 2.3 Preprocessing Module

The Preprocessing module is the next component in the workflow, which includes a variety of functions for tasks such as removing outliers, reducing noise (See sect. A.2), and filling in missing values. These processes are detailed in chapter 3. The selected functions are added to a preprocessing object and applied to the data in sequence. Additionally, the module allows users to easily create their own preprocessing functions and add them to the preprocessing object using an adding function.

## 2.4 Feature Extraction Module

The final step in the pipeline is feature extraction. We developed a component dedicated to this task which provides several ready-to-use built-in time domain, spectral domain, nonlinear, and wavelet features that can be selected and added to a feature object instance (See Chapter. 4). Running the feature object returns the calculated assigned features of a given time series. Similar to the preprocessing module, user-defined functions can be defined and merged into the feature object instance.

## 2.5 Pipeline

Given multivariate time series records, users have the flexibility to create an independent set of preprocessing functions, time binning, and features for each variable. This, for example, allows for integrating multiple variables with different sampling rates by defining a constant window length across all the variables in a segmentation

---

process. The entire process, from importing data to preprocessing and feature extraction, can be streamlined into a pipeline that can be executed in a parallel manner. However, it is important to note that each module can also be used separately in other projects, depending on the specific application. The final output of the pipeline will be extracted features from multiple records in either a Pandas DataFrame format or a Python Dictionary format. These features can then be saved as a JSON file and used in other programming languages or platforms.



# Preprocessing Methods

## 3.1 Normalization

Data normalization, also known as feature scaling, is a crucial step in the process of preparing data for analysis. It involves transforming the independent variables or features to a similar scale and range. Depending on the algorithm used for processing the data, data normalization may be essential as it can significantly impact the analysis outcome.

For example, in distance-based algorithms such as KNN, K-means, and SVM, the similarity measure is based on the pairwise distance of the samples. Without scaling, features with a higher order of magnitude will determine the weights and the final result of the ML task, potentially leading to inaccurate results.

Additionally, models that use gradient descent as the optimizer, such as Linear models and neural networks, are also heavily impacted by data normalization. As gradient descent updates the step size in each convergence step, the difference in the ranges of features will cause different step sizes for each feature. This can lead to a biased step size and prevent a smooth convergence. Therefore, scaling the data is required to avoid these issues. On the other hand, some other methods, such as tree-based methods like *Random Forests*, are not affected by the variables' range, and feature scaling is unnecessary. However, in post-processing and visualization, data normalization may still be beneficial in order to make the data more easily interpretable.

In this thesis, we employed Z-score as a commonly utilized method of standardization. This technique is preferred as it does not have a predefined limit and does not alter the data outliers. Z-score can be formulated as follows:

$$x' = \frac{x - \bar{x}}{\sigma} \tag{3.1}$$

Where  $x$  and  $X$  are the scaled and original feature vectors, respectively,  $X$  is the arithmetic mean of that feature vector, and  $s$  is its standard deviation.

## 3.2 Missing Value Imputation

Missing values are a common issue that arises during sequential data collection. Various factors, such as errors during data collection, lack of documentation, or malfunctioning recording devices, can cause the absence of observations. The presence of missing values in data can greatly impact the accuracy of any subsequent analyses. As a result, it is crucial to identify the cause of missing data in order to select the appropriate method for handling the absent values. This can be particularly challenging when the absence of data is dependent on the variable itself. In this thesis, we utilized two approaches to address incompleteness in time series data. Depending on the specific case, we employed conventional methods such as ignoring, deletion and mean imputation or interpolation methods. Interpolation is a technique that uses a function to create new data points or fill in missing points within the existing data points of a time series. The function used for interpolation can be a polynomial, such as a linear or cubic function, or a modified polynomial function, like the *Steinman* function [49]. Implementing the *Steinman* method has similar performance compared to polynomial interpolation. However, *Steinman*'s shows better reliability if the variable demonstrates sharp changes in slope [50].

## 3.3 Windowing

In many time-series applications, it is necessary to divide the original time series into smaller segments or sections. These segments can then be utilized in various ways, such as being used directly in temporal models such as autoregression models and recurrent neural networks, or in the feature extraction process.

In the feature extraction process, each segment is transformed into a set of features that can be used for further analysis. This technique is commonly used to reduce the complexity of high-resolution signals or to handle time series data that has been sampled at uneven frequencies.

It is important to note that feature extraction is a critical step in many time-series applications and a detailed explanation of this process can be found in chapter 4.

For a given time series data  $X = (X_1, X_2, \dots, X_N)$ , with a corresponding time indices  $t = (t_1, t_2, \dots, t_N)$ , where  $N$  is the total number of observations, we can define

a sliding window with time indices  $t'$  as follows:

$$t_{first} + m = t_{last} \quad (3.2)$$

$$t' = \{t_i\} \quad , i \in (1, \dots, N) \quad , \text{ where } t_{first} < t_i < t_{last} \quad (3.3)$$

Where  $t_{first}$  and  $t_{last}$  are the starting and end time of the window and  $m$  denotes the length of the window. Hence, with a step size equal to  $s$ , we can formulate the segmented time-series  $X'$ :

$$X'_{t'} = [X_{0:m}, X_{s:m+s}, X_{2s:m+2s}, \dots, X_{t_N-s:t_N}] \quad (3.4)$$

where each row of  $X'$  denotes an  $X$  segment within the window  $t'$ .



# Features

As described in section 1.5, feature extraction or engineering is the process of identifying and extracting important characteristics of raw data signals and understanding their underlying behavior. This can be done by transforming the input data into a set of features that correspond to specific patterns or behaviors within the signal. The goal of feature extraction is to create a concise and optimal representation of the data, which can be accomplished by reducing the dimensionality of the input signal or compressing its data. This reduction in resources required to evaluate the signal can result in a more efficient and accurate analysis system. In addition, feature extraction can also help identify and remove expendable elements from a raw signal, leaving a minimal set of distinctive data that can accurately represent the input signal and aid in developing robust pattern analysis systems. In the following sections, we will present the specific features that we used in this thesis.

## 4.1 Time-Domain Features

In the field of time-series data analysis, time-domain features are used to describe signals over a specific time span. These features provide a way to understand the characteristics and patterns of a signal as it varies over time. The following is a list of some of the most widely used and well-known time-domain features in time-series data analysis:

### 4.1.1 Main Statistical Features

The main statistical features that can be used as descriptive measures of time series data are: Mean, Median, Variance, Standard Deviation, Median Absolute Deviation, Maximum, Minimum, Quantiles, Skewness, and Kurtosis.

### 4.1.2 Peak to Peak distance

The peak to peak distance (P2P) is the absolute difference between a signal's extreme values.

$$P2P(x) = |Max(x) - min(x)| \quad (4.1)$$

### 4.1.3 Root Mean Square

The root mean square (RMS) of a signal is the square root of the average signal square values and assesses the power of a signal. It can be computed as follows:

$$RMS(x) = \sqrt{\frac{1}{T} \sum_{t=1}^T x_t^2} \quad (4.2)$$

Where  $T$  is the length of  $x$ .

### 4.1.4 Zero Crossing Rate

The zero-crossing rate (ZCR) is a measure of the rate at which a signal changes direction, or in other words, it is a measure of the number of times a signal's amplitude changes from positive to negative or from negative to positive values. This measure can be used to determine the noisiness of a signal, as signals with a high ZCR are typically considered to be more noisy. The ZCR of a centered signal specifically measures the rate of sign shifts around the mean value of the signal. To compute the ZCR, one can use the following method:

$$ZCR(x) = \frac{1}{2T} \sum_{t=2}^T |sgn(x_t) - sgn(x_{t-1})| \quad (4.3)$$

Where  $T$  is the length of  $x$ , and  $sgn$  is the sign function.

### 4.1.5 Auto-correlation

The autocorrelation coefficient (ACN) is the correlation coefficient of a signal ( $x$ ) with a lagged copy of itself. Autocorrelation can be a descriptor of periodicity or

randomness in the data. It can be defined as:

$$ACN(x, k) = \frac{\sum_{t=k+1}^T (x_t - \bar{x})(x_{t-k} - \bar{x})}{\sum_{t=1}^T (x_t - \bar{x})^2} \quad (4.4)$$

Where  $T$  is the length of  $x$ ,  $k$  is the lag value, and  $\bar{x}$  is the mean value of  $x$ .

## 4.2 Spectral Features

Frequency-domain features are signal characteristics in the frequency domain, which are extracted from the Fourier transform of the signal.

### 4.2.1 Power Spectrum Energy

The power spectrum energy is the sum of the power spectrum, also known as the zero spectral moment ( $SM_0$ ) [35]. It can be computed as follows:

$$PSE(P) = \sum_{j=1}^{j=M} P_j \quad (4.5)$$

where  $P_j$  is the power spectrum at the frequency bin  $j$ , and  $M$  is the length of frequency bin.

### 4.2.2 Mean Frequency

The mean frequency, also known as the mean spectral frequency or mean power frequency, is a measure of the center of distribution of power across different frequencies in a spectrum. It is calculated by taking the average of all the frequencies present in the spectrum, and is used to understand the distribution of energy across different frequencies. The mean frequency is an important metric in signal processing, as it can provide insight into the characteristics of a signal and help in identifying patterns

or anomalies in the data.[51]. The definition is as follows:

$$MNF(f, P) = \frac{\sum_{j=1}^M f_j P_j}{\sum_{j=1}^M P_j} \quad (4.6)$$

where  $f_j$  is the frequency value of the power spectrum at the frequency bin  $j$ ,  $P_j$  is the power spectrum at the frequency bin  $j$ , and  $M$  is the length of frequency bin.

### 4.2.3 Median Frequency

The median frequency is the frequency at which the power spectrum is divided into two parts with equal powers[51]:

$$\sum_{j=1}^{MDF} P_j = \sum_{j=MDF}^M P_j \quad (4.7)$$

where  $MDF$  is the median frequency,  $P_j$  is the power spectrum at the frequency bin  $j$ , and  $M$  is the length of frequency bin.

The estimation of MNF and MDF is almost identical. However, the MDF is less impacted by random noise.

### 4.2.4 Variance of Central Frequency

The variance of the power spectrum is the frequency dispersion from the spectrum mean frequency[51]:

$$VCF(f, P) = \frac{\sum_{j=1}^M P_j (f_j - MNF)^2}{\sum_{j=1}^M P_j} \quad (4.8)$$

where  $MNF$  (See section 4.2.2) is the mean frequency of the power spectrum,  $f_j$  is the frequency value of the power spectrum at the frequency bin  $j$ ,  $P_j$  is the power spectrum at the frequency bin  $j$ , and  $M$  is the length of frequency bin.

### 4.2.5 Peak Frequency

The frequency at which the maximum of power spectrum occurs:

$$f_{\max(P_j)} \quad j = 1, \dots, M \quad (4.9)$$

where  $f$  is the frequency values of the power spectrum,  $P_j$  is the power spectrum at the frequency bin  $j$ , and  $M$  is the length of frequency bin.

### 4.2.6 Power Spectrum Ratio

The power spectrum ratio, also called peak-to-average power ratio, is the energy around a predefined vicinity of the power spectrum maximum value divided by the whole energy of the power spectrum. It can be computed as follows:

$$PSR(f_0, P, n) = \frac{\sum_{j=f_0-n}^{j=f_0+n} P_j}{\sum_{j=1}^M P_j} \quad (4.10)$$

where  $f_0$  is the peak frequency (See section 4.2.5),  $n$  is the integral limit,  $P_j$  is the power spectrum at the frequency bin  $j$ , and  $M$  is the length of frequency bin.

### 4.2.7 Frequency Band Features

The frequency band features refer to a set of characteristics that are calculated within a specific range of frequencies [52]. One example of a frequency band feature is the normalized band energy, which can be calculated by normalizing the energy within a specific bandwidth in relation to the total energy within that bandwidth. This allows for a consistent comparison of energy levels across different frequency bands:

$$FPR(f_{low}, f_{high}, P) = \frac{\sum_{j=f_{low}}^{j=f_{high}} P_j}{\sum_{j=1}^M P_j} \quad (4.11)$$

where  $f_{low}$  and  $f_{high}$  are the low and high frequency band,  $P_j$  is the power spectrum at the frequency bin  $j$ , and  $M$  is the length of frequency bin. Other examples of band

features are band mean frequency (4.2.2), band median frequency(4.2.3), band peak frequency (4.2.5), band spectral entropy (4.3.3), and frequency variance (4.2.4).

## 4.3 Entropy Features

In information theory, Shannon entropy measures information and randomness in a random variable. We can formulate the entropy of the distribution  $X = \{x_1, x_2, x_2, \dots, x_n\}$  as follows:

$$H_a(P) = - \sum_{i=1}^n P(x_i) \cdot \log_a P(x_i), \quad a > 1 \quad (4.12)$$

where  $P$  is the probability mass function and  $a$  is the base of the logarithm. The potential usefulness of Shannon entropy in determining the complexity in dynamic systems led to the definition of various forms of entropy that we can derive as signal features.

### 4.3.1 Sample Entropy

Sample entropy is a measure of the complexity and irregularity of a time series signal. It is a derivation of Shannon entropy, which compares the similarity of patterns within a signal over different time scales. Sample entropy is calculated by comparing the probability of finding a similar pattern within a given time series signal. The lower the sample entropy, the more regular and predictable the signal is. Conversely, a higher sample entropy indicates a more complex and irregular signal. Sample entropy is commonly used in the fields of signal processing and biomedical engineering to analyze various types of data, including physiological signals, time-series data, and nonlinear systems [53].

To calculate sample entropy, we map signal  $x$  with the length of  $N$  into an embedding space of dimension  $m$  and  $m + 1$ , respectively:

$$x_m(i) = \{x_i, x_{i+1}, x_{i+2}, \dots, x_{i+m-1}\} \quad (4.13)$$

$$x_{m+1}(i) = \{x_i, x_{i+1}, x_{i+2}, \dots, x_{i+m}\} \quad (4.14)$$

Sample Entropy, then, can be defined as follows[54]:

$$SampEn(m, r, N) = -\log \left( \frac{\sum_{i=1}^{N-m} \sum_{\substack{j=1 \\ j \neq i}}^{N-m} [\text{number of times } d(x_{m+1}(i), x_{m+1}(j)) < r]}{\sum_{i=1}^{N-m} \sum_{\substack{j=1 \\ j \neq i}}^{N-m} [\text{number of times } d(x_m(i), x_m(j)) < r]} \right) \quad (4.15)$$

where  $r$  is a distance threshold, and  $d$  is a distance function. In other words, sample entropy calculates the probability that identical templates (when their distance is smaller than a threshold  $r$ ) in an embedding space  $m$  remains similar in higher dimensions, too.

As a rule of thumb,  $m$  to be 2 and  $r$  to be 0.2 of the standard deviation of  $x$ .  $d$  can be any arbitrary distance function, but usually, Chebyshev distance is used.

### 4.3.2 Permutation Entropy

Permutation entropy is another derivation of Shannon entropy, used to assess the complexity of time series data, proposed by Bandt[55]. Permutation entropy is a mathematical method used to quantify the complexity and predictability of a time series. It is based on the idea that a highly predictable time series can be described by a small set of patterns, while a more complex and unpredictable time series will have a larger set of patterns.

The method involves partitioning the time series into a series of overlapping windows, and then creating a probability distribution of the order of the values within each window. This probability distribution is then used to calculate the permutation entropy, which is a measure of the complexity and predictability of the time series.

To calculate the permutation entropy, similar to sample entropy, we map the signal  $x$  with the length of  $N$  to an embedding space with dimension  $m$  with delay  $\tau$ :

$$x_{m+1}(i) = \{x_i, x_{i+1}, x_{i+2}, \dots, x_{i+m}\} \quad (4.16)$$

Next, we map each  $x_m(i)$  to a symbolized ordinal value  $Y_m(i)$  from distribution  $\Pi$ , whose elements  $\pi_j$  are all possible  $m!$  permutations of ordinal pattern, where  $j : 1, \dots, m!$ , and calculate the relative probability of all  $Y_m$  elements,  $P(\pi_j)$ .

Using relative probability  $P(\pi_j)$  of ordinal patterns and Shannon entropy definition,

we can determine permutation entropy as follows:

$$PE(m) = - \sum_{i=1}^{m!} P(\pi_i) \cdot \log(P(\pi_i)) \quad (4.17)$$

Normalized permutation entropy, which is restricted between 0 and 1, can be defined as:

$$PE_{norm}(m) = - \frac{1}{\log_2 m!} \sum_{i=1}^{m!} P(\pi_i) \cdot \log(P(\pi_i)) \quad (4.18)$$

To give an example, assume  $x = (12, 10, 8, 14, 9, 8, 6)$ . all the possible ordinal templates of a 3-dimensional embedding space are:

$$\Pi_3 = \{\pi_1 = (0, 1, 2), \pi_2 = (0, 2, 1), \pi_3 = (1, 0, 2), \pi_4 = (1, 2, 0), \pi_5 = (2, 0, 1), \pi_6 = (2, 1, 0)\}$$

Therefore, the transformation of  $x$  into the ordinal embedded space with  $m=1$  and  $t=1$  is:

$$\begin{aligned} x_m &= \{(12, 10, 8), (10, 8, 14), (8, 14, 9), (14, 9, 8), (9, 8, 6)\} \\ Y_m &= \{(2, 1, 0), (1, 0, 2), (0, 2, 1), (2, 1, 0), (2, 1, 0)\} \end{aligned}$$

Consequently, the relative probability distribution of  $Y_m$  elements and corresponding permutation entropy are:

$$\begin{aligned} &\{P(\pi_1) = 0, P(\pi_2) = \frac{1}{5}, P(\pi_3) = \frac{1}{5}, P(\pi_4) = 0, P(\pi_5) = 0, P(\pi_6) = \frac{3}{5}\} \\ &-\frac{1}{\log_2 3!} \sum_{i=1}^6 P(\pi_i) \cdot \log(P(\pi_i)) = -\frac{1}{\log_2 3!} \left( -\frac{1}{5} \log_2 \frac{1}{5} - \frac{1}{5} \log_2 \frac{1}{5} - \frac{3}{5} \log_2 \frac{3}{5} \right) = 0.53 \end{aligned}$$

### 4.3.3 Spectral Entropy

The spectral entropy, based on the Shannon entropy, is a measure of the randomness or disorder within a signal's frequency spectrum. It is often used in signal processing and pattern recognition to analyze the complexity of a signal.

It can be calculated as follows[56]:

$$SpEn = - \sum_{j=1}^{j=M} \overline{P_j} \cdot \log_2 \overline{P_j} \quad (4.19)$$

where  $\overline{P_j}$  is the normalized power spectrum at the frequency bin  $j$ , and  $M$  is the length of frequency bin.  $\overline{P_j}$  can be calculated as follows:

$$\overline{P_j} = \frac{P_j}{\sum_{j=1}^M P_j} \quad (4.20)$$

where  $P_j$  is the power spectrum at the frequency bin  $j$ .

A higher spectral entropy value indicates a greater degree of randomness or disorder within the signal's frequency spectrum.

## 4.4 Wavelet Features

### 4.4.1 Discrete Wavelet Transform

The wavelet transform (WT) is a mathematical technique that is used to analyze signals in both the time and frequency domains simultaneously. This method of signal processing is based on the concept of wavelets, which are a set of basis functions that can be used to represent a signal in a different, more informative way. The WT utilizes these wavelets to project a temporal signal onto a new set of coordinates that are able to identify both the time and frequency characteristics of the signal. This is achieved by convolving the signal with different scaled and translated versions of the wavelet function, resulting in a set of coefficients that represent the signal in the wavelet domain. This method of signal processing was specifically designed to handle non-stationary signals, which are signals that vary over time and do not have constant statistical properties.

We can define the continuous wavelet transformation of signal  $x(t)$  as follows[57]:

$$C(a, b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} \psi\left(\frac{t-b}{a}\right) x(t) dt \quad (4.21)$$

where  $b$  is the translation parameter,  $a$  denotes the scale factor, and  $\psi$  is the

wavelet function.

Haar, Daubechies, Mexican hat, Morlet, and Meyer are some of the most popular wavelets that are commonly used in signal processing.

The continuous wavelet transform has redundancy disadvantage due to the nature of the wavelet transform. It suffers impractically, too, as both transform parameters are continuous. These two drawbacks can be overcome by using discrete wavelet transforms (DWT).

DWT is a method of signal decomposition that involves a series of band-pass filtering and down-sampling steps. This approach allows for the quick and efficient computation of wavelet transforms, and it delivers both high-frequency and low-frequency components of the signal. Overall, DWT is a more efficient and practical alternative to the continuous wavelet transform, and it can be used to overcome the redundancy and impracticality issues associated with the latter.

We can define DWT as follows[58]:

$$W_\phi[j_0, k] = \frac{1}{\sqrt{M}} \sum_n x[n] \phi_{j_0, k}[n] \quad (4.22)$$

$$W_\psi[j, k] = \frac{1}{\sqrt{M}} \sum_n x[n] \psi_{j, k}[n] \quad \text{for } j \geq j_0 \quad (4.23)$$

where  $n = 0, 1, 2, \dots, M-1$ ,  $j = 0, 1, 2, \dots, J-1$ ,  $k = 0, 1, 2, \dots, 2^j-1$ .  $M$  indicates the number of samples. This number is selected to be  $M = 2^J$ , where  $J$  denotes the number of transform levels. The basis functions  $\phi[j, k]$  and  $\psi[j, k]$  are defined as:

$$\phi_{j, k}[n] = 2^{j/2} \phi[2^j n - k] \quad (4.24)$$

$$\psi_{j, k}[n] = 2^{j/2} \psi[2^j n - k] \quad (4.25)$$

, where  $\phi[n]$  is the scaling function and  $\psi[n]$  is the wavelet function. The transformation can be explained in cascade form, where the scaling and wavelet functions can be related to a low pass filter  $G$  and a high pass filter  $H$ . The high-pass filter yields detailed information;  $C_d[n]$ , while the low-pass filter constructs coarse approximations,  $C_a[n]$ . Since the half-band filters produce signals covering only half the frequency band, half the samples are removed according to Nyquist's rule, which leads to decimation by a factor at each level. The filtering and subsampling process is resumed until reaching the selected level, and the elements of each scale output are the coefficients of the scale. Figure X shows the cascade tree, also known as the filter banks

| Coefficient | Level | Frequencies         | Samples |
|-------------|-------|---------------------|---------|
| $C_A$       | 4     | 0 to $f_n/16$       | 8       |
| $C_D$       | 4     | $f_n/16$ to $f_n/8$ | 8       |
| $C_D$       | 3     | $f_n/8$ to $f_n/4$  | 16      |
| $C_D$       | 2     | $f_n/4$ to $f_n/2$  | 32      |
| $C_D$       | 1     | $f_n/2$ to $f_n$    | 64      |

Table 4.1: A sample signal with 128 samples, frequency range 0 to  $f_n$  and 4 levels of decomposition and 5 output scales

of the DWT. For example, applying DWT on a signal with 128 samples, frequency range 0 to  $f_n$ , and four levels of decomposition results in 5 scales (See table 4.1).

Since DWT coefficients contain simultaneous localized time-frequency information of a signal, they can be used for signal analysis. We can use these coefficients directly in analysis or extract a group of features with the application's relevancy from them.

#### 4.4.2 SWT

In the preceding section, we discussed the process of decimation that occurs at each level of the discrete wavelet transform (DWT). As a result of this decimation, DWT is not a time-invariant transformation, meaning that it does not maintain the same properties when applied to the same signal at different times. As an alternative to DWT, the stationary wavelet transform (SWT) was introduced by Holschneider et al. [59]. This transform restores translation invariance in DWT by replacing the downsampling scheme with an upsampling filter before convolution.

This upsampling process, which is performed by a factor of 2, leads to detail and approximation output coefficients at each level that are equal to the length of the original signal. This redundancy in the SWT procedure means that it requires a signal with a length of  $2L$  when implemented with  $L$  levels. The filter bank of the SWT process is illustrated in figure X. Like DWT, a variety of features can be extracted from the SWT coefficients. However, the inherent nature of SWT in preserving the signal samples size in the transformation allows for more accurate extraction of spectral and entropy features at all levels. This is because the information from all signal samples is preserved, rather than losing information through decimation in DWT.

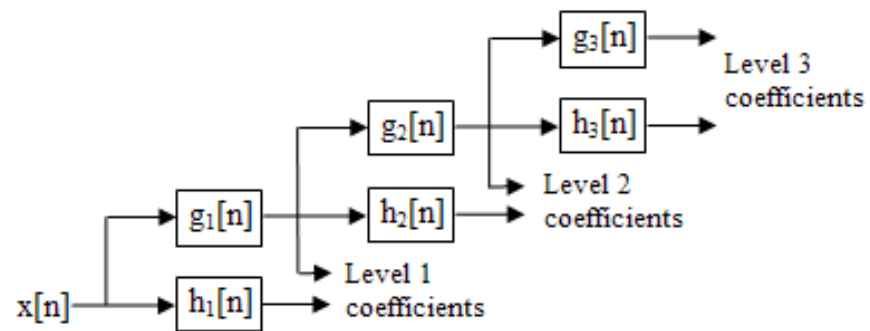


Figure 4-1: Stationary wavelet transform filter bank with 3 levels decomposition

# Statistical and Machine Learning Models

## 5.1 Statistical Tests

We perform a statistical significance test to estimate the probability that the difference between two sample data is merely due to the chance. The following sections describe two tests (Wilcoxon rank-sum test and Fischer's exact test) that we employed in this thesis.

### 5.1.1 Wilcoxon Ranksum Test

Wilcoxon rank-sum test, also known as the Mann-Whitney U test, is a non-parametric statistical test used as an alternative to a T-test [60] for two independent sample groups, introduced by Frank Wilcoxon [61]. A T-test tests the similarity of the mean of two independent samples, with the underlying assumption, that samples are from a gaussian distribution or very large in size to satisfy the central limit theorem. In case of small sample groups or unclear data distribution, the Wilcoxon rank-sum test can be used to test the difference between two groups if the data are at least ordinal [62].

The rank-sum test is used to determine the likelihood that two samples come from the same population. The test is based on the null hypothesis, which states that the two sample groups have the same distribution with the same median. The test calculates a p-value, which is an estimate of the probability that the samples come from the same population. A low p-value indicates that there is a low probability that the samples come from the same population, and thus, the null hypothesis is rejected. The alternative hypothesis, in this case, is that the samples come from different populations. In other words, a low p-value suggests that there is a significant difference

between the two sample groups, and they likely come from different populations.

Although calculating the test statistic is very straightforward, it needs many lines of text to present here. For a detailed formulation of this technique, please read [63].

### 5.1.2 Fisher's Exact Test

Fisher's exact test, introduced by Ronald Fisher [64], is a statistical significance test based on the hypergeometric distribution. This test examines the relationship between two categorical variables using a contingency table analysis. With the assumption of samples independency, we can define a p-value given a contingency table with two rows and two columns as follows:

$$p = \frac{\binom{a+b}{a} \binom{c+d}{c}}{\binom{n}{a+c}} = \frac{\binom{a+b}{b} \binom{c+d}{d}}{\binom{n}{b+d}} = \frac{(a+b)!(c+d)!(a+c)!(b+d)!}{a!b!c!d!n!} \quad (5.1)$$

where a,b,c, and d are the values in a 2-by-2 contingency table (See Table.5.1), and n is the total number of samples ( $n = a+b+c+d$ ). The p-value provides the probability

| Category | Sample Group |   |
|----------|--------------|---|
|          | X            | Y |
| A        | a            | c |
| B        | b            | d |

Table 5.1: A 2-by-2 contingency table, where X and Y are sample groups, and A and B are the categories

of the hypothesis that there is no relationship between the two categorical variables (null hypothesis). A significantly low p-value rejects the null hypothesis, indicating a relationship between the two categorical variables (alternative hypothesis).

## 5.2 Classification

Classification is a specific type of supervised learning, which is a subset of the broader field of machine learning. It involves the use of mathematical algorithms to create a model that can predict the output, or class, of a given set of input data. The input data is typically represented by variables  $X$ , while the output variable is known as  $y$ .

When the output variable is categorical, meaning it is a label or a class, the process of creating this model is referred to as classification. In this thesis, we utilized three different classification models described in the following sections.

### 5.2.1 Logistic Regression

Logistic regression, also referred to as the logit model, is a widely used and fundamental method for classification. This method was first introduced by David Cox [65] and has become a popular choice for both binary and multiclass classification. Logistic regression is a simple yet highly robust technique that is able to effectively predict the outcome of a binary classification problem.

The method is named *logistic regression* because it transforms a linear regression model by forming a linear combination of features, and then applying a nonlinear sigmoidal function to the result. This allows for the prediction of the probability of an event occurring, rather than a continuous value, making it well-suited for classification problems. The sigmoid function, also known as the logistic function, maps the input values to a range between 0 and 1, making it easy to interpret the output as a probability.

The sigmoid function can be formulated as:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (5.2)$$

As a result, for a given input vector  $X$  and binary output  $y$ , we can determine the logistic function as follows:

$$P(y = 1|x) = \frac{1}{1 + e^{-\beta_0 - \beta_1 x}} \quad (5.3)$$

where  $P(y = 1|x)$  is the probability of the event of interest ( $y=1$ ) occurring, given a set of predictors  $x$  and  $\beta_0$  and  $\beta_1$  are the coefficients of the logistic function, also known as the logit coefficients. We can use a negative log-likelihood to find the optimal betas in equation 5.3. Given  $x^{(i)}$  as the  $i^{th}$  training example and  $y^{(i)}$  as the true label of the  $i^{th}$  training example from a data sample of size  $m$ , the cost function for the best fit is given by:

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^m [y^{(i)} \log(h_{\theta}(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_{\theta}(x^{(i)}))] \quad (5.4)$$

where  $h_{\theta}(x^{(i)})$  is the predicted probability of the  $i^{th}$  training example belonging to class 1, which is given by the sigmoid function:  $h_{\theta}(x^{(i)}) = \frac{1}{1 + e^{-\theta^T x^{(i)}}}$  and  $\theta$  is the vector of parameters to be learned by the model.

L1 and L2 regularization are techniques used in logistic regression to prevent

overfitting by adding a penalty term to the cost function. The goal of regularization is to reduce the complexity of the model by adding a penalty term that discourages large weights.

L1 regularization, also known as Lasso regularization, adds the absolute value of the weights to the cost function. This results in some coefficients becoming exactly zero, effectively eliminating them from the model. This results in a sparse model, where only a subset of the features are used for prediction. The L1 regularization term is defined as:

$$\lambda \sum_{j=1}^n |w_j| \quad (5.5)$$

Where  $\lambda$  is the regularization parameter, and  $w_j$  is the weight for the  $j$ -th feature. The regularization parameter is a scalar that controls the strength of the regularization. A larger value of  $\lambda$  will result in smaller weights and a simpler model.

L2 regularization, also known as Ridge regularization, adds the square of the weights to the cost function. This results in all coefficients becoming smaller in magnitude, but none of them become exactly zero. This results in a model where all features are used for prediction, but with smaller coefficients. The L2 regularization term is defined as:

$$\lambda \sum_{j=1}^n w_j^2 \quad (5.6)$$

Where  $\lambda$  is the regularization parameter, and  $w_j$  is the weight for the  $j$ -th feature. Similar to L1 regularization, a larger value of  $\lambda$  will result in smaller weights and a simpler model.

The regularization term is added to the logistic regression cost function to obtain the regularized cost function. For L1 regularization, the regularized cost function is:

$$J(w) = -\frac{1}{m} \sum_{i=1}^m y^{(i)} \log(h_w(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_w(x^{(i)})) + \lambda \sum_{j=1}^n |w_j| \quad (5.7)$$

For L2 regularization, the regularized cost function is:

$$J(w) = -\frac{1}{m} \sum_{i=1}^m y^{(i)} \log(h_w(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_w(x^{(i)})) + \lambda \sum_{j=1}^n w_j^2 \quad (5.8)$$

Where  $J(w)$  is the regularized cost function,  $h_w(x)$  is the sigmoid function,  $m$  is the number of training examples,  $x^{(i)}$  and  $y^{(i)}$  are the  $i$ -th training example, and  $w$  is the weight vector.

In general, L1 regularization is preferred when there are many features and only a subset of them are important for prediction, while L2 regularization is preferred when all features are important but some of them may have very large coefficients.

A more detailed explanation of logit models can be found in [66]

### 5.2.2 Support Vector Machines

Support Vector Machines (SVM) are another group of popular supervised algorithms for classification tasks in Machine learning. A support vector machine creates a hyperplane or set of hyperplanes, which, in an  $m$ -dimensional space, is characterized as a flat affine subspace of  $m-1$ -dimensions to separate the data based on their target labels [67]. Figure 5-1a depicts an example of a 1-dimensional hyperplane (line) in a 2-dimensional space dividing the samples with regard to their class labels. In spite of that, not all datasets can be efficiently classified by a linear hyperplane. Therefore, support vector machines by means of *kernel* functions expand the data to high dimensional space. The kernel can be nonlinear, such as homogenous polynomial or Gaussian radial basis function. Figures 5-1b and 5-1c show polynomial and radial *kernels* in generating hyperplanes. Consequently, in contrast to logistic regression, SVMs can be used for both linear and nonlinear classification.

The *kernel*-based nature of SVMs makes them accurate and robust for classifying data with lots of features, presenting them as a popular method for the classification of clinical time series. For example, SVMs were utilized in EEG signal and heart rate classification [68, 69, 70].

The formulation of SVMs is lengthy and beyond the scope of this thesis; however, an interested reader can find a detailed description of the method in [71].

### 5.2.3 Random Forests

Random forests are an ensemble machine learning algorithm that combines multiple decision trees to make predictions. They are a type of supervised learning algorithm,

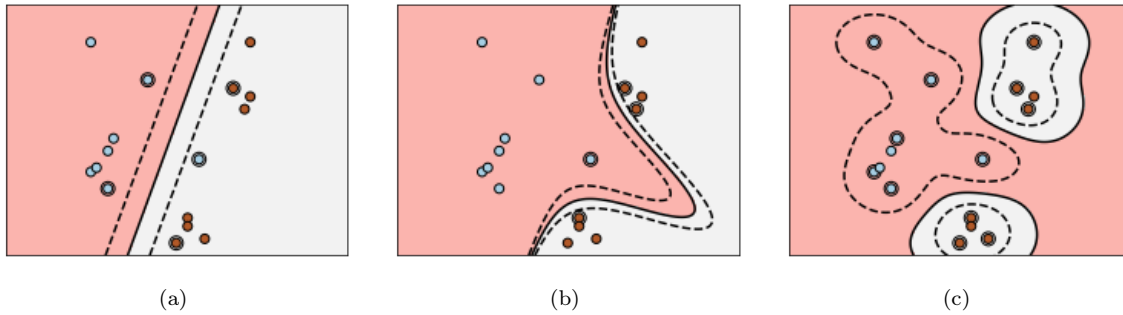


Figure 5-1: An example showing a) linear, b) polynomial and c) radial kernels in a 2-dimensional space dividing the samples with regard to their class labels

which means they are trained on a labeled dataset to make predictions on new, unseen data.

The basic idea behind random forests is to create multiple decision trees and then average their predictions to make a final prediction. Each decision tree is created by randomly selecting a subset of the training data, known as a *bootstrap sample*, and a subset of the features for each split. This allows for a diverse set of trees to be created, each with a different view of the data [72]. Figure X displays a simplified diagram of Random Forests.

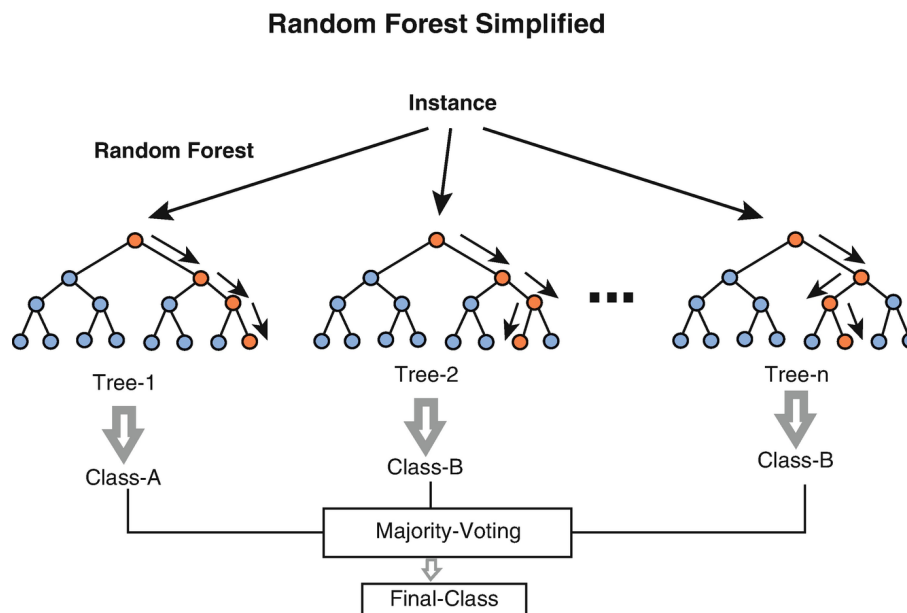


Figure 5-2: A simplified Random Forests schema [73]

Formally, a random forest is defined as an ensemble of decision trees, where each tree is constructed using a different bootstrap sample of the training data and a random subset of features for each split. The randomness in the data samples and

features selection leads to a reduction in overfitting, which is a common problem in decision trees.

Random forests are widely used in a variety of applications. They are known for their high accuracy, robustness, and ability to handle large datasets with many features. They are also relatively easy to implement and interpret.

A comprehensive introduction to decision trees and random forests is presented in [74].

## 5.3 Cross Validation

In machine learning, the process of determining the optimal parameters of a classification or regression model using a given dataset is known as training. However, utilizing the exact data for both training and testing can result in a phenomenon known as overfitting. This occurs when a model is overly attuned to the specific characteristics of the training dataset, leading to a high degree of accuracy when evaluated on that data, but poor performance when applied to new, unseen data.

To mitigate this issue, a technique known as cross-validation (CV) is often employed. The primary goal of CV is to assess the generalization performance of a model, or its ability to predict outcomes for new data accurately. This is achieved by dividing the original dataset into multiple subsets, training the model on one subset, and evaluating its performance on the remaining subsets. A more robust estimate of model performance on unseen data can be obtained by repeatedly training and evaluating the model on different subsets of the data.

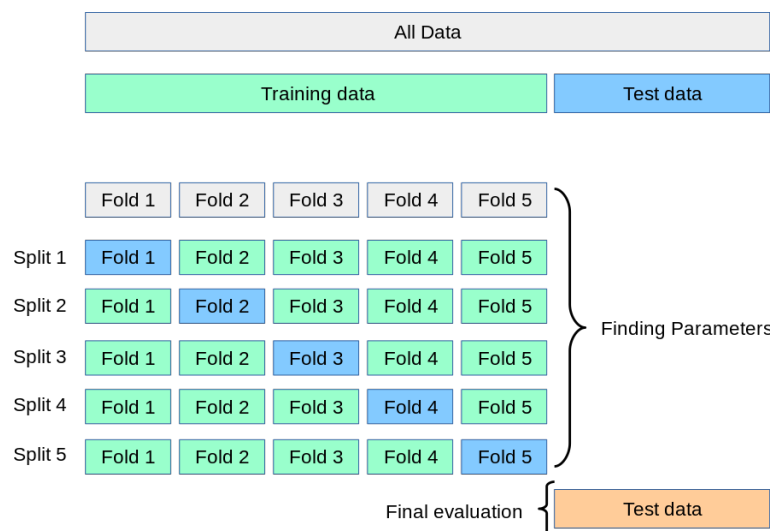


Figure 5-3: K-fold cross-validation schema [73]

K-fold cross-validation (K-fold CV) is a widely-used type of CV for evaluating the performance of a predictive model. This technique involves partitioning the data samples into  $k$  subsets, also known as folds, of approximately equal size. The process begins by selecting one of the folds as the test set and fitting the model on the remaining  $k-1$  folds. This procedure is then repeated  $k$  times, with each fold being used as the test set once. The average performance score across all iterations is calculated as the final measure of the model's performance. This method helps to provide a more robust estimate of the model's generalization performance by accounting for the potential variability in the data partitioning [71] (See Fig. 5-3).

In this thesis, we used two variations of K-fold CV. One is called Shuffled Stratified K-fold, which divides the dataset into  $K$  subsets, or folds, while ensuring that the ratio of samples belonging to each target class is maintained across all folds. This is accomplished through the use of a shuffling procedure, which randomly reassigns samples to the different folds while preserving the class proportions. Figure 5-4 represents Shuffled Stratified K-fold.

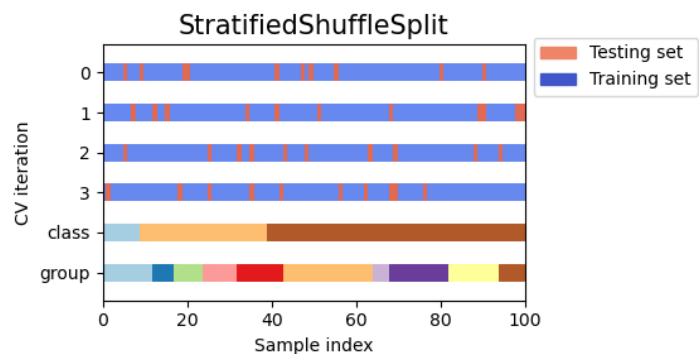


Figure 5-4: Shuffled Stratified cross-validation schema[73]

The other type of K-fold CV, used in this thesis, is Leave-One-Out (LOO) cross-validation method. This method involves training the model on all but one of the data samples and then evaluating the model's performance on the excluded sample. This process is repeated for each sample in the dataset, resulting in a separate model being trained and evaluated for each data point. The performance metric is then calculated as the average of the performance of all the models. This method is particularly useful when the dataset is small or when the goal is to obtain an unbiased estimate of the model's generalization performance.

## 5.4 Clustering

### 5.4.1 Consensus Clustering

In this thesis, we used a clustering algorithm based on Monti's consensus model[75].

This module can be used to determine the number of stable clusters ( $K$ ) in a dataset ( $D$ ) by performing multiple runs of a clustering algorithm, here K-Means, on randomly drawn subsamples of the data in order to improve the overall performance and robustness of the clustering process. This method have several advantages over traditional single-algorithm clustering approaches.

One key advantage of consensus clustering is that it can increase the robustness of the clustering process by reducing the impact of individual algorithm-specific biases and errors.

Another advantage of consensus clustering is that it can be used to handle complex and high-dimensional data sets, which can be difficult for individual algorithms to process. By combining the results of multiple runs, consensus clustering can extract more meaningful and informative clusters from these data sets.

Moreover, consensus clustering methods can also be used to identify and eliminate outliers and noise, which can be challenging for single-algorithm clustering approaches. This can lead to more accurate and robust clustering results.

Given a dataset of  $N$  points, implementing the algorithm computes a consensus matrix ( $N * N$ ) for each  $K$ , where rows and columns represent the points in the data, and each element denotes the fraction of times two points clustered together after multiple times of clustering resampled data. An ideally stable clusters structure in the data leads to a connectivity matrix that consists entirely of zeros and ones, representing all point pairs always clustering together or not together overall resampling repetitions. In other words, putative similar clusters in each run show that the obtained clusters are robust to sampling variability and represent an accurate partitioning in the data with the selected  $K$  clusters.

Given a dataset of interest  $D$  with  $N$  samples, let  $D^1, D^2, D^3, \dots, D^H$  be the list of  $H$  resampled datasets with a sampling ratio  $R$ . The connectivity matrix  $M^h$  (connectivity matrix corresponding to dataset  $D^h$ ) is generated and as follows:

$$M^h(i, j) = \begin{cases} 1 & \text{if samples } i \text{ and } j \text{ belong to the same cluster,} \\ 0 & \text{otherwise.} \end{cases}$$

In order to track which samples were selected during each resampling, an Indicator

matrix is defined, as follows:

$$I^h(i, j) = \begin{cases} 1 & \text{if both samples } i \text{ and } j \text{ are present in the dataset } D^h, \\ 0 & \text{otherwise.} \end{cases}$$

The consensus matrix  $C$  is defined as the sum of all connectivity matrices normalised by the corresponding indicator matrices:

$$C(i, j) = \frac{\sum_{h=1}^H M^h(i, j)}{\sum_{h=1}^H I^h(i, j)} \quad (5.9)$$

The consensus matrix is symmetric, and each element is between 0 and 1. This substantial property can be used as a similarity measure with a hierarchical tree model to have a final clustering of dataset  $D$ . More specifically,  $1 - C$  represents a new distance matrix that can be used instead of the conventional measures, such as Euclidean distance, Pearson correlation. In our model, we implement a ward hierarchical clustering model with a maximum number of clusters equal to  $K$  to cluster the  $D$  using the distance matrix obtained from the consensus matrix.

In the next step, a statistical score is derived from the consensus matrix to quantify the stability of each data sample within a cluster. We define this score for each sample of  $D$  and each cluster  $k$  as follows:

$$m_i(k) = \frac{1}{N_k - 1} \sum_{\substack{j \in I_k \\ i \neq j}} M(i, j), \quad (5.10)$$

where  $I_k$  is the set of indices of samples belonging to cluster  $k$ , and  $N_k$  is the size of cluster  $k$ .

In an ideal case of stability,  $m_i(k)$  would be 1 for all the data in cluster  $k$ . In other scenarios,  $m_i(k)$  has a value between 0 and 1. In this model, we define a *threshold* to separate the unstable samples in each cluster. For each  $i$  in cluster  $k$ , if  $m_i(k)$  is smaller than *threshold*, the sample is labeled as -1, implying that the sample belongs to no stable cluster and is considered noise. *threshold* is an important parameter to control the robustness of the obtained clusters. When chosen too large, most data will be labeled as noise. When chosen too small, it leads to unstable clusters.

Given a set of  $K$ s and a predefined *threshold*, we run the whole process for every  $K$ . The  $K$  that leads to a small number of unstable samples (labeled as -1) represents the optimal  $K$ .

### 5.4.2 Clusters Enrichment Analysis

In many real-world scenarios, clinical datasets contain categorical information such as phenotypes, diagnoses, and demographic information of samples in data. This type of data, which can be binary (e.g. gender) or multiclass (e.g. age groups), is referred to as true-labels in data analysis. In the context of cluster analysis, clusters enrichment is the process of using statistical methods to associate the true-labels with the clusters that are identified in the data, in order to identify meaningful groups of samples.

In this thesis, we employed Fisher’s exact test [5.1.2](#) to calculate the p-value, which indicates the relationship between a categorical variable population ratio in one cluster and the entire data set. We then defined the enrichment score as the product of the p-value and the difference in population ratio between the two groups.



# Application Showcase: ECG Signals Classification and Clustering

## 6.1 Introduction

Cardiovascular disease (CVD) is known as the dominant cause of mortality worldwide [76]. The first step in reducing mortality is a reliable and accurate disease diagnosis [77]. Monitoring devices have facilitated a more effective diagnosis for patients with cardiac disease due to their ability to monitor heart activity continuously. In particular, an electrocardiogram (ECG), as a noninvasive and dedicated signal, is used to evaluate the general cardiac state of a patient and is, consequently, a first-in-line inspection for the diagnosis of CVD [78].

Cardiac diseases alter different parts of the ECG components, which cardiologists manually observe and interpret based on a collection of diagnostic measures and domain knowledge. Nevertheless, such manual ECG examination is not timewise efficient and needs skill [78]. Therefore, building an accurate and robust automated workflow for classifying and diagnosing CVD using ECG signals is an interesting research subject for scientists. This study presents a workflow for classifying ECG records using the PTB-XL public dataset. [79].

## 6.2 ECG Signal

The heart is a muscle that pumps blood all over the body in rhythmical contraction. This rhythm is induced by the heart's pacemaker cells, the sinoatrial (SA) node. It generates an electrical signal that runs through various heart nerve cells and leads to a specific series of muscle contractions resulting in the heart's pumping activity [80]. This electrical activity yields electrical currents on the body's surface, inducing skin

surface electrical potential deviations. These changes can be captured and isolated by means of electrodes and appropriate equipment. The graphical representation of the measured signal is called electrocardiogram (ECG) [81].

A typical configuration of electrodes is 10 electrodes composition (12 leads) [82]: one of the electrodes is placed on the left arm (LA), one on the right arm (RA), one on the left leg (LL), one on the right leg (RL), and six on the chest (V1 to V6). Figure 6-1 displays this configuration. The signals obtained from 12 channels are called: I, II, III, aVR, aVL, aVF, V1, V2, V3, V4, V5, and V6.

ECG signal is described by five segments, known as PQRST complex. First, the

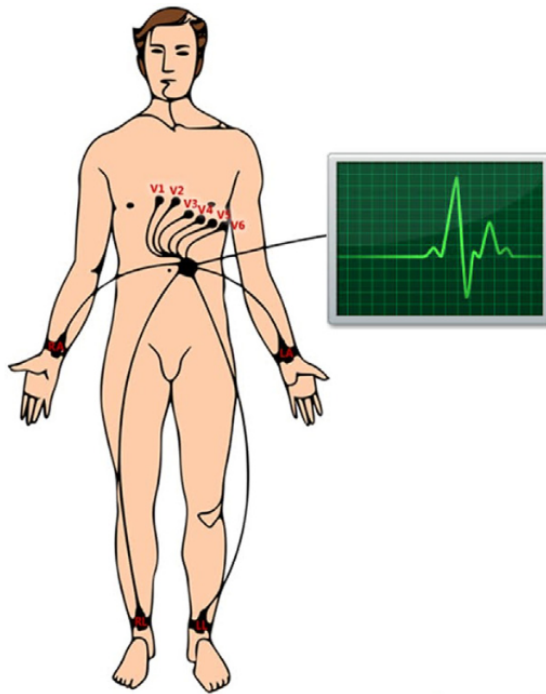


Figure 6-1: ECG typical 10 electrodes configuration [81]

SA node creates an electrical signal propagated through the right and left atrium, resulting in the contraction of the myocardium (P wave). Then the internodal tracts assist the signal in transiting to the atrioventricular node, the electrical gateway of the heart. Due to the delaying feature of the Atrioventricular Node (AV) node, the ventricles and the atria do not contract concurrently, causing the formation of the PR segment. The signal crosses the distal bundle of His and contacts the ventricle by splitting into the left and right ventricular septum (Q wave). Following the Q wave, the R wave arises due to the depolarization of the ventricles. The S wave follows the R wave as the last phase of ventricular depolarization at the base of the heart. Finally, the ventricular repolarization and the subsequent relaxation of the

myocardium generate the T wave [83]. Figure 6-2 depicts the described conduction system with its components.

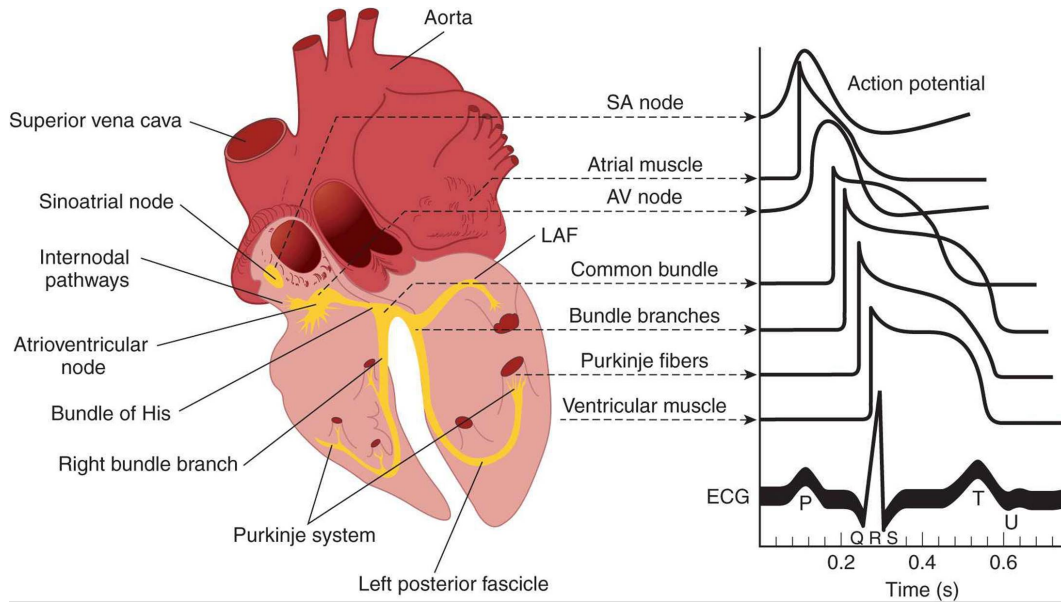


Figure 6-2: A depiction of anatomy of heart main components involve in its conduction system and the corresponding electrical potential waveform form an ECG signal[84]

## 6.3 Data

The PTB-XL dataset [79] contains 21837 clinical 12-lead ECG (I, II, III, aVL, aVR, aVF, V1–V6) records from 18885 patients, where the length of each record is 10 seconds. Additionally, the dataset provides gender (52% male and 48% female) and age (ranging from 0 to 95 with a median of 62 and interquartile range of 22) of the patients. The ECG records are labeled with at least one ECG statement out of 71 different statements, covering 44 diagnostic, 12 rhythm, and 19 form statements, 4 of which are also used as diagnostic ECG statements. Form refers to considerable changes of certain elements within the ECG, such as abnormal QRS complex, and rhythm is connected to specific changes of the rhythm, such as atrial fibrillation. The diagnostic statements are decomposed into five superclasses and 24 subclasses. The annotation process was done automatically, validated by up to two cardiologists, and presented according to the SCP-ECG standard. Each record label is a set of statements and corresponding confidence likelihoods from 15% to 100%. In our analysis, we filtered out all the diagnostic statements with a probability below 100%, resulting 17259 records. Tables 6.1 and 6.2 summarize the number of individual classes and

subclasses. The ECG records were sampled originally at 400 Hz frequency; however, they were published in an upsampled version (500 Hz) and downsampled version (400 Hz). In the present study, we utilized the 500 Hz sampling frequency records.

| Class | Description            | Number<br>of<br>Records |
|-------|------------------------|-------------------------|
| NORM  | Normal ECG             | 7185                    |
| CD    | Conduction Disturbance | 4787                    |
| STTC  | ST/T change            | 4579                    |
| MI    | Myocardial Infarction  | 2984                    |
| HYP   | Hypertrophy            | 1473                    |

Table 6.1: Numbers of ECG records of individual diagnosis classes in PTB-XL database[79]

## 6.4 Feature Extraction

According to the Nyquist theory, the frequency range of the ECG signals spectral-domain with 500 Hz sampling frequency is 0-250 Hz [85]. Despite that, exploring the whole frequency band is unnecessary for ECG analysis, as many studies suggest that only a specific frequency band carries useful information for frequency-based feature analysis. In general, it is accepted that the spectrum of a typical QRS contains almost all the power below 30 Hz, with a peak occurring in the range of 4 to 12 Hz, and higher frequencies represent the noise in the data (For example, noise coming from the electrical network is in the range of 50-60 Hz) [86]. In particular, low-frequency parts of the ECG such as low-power P- and T-waves, PQ-, ST- and TP- segments can be detected below 14 Hz, while the high-power P- and T-waves are located in the 20-30 Hz band. In the context of cardiac diseases, for instance, Atrial Fibrillation episodes lead to higher power concentration in the 4-9 Hz frequency range [87]. Figure 6-3 displays a representation of the relative spectral power and different components frequencies of a healthy subject's ECG signal. Despite the accepted mentioned frequency band, it is reported in various research that the contents of high-frequency components of QRS complex can be observed below 100 Hz, and in unhealthy cases, analysis frequencies >100 Hz might be clinically informative too [89, 90]. For instance, ventricular tachycardia (VT) initiation can be characterized by high-frequency segments above at least 30 Hz and significantly above 70 Hz [91]. Moreover, baseline drift causes interference data below 0.5 Hz frequency. Therefore, we applied a highpass and lowpass filter with

| Class | Subclass  | Description                                                  | Number of Records |
|-------|-----------|--------------------------------------------------------------|-------------------|
| NORM  | NORM      | Normal ECG                                                   | 7185              |
| CD    | LAFB/LPFB | left anterior/left posterior fascicular block                | 1754              |
|       | IRBBB     | incomplete right bundle branch block                         | 1074              |
|       | ILBBB     | incomplete left bundle branch block                          | 74                |
|       | CLBBB     | complete left bundle branch block                            | 535               |
|       | CRBBB     | complete right bundle branch block                           | 537               |
|       | _AVB      | AV block                                                     | 812               |
|       | IVCD      | non-specific intraventricular conduction disturbance (block) | 778               |
|       | WPW       | Wolff-Parkinson-White syndrome                               | 71                |
|       | LVH       | left ventricular hypertrophy                                 | 1089              |
|       | RVH       | right ventricular hypertrophy                                | 55                |
| HYP   | LAO/LAE   | left atrial overload/enlargement                             | 314               |
|       | RAO/RAE   | right atrial overload/enlargement                            | 82                |
|       | SEHYP     | septal hypertrophy                                           | 17                |
| MI    | AMI       | anterior myocardial infarction                               | 1969              |
|       | IMI       | inferior myocardial infarction                               | 1447              |
|       | LMI       | lateral myocardial infarction                                | 57                |
|       | PMI       | posterior myocardial infarction                              | 4                 |
| STTC  | ISCA      | ischemic in anterior leads                                   | 774               |
|       | ISCI      | ischemic in inferior leads                                   | 284               |
|       | ISC_      | non-specific ischemic                                        | 1122              |
|       | STTC      | ST-T changes                                                 | 2052              |
|       | NST_      | non-specific ST changes                                      | 533               |

Table 6.2: Numbers of ECG records of individual diagnosis subclasses in PTB-XL database[79]

0.5 and 130 Hz cutoffs in this study, respectively.

### 6.4.1 Spectral Features

For our analysis, we represented each ECG record via a set of spectral features directly from the signal’s power spectrum and implicitly using stationary wavelet transformation. We generated the first feature data by extracting the following features from the power spectrum density of ECG records, using the built-in functions of CMDA package:

- MNF: Mean Frequency

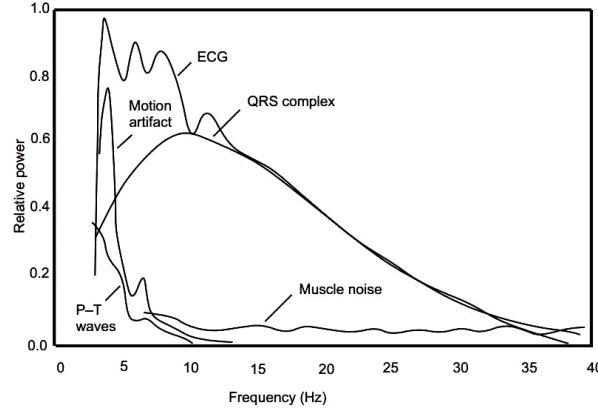


Figure 6-3: QRS complex frequency range of a healthy ECG signal [88]

- MDF: Median Frequency
- PSR: Power Spectrum ratio
- VCF: Variance of Central Frequency
- Spectral Entropy
- Permutation entropy with  $m=3$  and  $m=4$
- Sample entropy with  $m=2$  and  $m=3$

We also calculated the power, MNF, VCF, Peak, and Entropy of the PSD of the following manually segmented frequency bands: 0.5-1 Hz, 1-2 Hz, 2-4 Hz, 4-6 Hz, 6-9 Hz, 9-12 Hz, 12-15 Hz, 15-20 Hz, 20-30 Hz, 30-70 Hz, 70-125 Hz. The feature generation process from each channel of 12-lead ECG data leads to a 2-dimensional feature space with 21837 rows (number of records) and 744 variables. We called the obtained data spectral features. The whole process took 29430+3300 seconds running on 1 CPU core of a machine with a 3.5 GHz Dual-Core Intel Core i7 processor and 16 GB 2133 MHz LPDDR3 memory. Running the process using all four cores of the CPUs reduced the time to 11733+1106 seconds, demonstrating the power of using the package in parallel mode.

### 6.4.2 SWT Features

The next feature representation was performed using Stationary Wavelet Transform (SWT). As explained in section X, SWT decomposes the signal into different scales coefficient, representing its different spectral-domain characteristics. A group of SWT scales carries practical and informative features in ECG signals. Hence, we applied SWT on the ECG records and picked only specific scales for the next feature extraction step [92]. In this study, we employed the 9 scale SWT, resulting in the decomposition of the raw ECG signal into 10 sets of wavelet coefficients with the

following frequency bands: 125-250 Hz, 62.5-125 Hz, 31.25-62.5 Hz, 15.63-31.25 Hz, 7.81-15.63 Hz, 3.91-7.81 Hz, 1.95-3.91 Hz, 0.98-1.95 Hz, 0.49-0.98 Hz, and 0-0.49 Hz. The informative part of an ECG record can be discovered in the 0.5125 Hz interval. Therefore, we selected only the SWT coefficients corresponding to the mentioned frequency bands, including D2 to D9 scales, and excluded the remaining scales (D1, A9). Figure 6-4 exhibits the decomposition of a sample ECG signal using a 9 scale SWT. Subsequently, we extracted the following features from each coefficient vector:

- SEE: Shannon Entropy
- MNF: Mean Frequency
- PSR: Power Spectrum ratio
- NSE: Normalized Sub-band Energy
- MDS: Median of Slopes
- MNS: Mean of Slopes
- peak frequency
- Permutation entropy with  $m=3$

Applying the features to all ECG channels' SWT coefficients scales (D2-D9) resulted in 2-dimensional data of 21837 rows (number of records) and 768 variables. We ran the process of operating the CMDA pipeline on a similar machine, where the computation time of 1 CPU core was 3024+38, and 4 CPU cores were 1321+20 seconds. Likewise, we observed a notable reduction in the computation using the pipeline in a multiprocessing manner.

## 6.5 Results

### 6.5.1 Classification

Using the two generated feature datasets, we employed a classification model to predict the diagnostic statements of each record. As stated, some records have more than one statement; hence we used the one-vs-rest (OvR) strategy. This study aimed not to find the optimal prediction but merely to show how one can achieve reliable, fast, and interpretable results in a transparent way. Therefore we utilized random forests (See Sect. 5.2.3) as a robust classifier model without any parameter tuning. Besides, we handled the imbalanced labels issue (See Table.6.1) by performing the classification in a bootstrap sampling manner, where at each run, random samples without replacement with similar size were picked from each label, and the classifier model was fitted with 5-fold stratified cross-validation (See Sect.5.3) on the sample

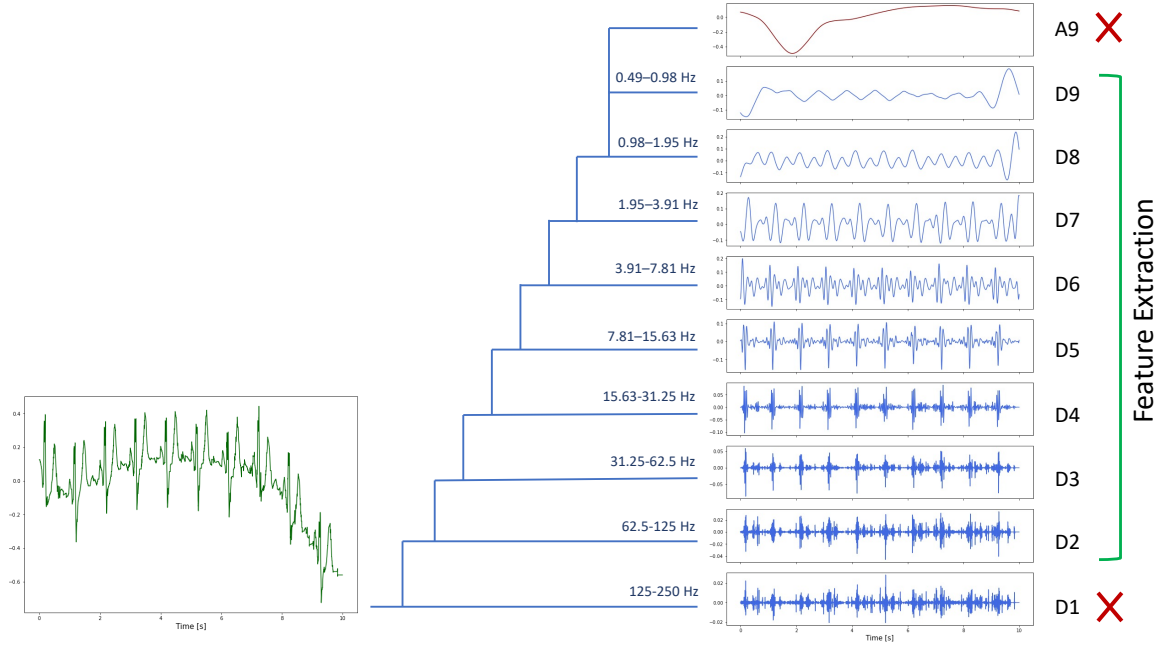


Figure 6-4: Flowchart of ECG signal decomposition using 9 scale SWT

data. The process was repeated 50 times, and average values of prediction metrics were reported.

| Diagnosis Statement | Metric   |      |           |        |      |
|---------------------|----------|------|-----------|--------|------|
|                     | Accuracy | AUC  | Precision | Recall | F1   |
| NORM                | 0.84     | 0.91 | 0.84      | 0.85   | 0.85 |
| CD                  | 0.79     | 0.87 | 0.81      | 0.75   | 0.78 |
| STTC                | 0.79     | 0.87 | 0.76      | 0.84   | 0.80 |
| HYP                 | 0.80     | 0.88 | 0.79      | 0.81   | 0.80 |
| MI                  | 0.79     | 0.87 | 0.78      | 0.81   | 0.80 |
| Average             | 0.79     | 0.87 | 0.79      | 0.80   | 0.79 |

Table 6.3: Classification metrics of individual diagnosis classes, and the average measures computed from a One-vs-All Random Forests model that was employed in a bootstrapping fashion to classify diagnosis classes using Spectral features data.

Tables 6.3 and 6.4 show the average classification measures (Accuracy, AUC, Precision, Recall, and F1) using spectral and SWT features, respectively. It can be seen that using SWT features leads to slightly better results in classifying Normal, Hypertrophy (HYP), Myocardial Infarction (MI), and ST/T change (STTC), while there is no difference in the performance of Cardiac Disturbance (CD) prediction. Our model achieved AUC=0.93 in the classification of Normal ECG records, which is close to

| Diagnosis Statement | Metric   |      |           |        |      |
|---------------------|----------|------|-----------|--------|------|
|                     | Accuracy | AUC  | Precision | Recall | F1   |
| NORM                | 0.86     | 0.93 | 0.85      | 0.87   | 0.86 |
| CD                  | 0.79     | 0.87 | 0.81      | 0.73   | 0.77 |
| STTC                | 0.81     | 0.89 | 0.77      | 0.87   | 0.82 |
| HYP                 | 0.82     | 0.90 | 0.81      | 0.83   | 0.82 |
| MI                  | 0.80     | 0.88 | 0.77      | 0.84   | 0.80 |
| Average             | 0.80     | 0.89 | 0.80      | 0.82   | 0.80 |

Table 6.4: Classification metrics of individual diagnosis classes, and the average measures computed from a One-vs-All Random Forests model that was employed in a bootstrapping fashion to classify diagnosis classes using SWT features data.

the AUC=0.96 reported by Śmigiel et al. [93], implementing a complex convolutional network on original signals combined with entropy features. They also reported the average AUC=0.91 for the classification of all classes, while we obtained 0.89. In another study [94], different models were tested, where a Resnet-based architecture delivered the best performance (AUC=0.93), while A simple neural network trained on wavelet features showed AUC=0.87.

Additionally, we calculated the AUC in classifying the diagnosis subclasses described in table 6.2, after discarding subclasses with lower than 20 encounters (PMI and SEHYP). Figure 6-5 depicts the distribution of AUCs of individual subclasses computed in a bootstrapping fashion. We can observe a near 1 AUC of CLB BB. LAFB/RAFB and CRBBB show a very high AUC (>0.95) among CD diagnoses. LVH has the best AUC in HYP class and ICS among STTC classes.

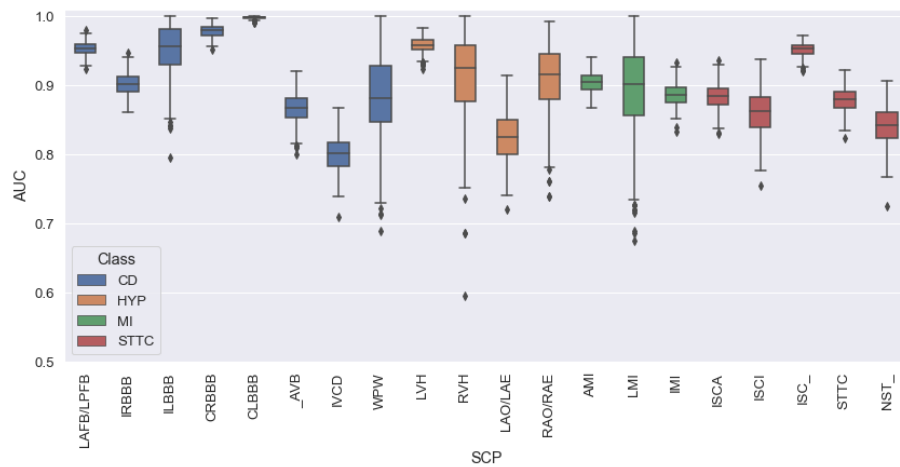


Figure 6-5: AUCs distribution of individual diagnosis subclasses; A One-vs-All Random Forests model was employed in a bootstrapping fashion to classify subclasses using SWT features data.

Next, we found statistically pronounced features discriminating each class from

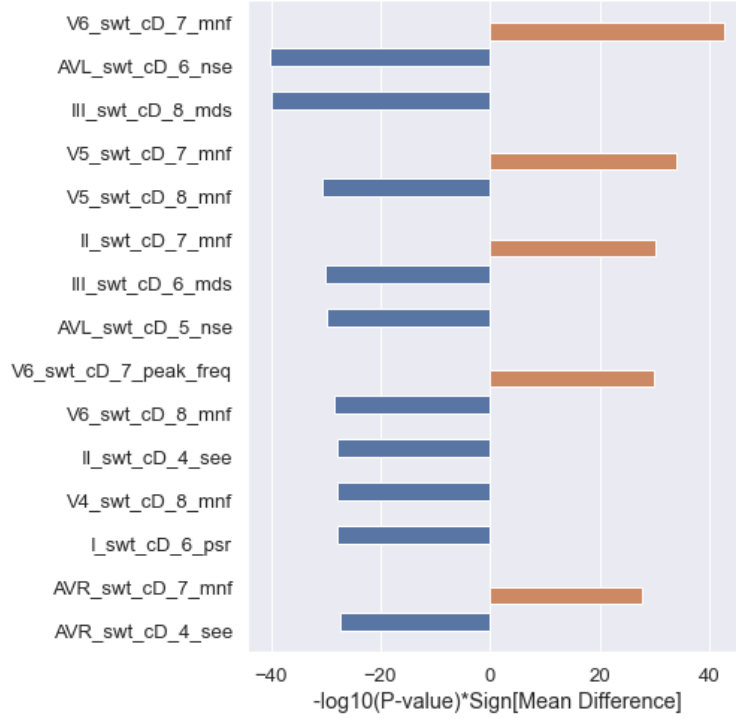


Figure 6-6: The first 15 most differentiating features in *Normal* class, resulted from calculating the  $-\log_{10}(p\text{-value}) * \Delta\text{mean}$  by applying a Ranksum significance test between the *Normal* class and the rest of the data.

the rest of the data. For this purpose, first, we removed the features with high correlations ( $>0.95$ ). Then, we applied a Ranksum significance test on each feature in a bootstrapping fashion. Figure 6-6 shows the  $-\log_{10}(p\text{-value}) * \Delta\text{mean}$  of the first 15 features, differentiating the normal signals from the rest of the records. Increasing D7 SWT scales mean frequency (1.95-3.9 Hz) and reduction in median deviation and power at D6 (3.9 - 8 Hz) and D8 (1-1.95 Hz) indicate low relative power concentration in non-normal records at 2-4 Hz band and spreading the power to higher ( $>4$  Hz) and lower frequencies ( $<2$  Hz). Figure 6-7 depicts the results obtained from the same process in other major classes (CD, MI, STTC, HYP). For example, in HYP class (See Fig.6-8e), we can observe the overexpression of all features in 1-15 Hz interval (D5-D8), indicating power spectrum energy increase and variation towards higher frequencies in this band. On the other hand, we can notice a shift of mean frequency towards low frequencies and a decline in relative energy in the 4-30 Hz band(D4-D6) in myocardial infarction (MI) records (See Fig.6-8c).

Another observation is that the expressed features are mostly from specific ECG leads. Keeping all the features with a p-value lower than 97.5% of all the p-values showed that the expressed features are mostly from III, aVL, aVF, and aVR leads in

the CD group. Quite the opposite, V1, V5, and V6 have the most prominent features in the HYP group.

### 6.5.2 Clustering

Finally, we explored the clustering of ECG records using their SWT features. For this purpose, we applied the consensus clustering described in section 5.4.1, presuming  $K$ : 2-8 clusters, number of iterations: 100 and threshold: 0.3. The correct number of clusters is subjective and depends on many factors, e.g., the stability and robustness of the obtained clusters. The consensus clustering method discovers the stable clusters by nature, considering any number of clusters ( $K$ ); however, one determining factor in selecting the number of clusters using this strategy can be the number of unstable (data points not in any stable clusters) data. Figure 6-9 reveals that  $K=2,3,5,6$  contributed to relatively lowest number of unstable data points.  $K=6$  shows an approximately similar cluster distribution to  $K=5$ , while the smallest cluster has smaller members than the unstable group. Therefore, we picked only  $K=2,3$  and 5 for further analysis (0.7%, 1.5% and 3.2% unstable points, respectively).

Tables 6.5, 6.6 and 6.7 show the ratio of diagnosis classes in each cluster. Given  $K=2$ , we have the highest proportion of Normal records in cluster 1 ( $K=2$ ). However, by increasing the number of clusters to 3 or 5, we found a cluster without almost any NORM records (cluster 1, where  $K=3$  and cluster 4, where  $K=5$ ). This explains the misclassification of some records in the classification process; for instance, the data with other diagnosis classes in cluster 0 have more similarity to the NORM class and will be classified as NORM. Generally,  $K=5$  gives us a finer separation of the other diagnostic labels in each cluster. For example, we can see that nearly half of the MI and STTC records are in cluster 1, or cluster 4 designate a specific group of CD and HYP records, whereas other diagnostic classes have a low presence in this cluster. We performed an enrichment analysis to understand better the association between the obtained clusters and diagnosis classes (See Sect. 5.4.2). The enrichment analysis results (See Fig. 6-10) confirm a statistical enrichment of CD and HYP records in cluster 4 and negligible presence of NORM records in clusters 1 and 4. We can notice a slight enrichment of STTC class in cluster 1 as well.

The enrichment analysis of the diagnosis subclasses, displayed in figure 6-11, shows a considerable enrichment of CLBBB in cluster 4. On the other, IRBBB records, as well as STTC and NORM, have a low presence in cluster 4. This explains the high AUC computed in the classification of CLBBB (See Fig. 6-5). We can notice the

| Class | Cluster |       |       |
|-------|---------|-------|-------|
|       | -1      | 0     | 1     |
| MI    | 0.006   | 0.362 | 0.631 |
| STTC  | 0.011   | 0.349 | 0.640 |
| CD    | 0.007   | 0.459 | 0.533 |
| HYP   | 0.010   | 0.513 | 0.476 |
| NORM  | 0.005   | 0.129 | 0.866 |

Table 6.5: Extracted SWT features data of ECG records were clustered using the consensus clustering method. The population ratio of diagnosis classes in each cluster was computed, given  $K=2$ .

| Class | Cluster |       |       |       |
|-------|---------|-------|-------|-------|
|       | -1      | 0     | 1     | 2     |
| MI    | 0.022   | 0.561 | 0.101 | 0.316 |
| STTC  | 0.023   | 0.542 | 0.068 | 0.368 |
| CD    | 0.028   | 0.455 | 0.214 | 0.304 |
| HYP   | 0.032   | 0.428 | 0.228 | 0.312 |
| NORM  | 0.007   | 0.750 | 0.000 | 0.243 |

Table 6.6: Extracted SWT features data of ECG records were clustered using the consensus clustering method. The population ratio of diagnosis classes in each cluster was computed, given  $K=3$ .

enrichment of LAFB/RAFB, LVH and ISC records in cluster 4 as well.

Finally, we applied the represented bootstrapping classification model to classify each diagnosis class in each cluster from the rest of the data. For example, table C.5 shows the classification metrics of CD records in each cluster, given  $K = 5$ . We observe an AUC=1.00 in predicting CD records in cluster 4, indicating a maximum separability of CD records from the other records in this cluster. Contrary to cluster 4, cluster 0 shows the lowest AUC=0.87, where we have the highest similarity to most of the NORM signals, implying the misclassification of some CD records as NORM. In general, clustering the records using CD-specific features enhanced the weighted average of all the metrics. We carried out a similar process for classifying other diagnosis classes. The obtained classification metrics are shown in Appendix. C. Figure 6-12 demonstrates the distribution of AUCs in each cluster for classifying diagnosis classes. The corresponding weighted average AUCs are higher than the reported ones employing a general model (See Table. 6.4) and comparable to the reported AUCs in the literature [93, 94].

The obtained results also suggest that an ensemble approach for classifying signals might improve the general prediction performance. In other words, instead of fitting one model on the data, we can categorize the data into meaningful groups and fit

| Class | Cluster |       |       |       |       |       |
|-------|---------|-------|-------|-------|-------|-------|
|       | -1      | 0     | 1     | 2     | 3     | 4     |
| MI    | 0.038   | 0.109 | 0.426 | 0.237 | 0.096 | 0.095 |
| STTC  | 0.032   | 0.068 | 0.431 | 0.301 | 0.104 | 0.065 |
| CD    | 0.035   | 0.121 | 0.307 | 0.225 | 0.104 | 0.209 |
| HYP   | 0.039   | 0.109 | 0.297 | 0.264 | 0.075 | 0.216 |
| NORM  | 0.031   | 0.496 | 0.178 | 0.231 | 0.063 | 0.000 |

Table 6.7: Extracted SWT features data of ECG records were clustered using the consensus clustering method. The population ratio of diagnosis classes in each cluster was computed, given K=5.

| Cluster          | Metric   |      |           |        |      |
|------------------|----------|------|-----------|--------|------|
|                  | Accuracy | AUC  | Precision | Recall | F1   |
| 0                | 0.81     | 0.87 | 0.77      | 0.88   | 0.82 |
| 1                | 0.84     | 0.91 | 0.81      | 0.89   | 0.85 |
| 2                | 0.84     | 0.92 | 0.80      | 0.92   | 0.86 |
| 3                | 0.93     | 0.97 | 0.89      | 0.98   | 0.94 |
| 4                | 0.98     | 1.00 | 0.97      | 0.99   | 0.98 |
| Weighted Average | 0.87     | 0.93 | 0.84      | 0.93   | 0.88 |

Table 6.8: Extracted SWT features data of ECG records were clustered using the consensus clustering method. A One-vs-All Random Forests model was employed in a bootstrapping fashion to classify CD records. Classification metrics of CD prediction in individual clusters, as well as the average weighted metrics were computed, given K=5.

individualized classifiers on each cluster.

## 6.6 Discussions

Patient monitoring devices record millions of data points per hour and can generate thousands of records. These biosignals play a vital role in diagnosing and treatment of various patient disorders and illnesses; however, manual assessment and examination of such signals can be challenging and prone to error. Over the past decades, countless attempts have been made to automatize the analysis procedure using computational methods, e.g., machine learning models. ECG signal examination is one of the most standard diagnostic strategies for cardiovascular diseases. Hence, an automatic interpretation of ECG records can enhance the diagnostic procedure. This chapter presented a well-defined procedure for examining large ECG datasets, including importing, denoising, and feature extraction by operating our proposed time series analysis tool. We employed our tool on the PTB-XL dataset, a database of ECG signals containing 21873 records, and extracted two feature sets from each record (Spectral

and SWT features). Next, we showed how the extracted feature data could be used to classify diagnostic classes by conducting a random forests bootstrap classification process. The calculated classification metrics showed a better performance of SWT over spectral features in separating five major diagnosis classes (NORM, CD, MI, HYP, and STTC); However, the state-of-the-art deep learning approaches outperformed our model by 2 to 3 %. Despite the inferior performance of our model, our approach objective was not to achieve the best result, and consequently, no model parameter optimization was done. In contrast, we offered a robust, reliable, and easy to implement and interpret procedure, especially for clinicians and non-deep learning experts. Moreover, due to the fast operating characteristic of the tool, the presented procedure can be employed by experts to get initial results and justify using more sophisticated and complex models such as deep learning. For instance, we demonstrated a statistical analysis to discover the most critical features and ECG leads in discriminating diagnosis statements. Additionally, we showed how a cluster-based classification ensemble could outperform one generic model running on the whole dataset. In this approach, we discovered stable groups and fitted a classifier for predicting the target class inside the cluster. This approach improved the overall classification performance and made it proximate to the reported ones in the literature. This strategy can be implemented on any large dataset, yet one of the limitations is the size of the target group in the data since the method requires an adequate number of target class data in individual clusters.

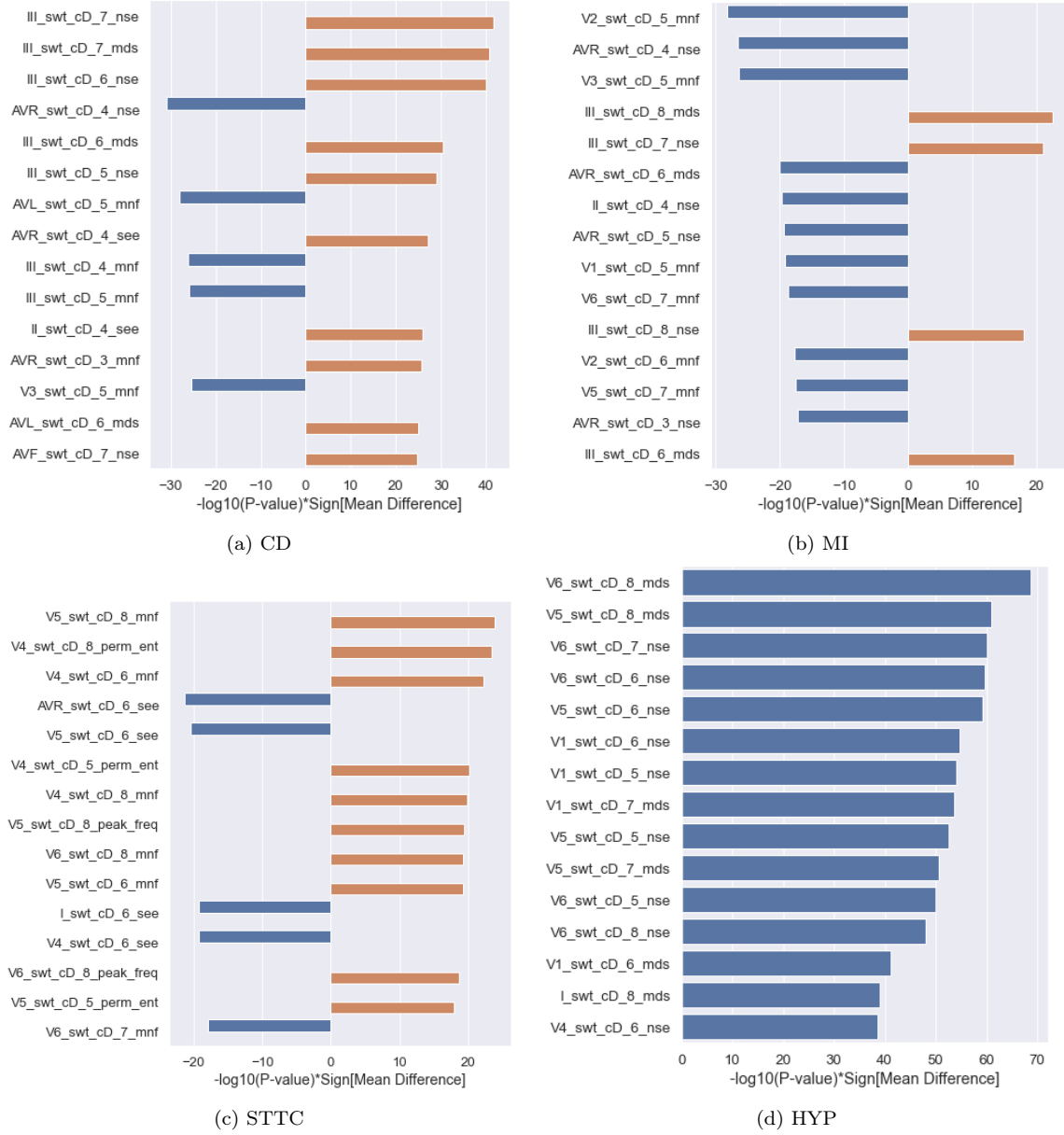


Figure 6-7: The first 15 most differentiating features in diagnosis classes, resulted from calculating the  $-\log_{10}(p - \text{value}) * \Delta_{\text{mean}}$  by applying a Ranksum significance test between each diagnosis class and the rest of the data.

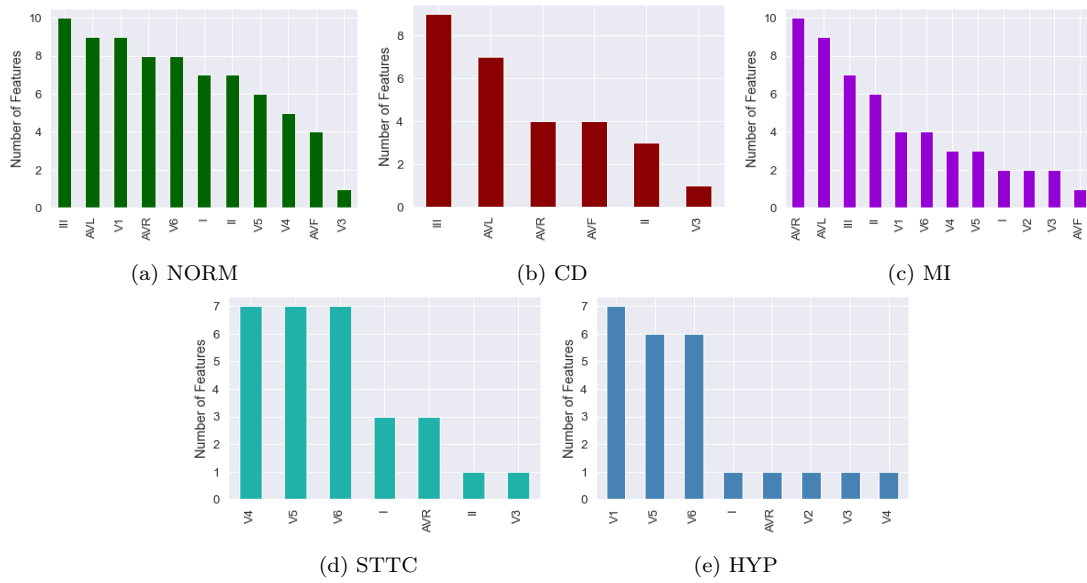


Figure 6-8: Number of expressed extracted features (p-value lower than 97.5% of all the p-values) from each ECG lead of individual diagnosis group.

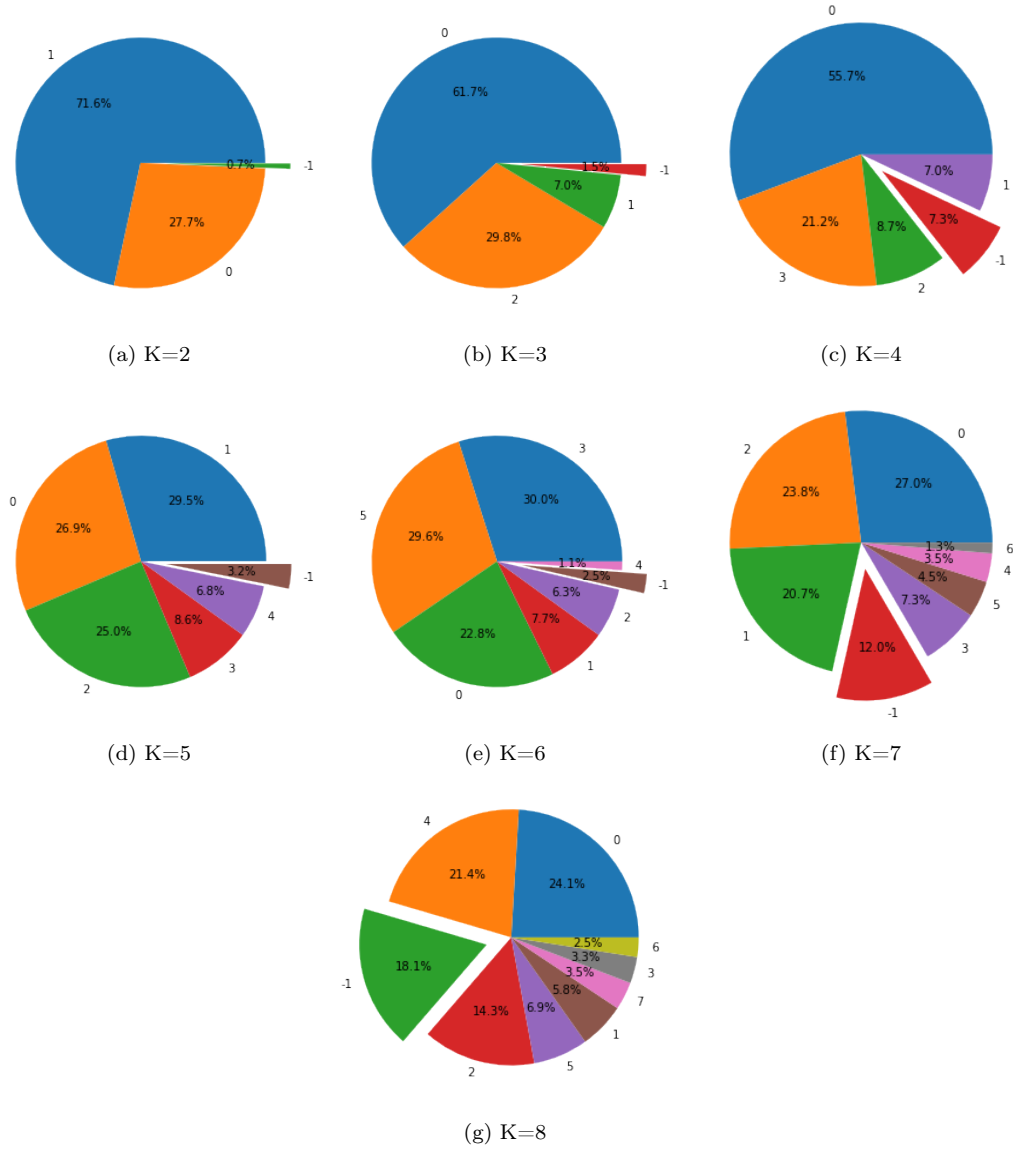


Figure 6-9: Computed SWT features of ECG records data were clustered using the consensus clustering method. Clusters' size ratios for the given number of clusters ( $K$ ) were computed

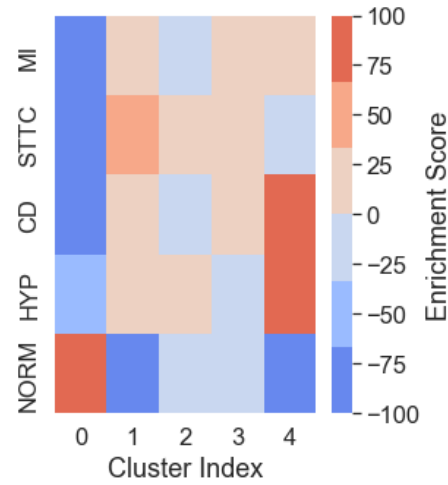


Figure 6-10: Enrichment scores of individual diagnosis classes in each cluster. high positive values (red color) indicates enrichment and low negative values (blue color) shows low presence.

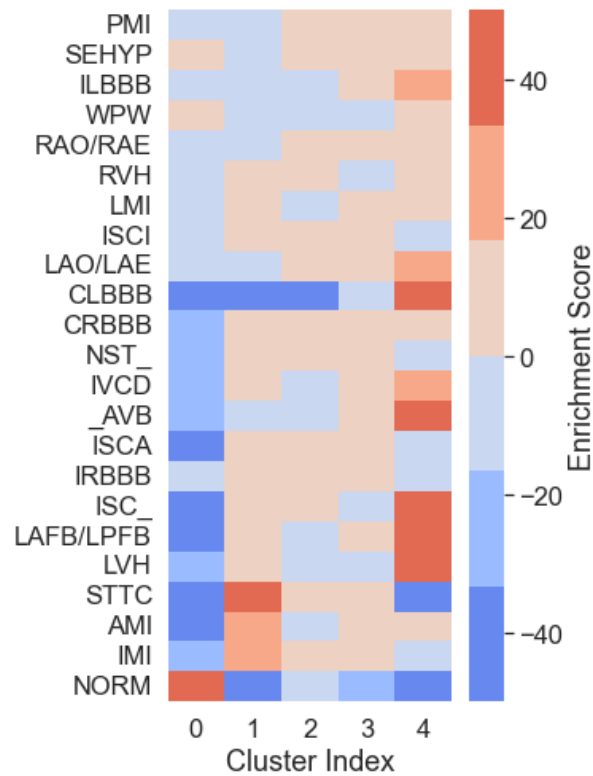


Figure 6-11: Enrichment scores of individual diagnosis subclasses in each cluster. high positive values (red color) indicates enrichment and low negative values (blue color) shows low presence.

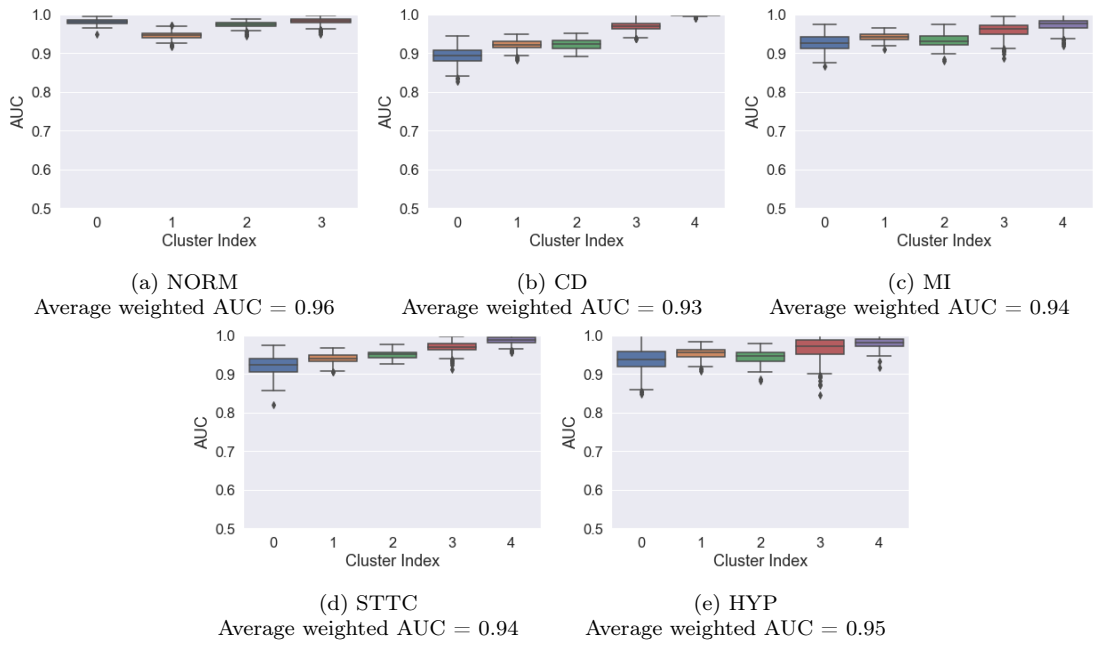


Figure 6-12: Extracted SWT features data of ECG records were clustered using the consensus clustering method. A One-vs-All Random Forests model was employed in a bootstrapping fashion to classify diagnosis classes: a) NORM, b) CD, c) MI, d) STTC, e) HYP, and the distribution of AUCs of individual diagnosis classes in each cluster were computed.



# Application Showcase: Risk of bradycardia during hypoglycemia in patients with type 2 diabetes

## 7.1 Introduction

Hypoglycemia is a condition where glucose level drops from the normal range. In patients with type 2 diabetes (T2DM), hypoglycemia may occur as an unfavorable complication of insulin treatment. Different studies indicate a strong relation between hypoglycemia events and cardiovascular mortality [95].

Clinical and empirical data suggest that hypoglycemia can lead to fatal arrhythmias through cardiac repolarisation alteration in subjects with diabetes [96, 97, 98]. Arrhythmia refers to irregular heartbeat series and is one of the main CV risk factors. In particular, Chow et al. [99] showed a significant increase in the frequency of Bradyarrhythmias in T2DM subjects, while Novodvorsky et al. [100] reported the same pattern in T1DM patients. Bradycardia is one of the common types of cardiac arrhythmias and can be defined as a sequence of heartrate below an absolute threshold. The mentioned studies hypothesize a causal relationship between hypoglycemia and an increased risk of Bradycardia events in such patients.

This section studied this hypothesis by examining the association of hypoglycemic glucose levels with bradycardia in insulin-treated patients with type 2 diabetes and chronic kidney disease, a high-risk population for hypoglycemic events.

Part of the work presented in this chapter has also been published in a peer-reviewed, open-access scientific journal [101].

## 7.2 Data

The data provided by Uniklink Aachen consists of three main recordings of 85 patients measured over 7 days: RR intervals, interstitial glucose levels, and bradycardia events.

The RR intervals were extracted from simultaneous 12-lead Holter-ECG records with a sampling rate of 1000 Hz, using the proprietary software Darwin 2 (Schillermed). The extracted RR intervals were used later for arrhythmia analysis to detect bradycardia episodes.

The interstitial glucose was recorded every minute, and 5 min averages were reported. It is measured using a sensor filament usually placed on the abdomen and not close to the insulin pump injection side.

It should be noted that during the monitoring, subjects continued their daily activities and diabetes management, while sensor calibrations were performed at least 3 times during the 7 days. Bradycardia, defined as  $> 4$  consecutive beats at  $RR < 1333$  ms (instantaneous heart rate  $< 45$  beats/min), occurred in 19 of 85 patients (22%); the number of episodes varied between 1 and 538 episodes with a mean of  $106.4 \pm 157.6$  episodes over 7 days.

Figure 7-1 depicts the distribution of number bradycardia events frequency in each patients. It can be seen that, in more than a third of the patients, there exist more than 50 bradyarrhythmias during 7 days. In another third, this frequency is less than 10. The occurrence time of bradycardias is another critical factor to be

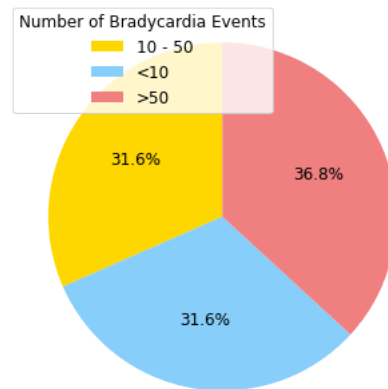


Figure 7-1: Distribution of bradycardia episodes per patients, based on three frequency levels: 1) less than 10 events, 2) from 10 to 50 events, and 3) more than 50 events [101].

considered. Figure 7-2 shows an apparent increase in such events after midnight and before morning. The same pattern can be seen shortly before noon, while the number

of bradycardias drop in the afternoon and evening. We studied this trend in detail in the results section.

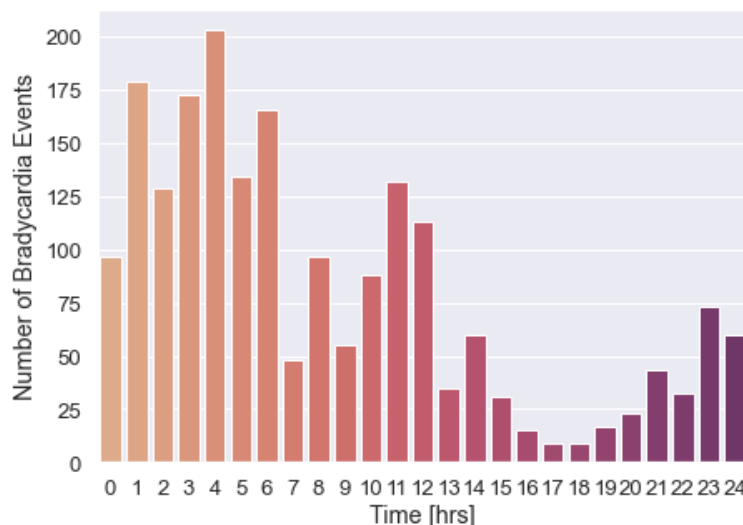


Figure 7-2: Number of bradycardia events in all patients at every hour of the day [101]

## 7.3 Heart Rate Variability Analysis

The heart rate is a measure of the distance between two consecutive R peaks in an ECG signal. It is important to note that a healthy heart is not characterized by a constant heart rate, but rather by complex and nonlinear fluctuations. This is due to the dynamic nature of the cardiac system and its intricate interactions with other physiological systems. However, in cases of cardiac abnormalities, such as arrhythmias or heart disease, the heart rate dynamics may be altered, resulting in an increase or decrease in the complexity of the heart rate oscillations. [34].

Heart rate variability (HRV) evaluates fluctuations in the time interval between consecutive heartbeats. This method quantifies the complexity of the cardiac system in both spatial and temporal dimensions through the analysis of heart rate parameters in the time and frequency domains [102]. Research has shown that alterations in resting HRV parameters can be indicative of cardiac pathologies or emotional states, making HRV a valuable tool for assessing the cardiac health status of individuals. In particular, HRV features can provide important information for identifying and characterizing cardiac disorders in patients [103]. HRV can be analyzed using various methods, including time-domain, frequency-domain and nonlinear techniques, in different time intervals. These intervals include 24 hours, short-term (approximately 5

minutes) and ultrashort-term (less than 5 minutes). The choice of time interval and analysis method will depend on the specific research question and the goals of the study [103].

In this thesis, we utilized a short-term HRV analysis approach due to the fact that the glucose sampling rate was set at 5 minutes.

### 7.3.1 HRV Time domain features

Time-domain features of HRV estimate the RR intervals' variability in a monitoring span (here 5 minutes). We used the RR intervals time-domain features used in this study, suggested by Shaffer and Ginsberg [103]. We also defined instantaneous heart rate time-domain parameters using the features described in section 4.1. These features are depicted in Table 7.1.

### 7.3.2 HRV Spectral domain features

Heart rate variability (HRV) can be quantified in the frequency domain by utilizing the Fourier transform to estimate the absolute or relative power within specific frequency bands. The Task Force of the European Society of Cardiology and the North American Society of Pacing and Electrophysiology (1996) [103] proposed four frequency bands to represent HR fluctuations in the frequency domain: the ultra-low-frequency (ULF) band (0.003 Hz), the very low frequency (VLF) band (0.0033–0.04 Hz), the low frequency (LF) band (0.04–0.15 Hz), and the high frequency (HF) or respiratory band (0.15–0.40 Hz). However, it should be noted that the ULF band requires a recording interval of at least 24 hours [104], and thus, it was not utilized in this current research study.

Table 7.2 shows the parameters defined in VLF, LF, and HF bands. Performing Fourier transform on a time series requires an invariant sampling rate, while the RR intervals data by nature is unequally spaced sampled. Hence, before HRV frequency-domain extraction, the data is interpolated using a cubic polynomial on a 4 Hz constant sampling frequency-time span.

### 7.3.3 HRV Nonlinear features

The measurement of RR intervals, or the time intervals between consecutive heartbeats, has been found to display intricate and unpredictable dynamics. In order to evaluate the level of randomness and nonlinearity present within a heart rate time

| Feature     | Unit | Description                                                         |
|-------------|------|---------------------------------------------------------------------|
| SDNN        | ms   | Standard deviation of RR intervals                                  |
| RMSSD       | ms   | Root mean square of consecutive RR interval differences             |
| pNN50       | %    | Percentage of consecutive RR intervals that vary by more than 50 ms |
| pNN20       | %    | Percentage of consecutive RR intervals that vary by more than 20 ms |
| HR Mean     | bpm  | The mean value of heart rates                                       |
| HR Median   | bpm  | The median value of heart rates                                     |
| HR Max      | bpm  | The highest value of heart rates                                    |
| HR Min      | bpm  | The lowest value of heart rates                                     |
| HR P2P      | bpm  | The difference of the highest and lowest values heart rates         |
| HR Kurtosis |      | The kurtosis value of heart rates                                   |
| HR Skewness |      | The skewness value of heart rates                                   |
| HR MAD      | bpm  | The median value of absolute changes of heart rates                 |
| HR ZCR      |      | The normalized zero-crossing rate of heart rates                    |

Table 7.1: HRV time-domain features

series, nonlinear parameters such as fractal dimension and entropy can be utilized. These nonlinear parameters provide a comprehensive analysis of the complexity and irregularity of the RR interval measurements, providing valuable insights into the underlying cardiovascular physiology. Table 7.3 shows the employed nonlinear features using entropy analysis and the Poincaré plot.

The Poincaré plot, a specific type of recurrence plot, is utilized in the estimation of self-similarity within dynamic systems. This methodology has been extensively employed in the analysis of heart rate variability (HRV), as it is a scatter plot representation of the R–R intervals in relation to their preceding interval ( $RR_t$  vs.  $RR_{t-1}$ ). The Poincaré plot is a widely accepted technique in the field of HRV analysis, as demonstrated by its inclusion in various studies and research articles such as [104].

A Poincaré plot can be explored by fitting an ellipse to the plotted points, and the

| Feature       | Unit            | Description                                                           |
|---------------|-----------------|-----------------------------------------------------------------------|
| VLF power     | ms <sup>2</sup> | Absolute power of the very-low-frequency band (0.0033–0.04 Hz)        |
| LF power      | ms <sup>2</sup> | Absolute power of the low-frequency band (0.04–0.15 Hz)               |
| HF power      | ms <sup>2</sup> | Absolute power of the high-frequency band (0.15–0.4 Hz)               |
| Total power   | ms <sup>2</sup> | Absolute power of the very-low to high-frequency band (0.0033–0.4 Hz) |
| LF normalized | %               | relative LF power to the sum of LF and HF power                       |
| HF normalized | %               | relative HF power to the sum of LF and HF power                       |
| LF peak       | Hz              | Peak frequency of the low-frequency band (0.04–0.15 Hz)               |
| HF peak       | Hz              | Peak frequency of the high-frequency band (0.15–0.4 Hz)               |
| LF/HF         | %               | Ratio of LF-to-HF power                                               |

Table 7.2: HRV frequency domain features

corresponding area and diameters of the ellipse can be extracted as nonlinear features. The area represents total variability, the length of the fitted ellipse measures short-term and long-term variability, while the width quantifies the short-term HRV [104]. For the entropy analysis, we used sample entropy of 2-dimensional embedded space (See Sect. 4.3.1), and permutation entropy of 3-dimensional embedded space with one step lag (See Sect. 4.3.2)

## 7.4 Data Preprocessing and Feature Extraction

### 7.4.1 Windowing

As mentioned in the previous section, due to 5 minutes sampling rate of glucose measurement in the data, we performed a short-term (5 minutes) HRV Analysis. Therefore, the feature extraction process was performed in a moving window fashion, as described in section 3.3. The length of the window was set to 5 minutes with a

| Feature | Unit | Description                                                                          |
|---------|------|--------------------------------------------------------------------------------------|
| S       | ms   | Area of the fitted ellipse on the Poincaré plot                                      |
| SD1     | ms   | Width of the fitted ellipse on the Poincaré plot                                     |
| SD2     | ms   | Length of the fitted ellipse on the Poincaré plot                                    |
| SD1/SD2 |      | Ratio of SD1-to-SD2                                                                  |
| SampEnt |      | Sample entropy, which measures the irregularity and randomness of a time series      |
| PermEnt |      | Permutation entropy, which measures the irregularity and randomness of a time series |

Table 7.3: HRV nonlinear features

similar step size, resulting in no overlaps between the windows. To lower the impact of high variance in the number of bradycardias, we considered all bradyarrhythmias in a 5 minutes window as one Brady episode. We extracted HRV features (See Sect. 7.3) and the measured glucose value within each window. Additionally, to study the daytime effect, we calculated a continuous form of 24 hours clock by decomposing the time into a Sin and Cosine function with 24 hours periodicity. The required denoising and missing value imputation prior to feature calculation were conducted in the workflow of windowing the RR intervals and glucose values, as explained in the following sections.

### 7.4.2 Filtering

We started with cleaning and removing the outliers in the RR intervals measurement in the windowing process. First, we removed the RR intervals out of the physiological accepted range (200 ms - 2500 ms). Next, we removed the ectopic or artifact RR intervals using a recursive filtering procedure [105]. Ectopic beats refer to premature ventricular and atrial contraction. Although ectopic beats are considered normal and do not indicate cardiac pathology, they can affect the analysis of the heart rate variability and other types of arrhythmias. The glucose values outliers ( $<10$  mg/dL and  $>700$  mg/dL) were also removed during the windowing procedure.

### 7.4.3 Missing value imputation

Missing value imputation is another crucial step before performing HRV analysis. It is shown that missing values can cause misleading results in HRV estimation [106]. The missing values are present in the data either due to monitoring error (detachment of the sensors) or after removing the outliers. As explained in section 3.2, we implemented a linear interpolation to handle the missing RR intervals (See Sect. 3.2), where the number of missing values between two available values is less than 5. We dropped the window in the presence of long missing values as it can immensely affect the results.

## 7.5 Results

### 7.5.1 Association between glucose level and bradycardia events

Given the hypothesis that extreme glucose levels may induce severe bradycardia in patients with diabetes, we analyzed the connection between the number of bradycardic events and the absolute glucose value. In all the patients with bradyarrhythmias, the majority of bradycardias occurred at normal glucose levels above 70 mg/dL and below 180 mg/dL, while only 4% of all bradycardia episodes arose at glucose values below 70 mg/dL (See Fig. 7-3).

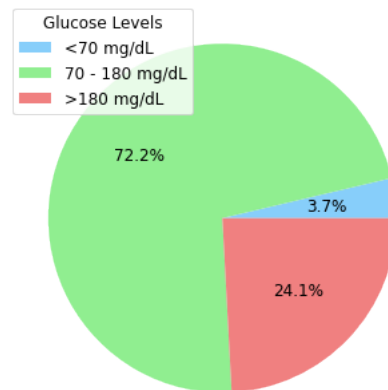


Figure 7-3: Percentage of bradycardia events at absolute glucose levels

Further examinations of glucose profiles in the patients with bradycardia episodes indicated a high variability of glucose values over 7 days with significantly different mean glucose values and very dissimilar glucose profiles (Fig. 7-4a). For illustration,

figure 7-4b displays the glucose profiles of 2 patients. Patient 2 has a mean glucose value of 197.2 mg/dL over 7 days, and despite its high glucose volatility, it never goes below 70 mg/dl. On the other hand, patient 11 underwent several hypoglycemia episodes with glucose measurements  $< 70$  mg/dl.

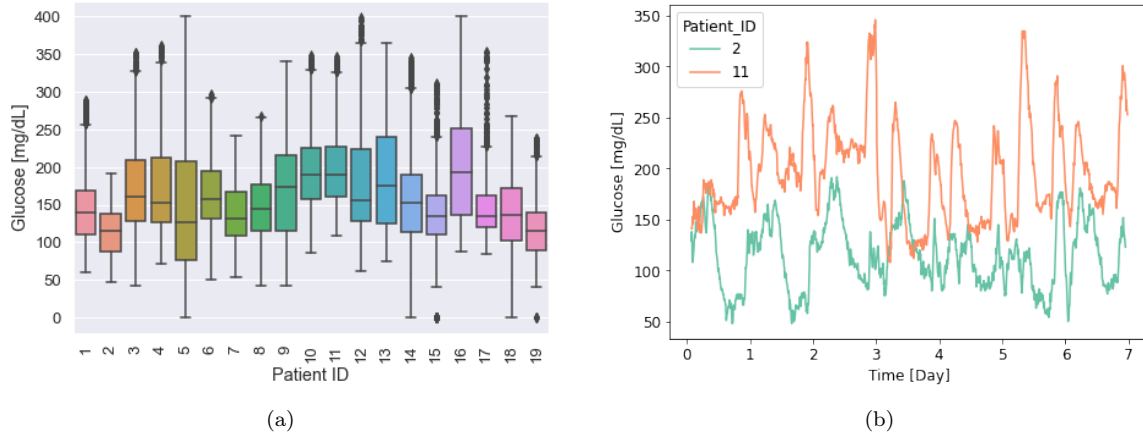


Figure 7-4: a) Absolute glucose range in each patient with bradycardia episodes, b) Example of glucose profile of two patients with bradycardia episodes, which shows a clear difference in glucose level range in these two patients [101]

Based on these results, we hypothesized that relative rather than absolute glucose values might be more applicable for analyzing the bradycardias in relation to glucose levels in these patients. Accordingly, we categorized glucose levels in tertiles of each patient. As expected, due to the different glucose profiles in the patients, we can see an overlap in the glucose level distribution at each relative tertile (See fig. 7-5). The relative categorization of glucose levels shows that more than 45% of bradycardias occurred in the low glucose level tertile, while around 28% of these events appeared at the high glucose level tertile (See fig. 7-6)

We executed a Fisher's Exact test (See Sect. 5.1.2) in a bootstrap sampling manner to examine whether this unequal occurrence of bradycardia events in the relative glucose tertiles is statistically significant. The null hypothesis is that ratio of bradycardias number in each tertile to the whole number of events is similar to the glucose levels ratio in each tertile ( $1/3$ ), while the alternative hypothesis is that the bradycardia frequency ratio is significantly different from the glucose levels ratio.

We selected selected and analyzed iteratively using a statistical approach (100 times). The p-value for each tertile was calculated by multiplying the expression direction by the ratio of bradycardia events in each tertile and the corresponding glucose ratio. The expression direction was determined by evaluating the sign of the difference between the ratio of bradycardia events and the glucose ratio in each

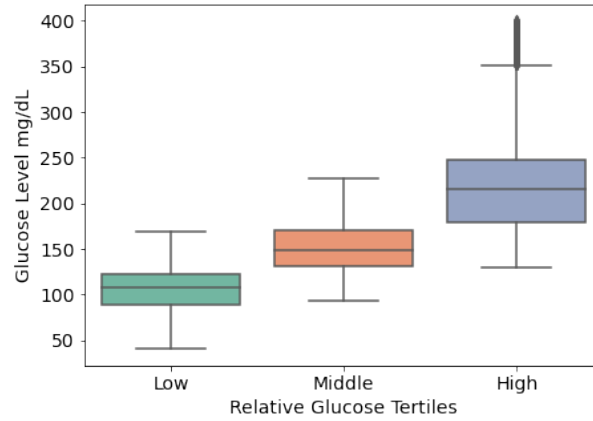


Figure 7-5: Distribution of glucose levels in each relative individualized glucose tertile [101]

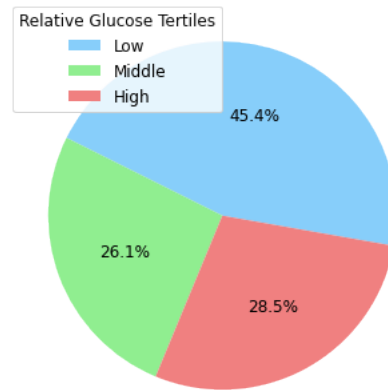


Figure 7-6: Percentage of bradycardia events at different glucose tertiles [101]

tertile. A positive value indicates the over-expression of bradycardia events in the tertile, while a negative value indicates the under-expression of bradycardia events in the tertile.

As anticipated, our results revealed that bradycardia events were significantly pronounced in the low relative glucose tertile. However, in most iterations, we observed a low occurrence of bradyarrhythmias in the normal and high glucose tertiles (See fig. 7-7).

In the next step, we investigated the association between glucose level and the occurrence time of bradycardia events. Figure 7-8 shows a clear correlation between the mean heart rate and the mean value of glucose level of all patients over the 24 hours time course with a 95% confidence interval. We also showed in figure 7-2 that the frequency of bradycardia events is a function of time. The same pattern can be noted in the alteration of relative glucose tertiles distributions over 24 hours, where

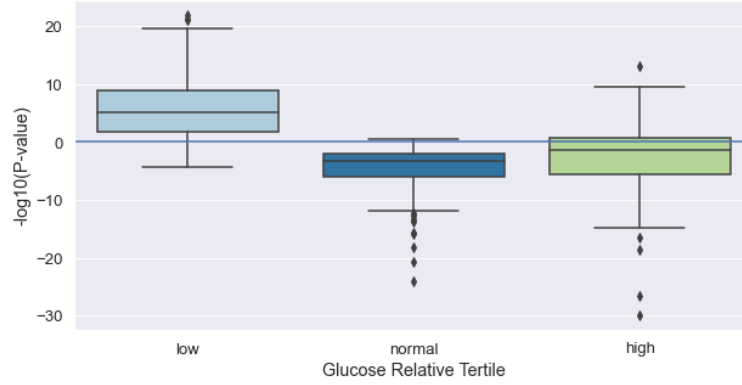


Figure 7-7: Distribution of p-values multiplied by the expression direction of bradycardia events (enriched or not enriched) in each relative glucose tertile obtained by Fisher's Exact test in a bootstrap sampling strategy

there is an enrichment of high glucose tertile before midnight and over-expression of low glucose tertile after midnight (See Fig. 7-9).

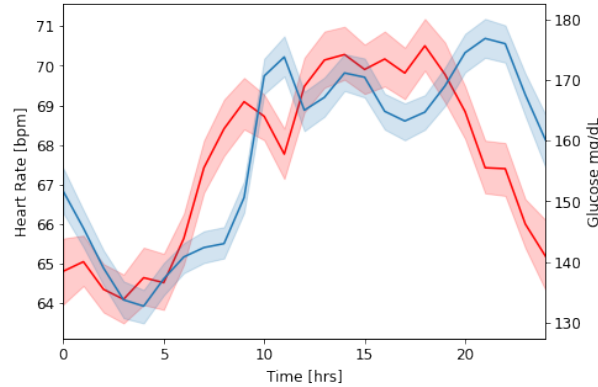


Figure 7-8: Red line is the mean value of heart rate of all patients during a complete day with 95% confidence interval; Blue line is the mean value of glucose level in all patients during a complete day with 95% confidence interval [101]

Combining all three factors (time, glucose relative tertiles, and bradycardias frequency), we observe that most bradycardias occur after midnight (from 12:00 AM to 09:00 AM) at glucose levels in the low individual tertile (See Fig. 7-10). In contrast, there is an enrichment of bradycardias in the highest glucose tertile at night (from 6:00 PM to 03:00 AM).

To quantify this difference, we conducted a Fisher's Exact test (See Sect. 5.1.2) to study whether the ratio of bradycardias in each glucose tertile is identical to the ratio of overall time intervals in the corresponding tertile at each 3 hours time period (Null hypothesis) or not (Alternative hypothesis). The statistical test was performed in a bootstrap sampling fashion with 100 times iteration, and the p-value multiplied

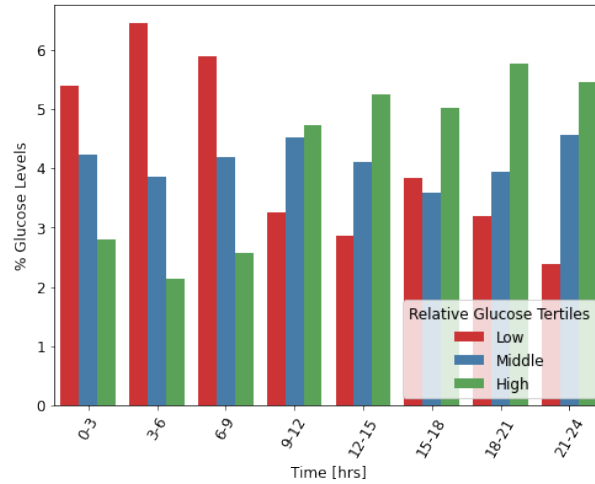


Figure 7-9: Percentage of data points at different glucose tertiles in 3 hours' time intervals [101]

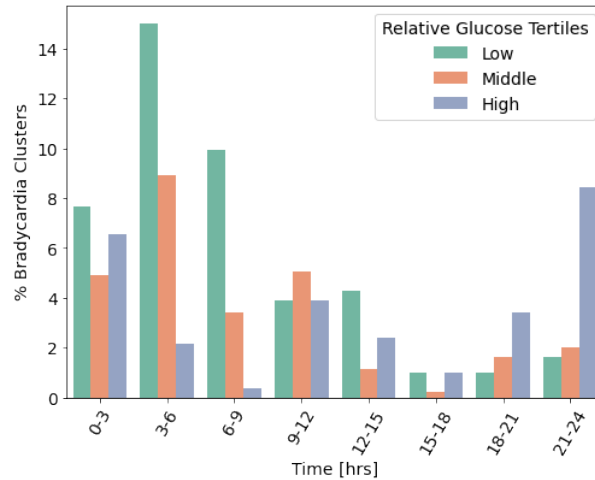


Figure 7-10: Percentage of bradycardia events at different glucose tertiles in 3 hours time intervals [101]

by the expression direction was computed in each iteration. Figure 7-11 shows the obtained result for low and high relative glucose tertiles at each time interval. We can see an enrichment of Bradycardia events at low relative glucose levels between 06:00 and 09:00, as well as 12:00 and 15:00, which indicated a higher risk of bradycardias event at these time intervals during a drop in glucose level. At other time intervals, we do not see any significant effect of glucose levels on the bradyarrhythmia occurrence. On the other hand, there is a high risk of such episodes between 21:00 and 03:00 during high glucose levels. However, this pattern alters quickly after 3, and the risk remains mostly at low glucose levels.

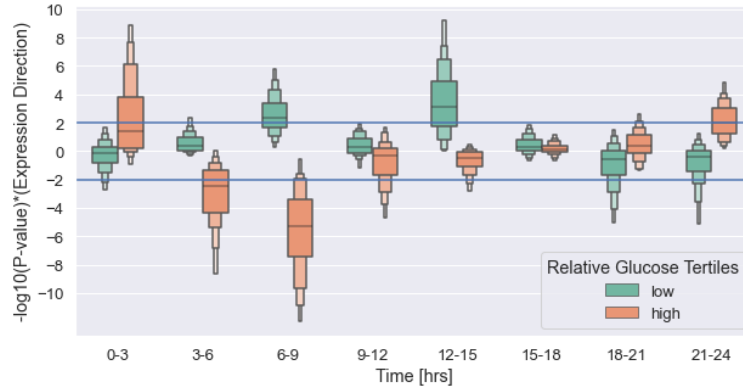


Figure 7-11: Distribution of p-values multiplied by the expression direction of bradycardia events (enriched or not enriched) at low and high relative glucose tertiles in 3 hours time intervals obtained by Fisher’s Exact test in a bootstrap sampling strategy [101]

### 7.5.2 Association between HRV features and Bradycardia events

The association between HRV features and the bradycardia events is another question we tried to answer. In the first place, we computed the pairwise correlation of extracted features in all patients (See Fig. 7-12), where a noticeable positive correlation between SDNN, RMSSD, spectral powers, and Poincaré plot parameters is shown (As stated in the literature too [103]). Such positive correlations can be seen between heartrate statistical features (Mean, Median, Max, and Min).

To estimate whether the HRV change in the course of bradycardia episodes, we performed a Wilcoxon Ransum significance test between the distribution of each HRV parameter at time windows containing bradycardia episodes and windows without such events. Nevertheless, the size of windows with bradyarrhythmias is much smaller than those in the absence of bradycardias (827 to 35415). Hence, to reduce the effect of disproportions distributions in the results, we used a bootstrap sampling technique in a loop with 100 times repetition, where small samples with a similar size (200 sample points) from each class were selected, and corresponding p-values were computed. Figure 7-13 shows the distribution of the calculated p-value of each feature multiplied by the expression direction (the sign of the difference between two groups’ mean value).

The result shows a statistically significant increase in SDRR, RMSSD, Low- and High-frequency power, as well as Poincaré plot parameters (S, SD1, SD2) during Bradycardia episodes. The elevation of Poincaré features is expected due to their correlation to RR interval short- and long-term volatility (SDRR and spectral powers). On the other side, as anticipated, a pronounced drop in Heart rate mean, median and

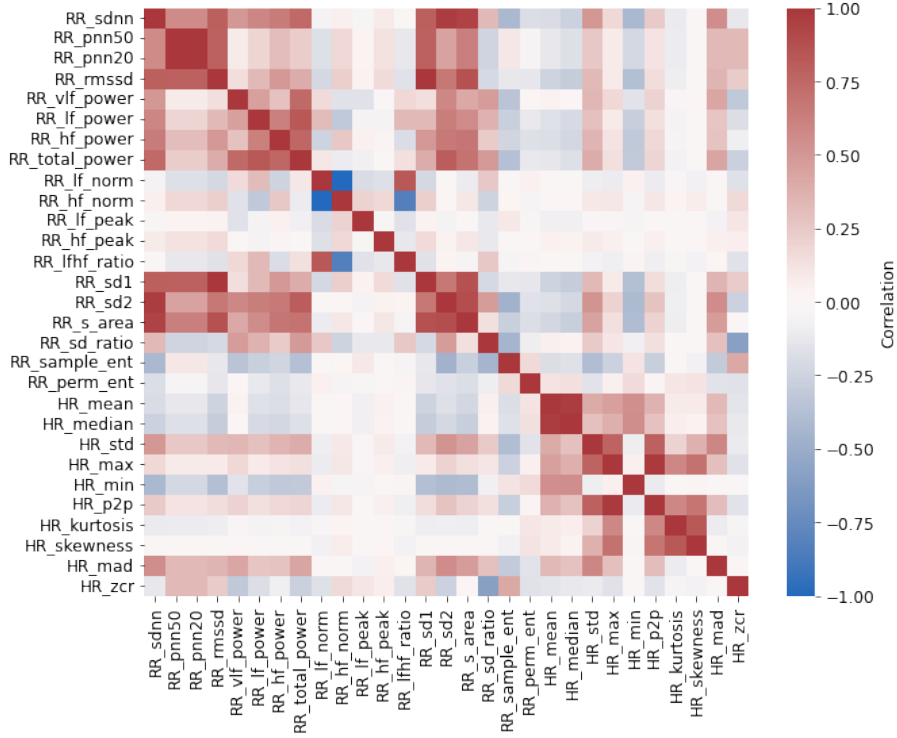


Figure 7-12: Heatmap of HRV extracted features pairwise correlation

minimum values can be noticed. Further, we can see a slight decrease in sample and permutation entropies and heart rate max value, together with an elevation in VLF power in the course of bradyarrhythmias.

### 7.5.3 Forecasting the Bradycardia events

Considering the findings in the sections above and the relationship between the extracted HRV features, glucose value, and day-time, we explored whether these parameters can be used for forecasting bradycardia events in diabetic patients. For this purpose, we created a classification model where each feature window was considered a data point. To build the classifier, we identified and labeled all the windows 5 minutes prior to the bradycardias as risk zone and labeled all the windows without and far from such events as normal zones (See Fig. 7-14). We removed all the windows containing at least one bradycardia event, since predicting such windows as the target class does not have a forecasting advantage. In short, the model aims to forecast the bradyarrhythmias by classifying the risk zones from the normal zones.

Before proceeding to the classification part, we normalized each patient's data points using Z-score (See Sect. 3.1) and then pooled all the patients' data. Due to

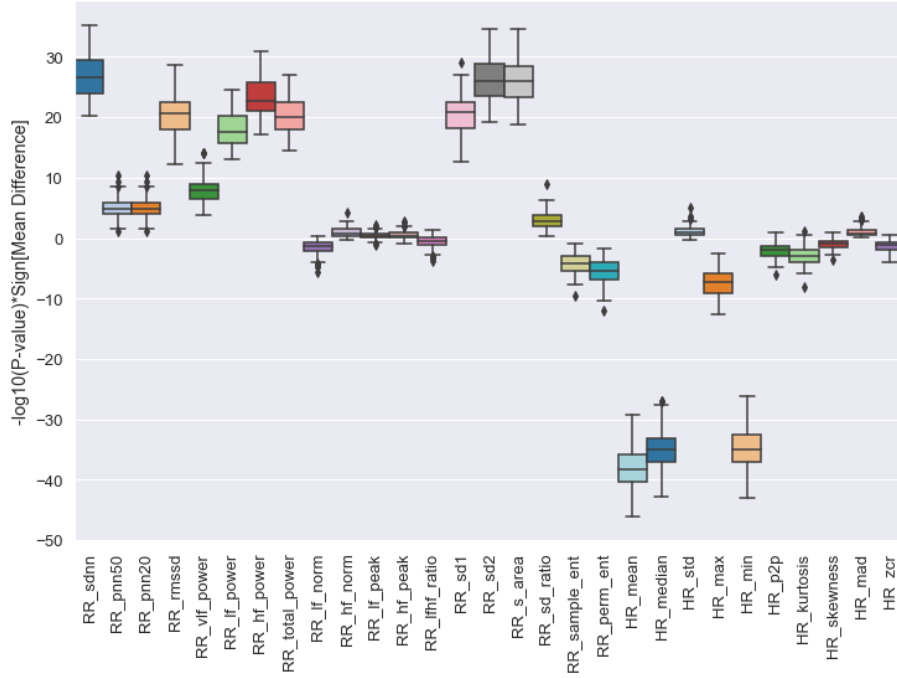


Figure 7-13: A statistical significance test (see Sect. 7.5.2) was performed to test whether HRV features values were able to differentiate the data points with bradycardia events from time points without bradyarrhythmia. In a bootstrapping approach, the distribution of P-values was computed and multiplied by the difference between the two groups' mean values for each HRV feature.

the rare occurrence of arrhythmia episodes in these patients, the labeling approach leads to imbalanced classes in the extracted feature data (28873 normal points and 546 risk points in all the patients). In order to handle the imbalanced classes in the data, a bootstrap technique was employed, where at each run, we took a sample without replacement with similar size of risk class from the normal group. Afterward, we applied a classifier with 5-fold stratified cross-validation (See Sect. 5.3) on the merged sampled normal group and fixed risk group, resulting in a mean prediction score for each bootstrap run. This process was repeated 50 times, and the mean and standard deviation of the prediction scores were calculated.

To do the classification, we utilized four models: Logistic Regression with L1 regularization, Logistic regression with L2 regularization (See Sect. 5.2.1), Support Vector Machines with the radial kernel (See Sect. 5.2.2), and Random Forests (See Sect. 5.2.3). After tuning and finding the optimal parameters and running the described process using each classifier, we discovered that the random forests model showed better performance compared to the others. Figure 7-15 shows the mean ROC curve of all four models. It can be seen that the computed mean AUC of SVM is a bit better than logit models (0.90 to 0.87), while random forest demonstrates the

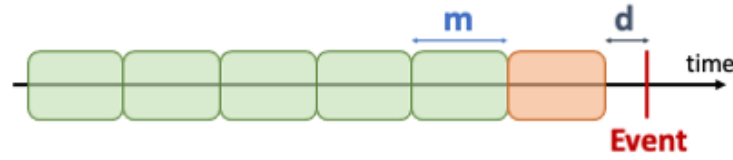


Figure 7-14: Bradycardia risk zone and normal zone identification: Green windows, with the length  $m$ , are the normal data points, and the orange one is the risk data point that occurred  $d$  minutes before the bradycardia event.

best performance (0.93).

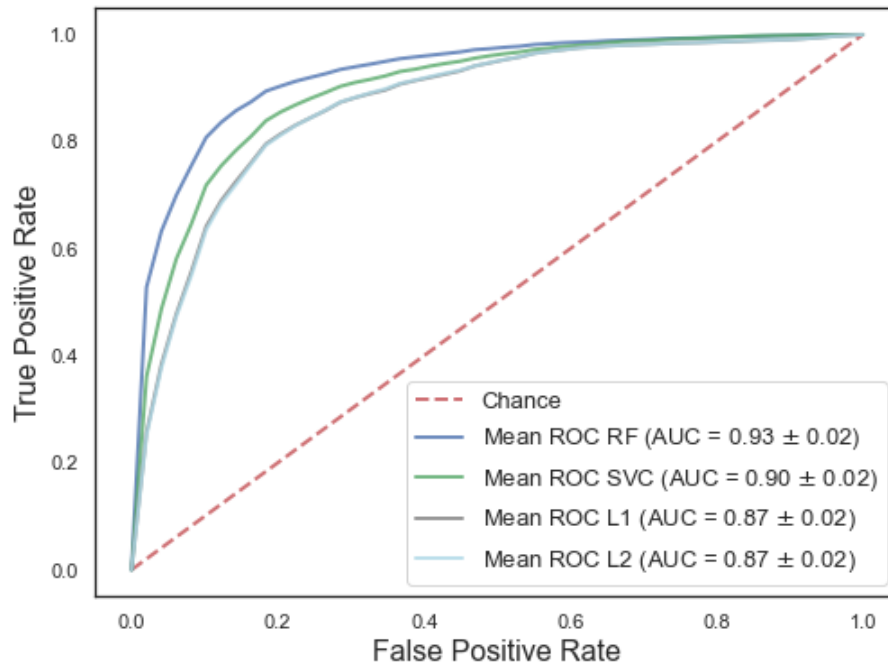


Figure 7-15: Four classification models were used to predict the risk of Bradycardia in diabetic patients in a bootstrapping form (see Sect. 7.5.3). The mean ROC curves and corresponding AUC were calculated for each model.

The distribution of the classification metrics obtained using random forests from bootstrap sampling, including Accuracy, Area Under the Curve (AUC), Precision, Recall, F1 score, are displayed in figure 7-16.

As it can be seen, all the measures show an acceptable to the perfect performance of the model in identifying risk zones (from 0.7 to 1). Next, the effect of proximity to the bradycardia episodes on the classification performance was investigated. We repeated the classification process using Random Forests by changing the temporal distance of the risk zones to 1, 15, and 60 minutes before the events, respectively. Table 7.4

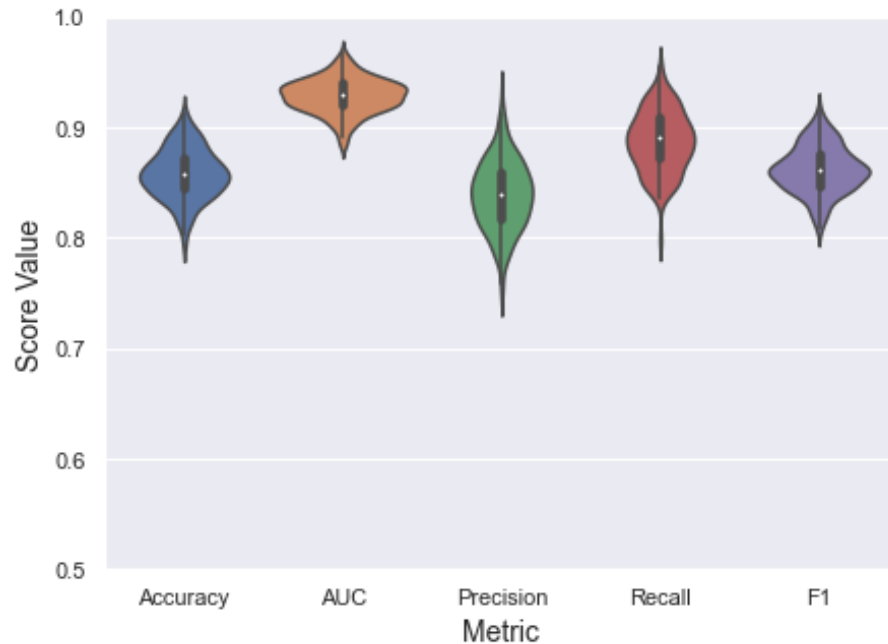


Figure 7-16: The distribution of classification measures of bradycardia risk prediction using Random Forests.

shows the classification metrics of the dataset based on the risk zones time-to-event. The results indicate no significant change in the model performance by keeping the distance of the risk zones to 15 minutes. However, a notable decline in the measure by classifying the risk zones 1 hour before the bradycardias can be noticed. This finding suggests that the transition to the occurrence starts merely some minutes earlier in most arrhythmias, and an accurate forecast can be done within this period.

| Time to Event | Metric          |                 |                 |                 |                 |
|---------------|-----------------|-----------------|-----------------|-----------------|-----------------|
|               | Accuracy        | AUC             | Precision       | Recall          | F1              |
| 1 minute      | $0.86 \pm 0.02$ | $0.93 \pm 0.02$ | $0.84 \pm 0.03$ | $0.87 \pm 0.04$ | $0.85 \pm 0.02$ |
| 5 minutes     | $0.85 \pm 0.03$ | $0.92 \pm 0.02$ | $0.84 \pm 0.03$ | $0.87 \pm 0.03$ | $0.85 \pm 0.02$ |
| 15 minutes    | $0.85 \pm 0.03$ | $0.91 \pm 0.02$ | $0.83 \pm 0.04$ | $0.86 \pm 0.05$ | $0.85 \pm 0.03$ |
| 60 minutes    | $0.79 \pm 0.03$ | $0.83 \pm 0.03$ | $0.79 \pm 0.04$ | $0.8 \pm 0.05$  | $0.8 \pm 0.03$  |

Table 7.4: Risk zones were pinpointed based on the proximity to the bradycardia events. Four time distances to bradyarrhythmias were defined, and the corresponding Mean and standard deviation of classification metrics of risk prediction were calculated.

For a practical alarm system in a clinical setting, two factors are essential: The sensitivity of the model (recall), where the maximum number of events forecast is a matter of interest, while the model must signal false alarms as few as possible. Although our model shows good accuracy and acceptable recall, the precision might not satisfy the second condition since, for such an imbalanced dataset, any precision

value below 1 might not be applicable for a warning system as it leads to several false alarms. Figure 7-17 displays the Recall-Precision curve with a mean AUC of 0.93; however, the mean precision value is between 0.6 and 1. In other words, to ensure a high precision (few false positives), the sensitivity must be lowered. In the data preparation step, we normalized patients' data individually as a required step before classification.

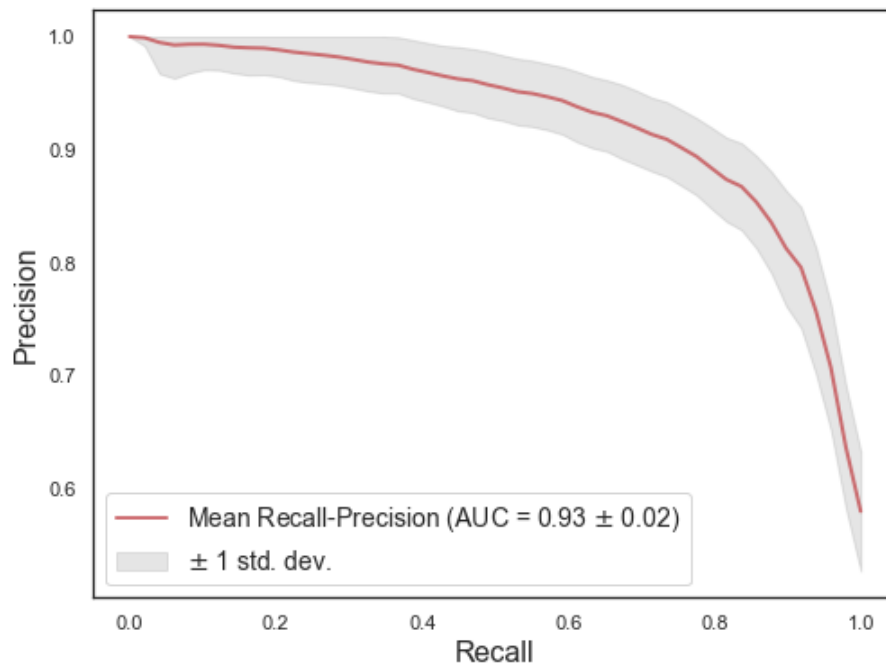


Figure 7-17: The recall-precision curve of Random Forests classifier trained on normalized features

Nevertheless, bradycardia arrhythmia is annotated in the data based on an absolute definition ( $hr < 45$  bpm). Hence, we presupposed that skipping the normalization step might improve the model's performance, and since the Random Forests does not require normalized data, it allowed us to do so. Consequently, we conducted the process mentioned earlier using Random Forests to classify risk zones defined 5 minutes prior to bradycardias. The obtained results show a slight increase in the metrics, particularly precision, by 2 percent (See Table 7.5). The recall precision curve also demonstrates better performance and permits increasing the precision with a more acceptable recall level (See Fig. 7-18). Nonetheless, whether this improvement makes the model beneficial in a clinical environment is a matter of decision by clinicians.

Finally, we explored the significance of each feature in separating the risk zones from the normal zones using the Random Forest classifier. Figure 7-19 reveals the distribution of each feature's importance scores, where we see that heart rate lowest,

| Normalized Data | Metric          |                 |                 |                 |                 |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
|                 | Accuracy        | AUC             | Precision       | Recall          | F1              |
| No              | $0.87 \pm 0.02$ | $0.94 \pm 0.02$ | $0.87 \pm 0.03$ | $0.88 \pm 0.02$ | $0.87 \pm 0.03$ |
| Yes             | $0.85 \pm 0.03$ | $0.93 \pm 0.02$ | $0.84 \pm 0.03$ | $0.87 \pm 0.03$ | $0.85 \pm 0.03$ |

Table 7.5: The mean and standard deviation of performance measures of the Random Forests classifier trained on normalized data and non-normalized.

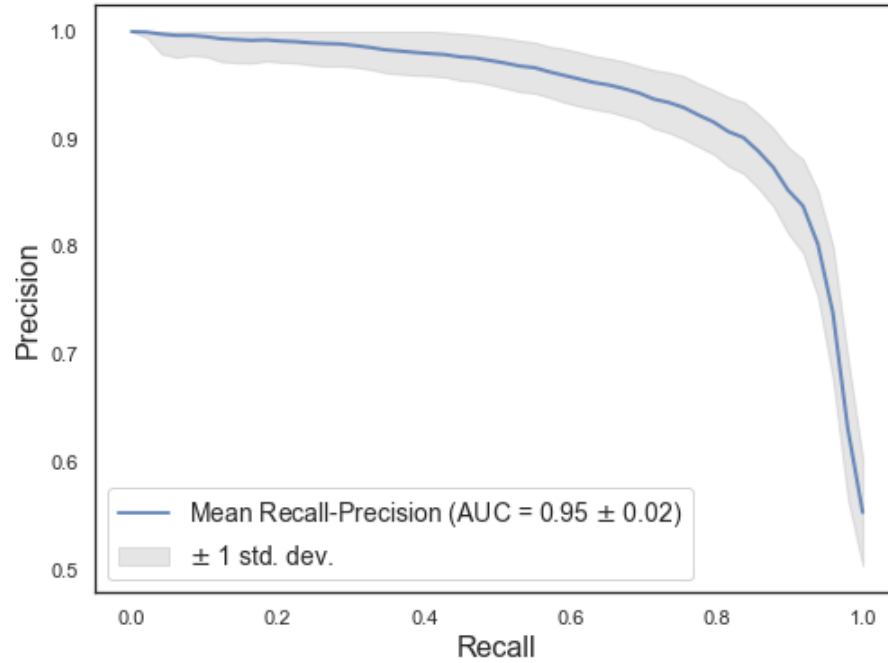


Figure 7-18: The recall-precision curve of Random Forests classifier trained on non-normalized features [101]

median, and mean values are the most critical parameters in classifying the risk windows. Furthermore, following them, we have the Poincaré SD2, sample entropy, LF power, total power, SDNN, and daily time clock.

## 7.6 Discussion

The current research aimed to examine the correlation between heart rate parameters and glucose levels with bradycardia events in a sample of 19 insulin-treated individuals with type 2 diabetes. Previous research has established a connection between instances of bradycardia and absolute hypoglycemia levels, defined as a glucose value below 70 mg/dL [99, 100]. The present investigation sought to further explore this association in a population of insulin-treated patients with type 2 diabetes.

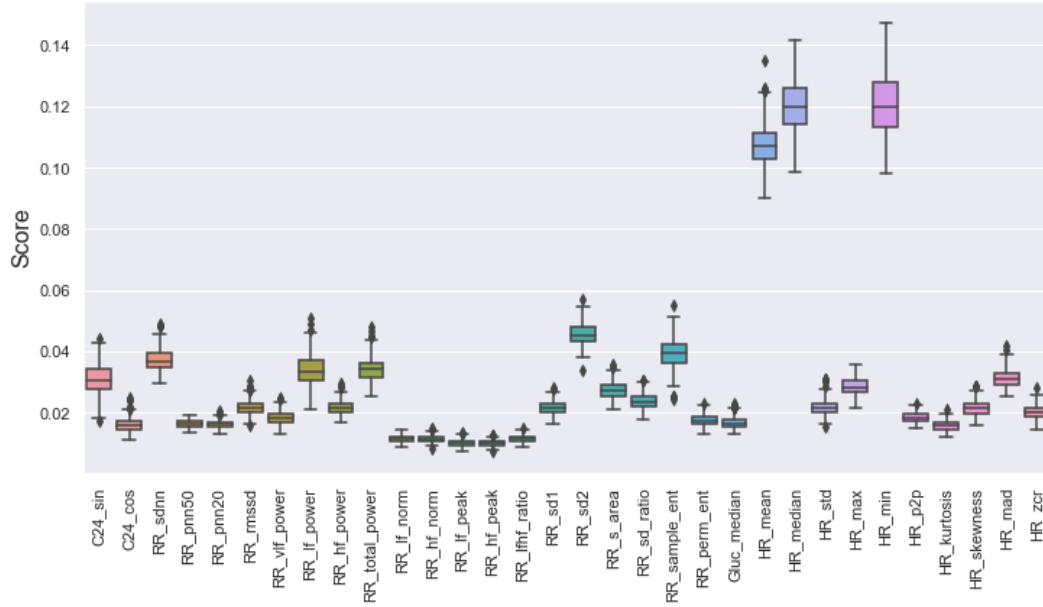


Figure 7-19: A Random Forests classifier was utilized to predict the bradycardia risk in diabetic patients in a bootstrapping manner. The model returns a score showing the percentage of each feature’s involvement in the prediction. Accordingly, the distribution of each feature’s score was obtained [101].

Upon conducting a thorough exploration of our data, we found that a mere 4% of episodes of bradyarrhythmias occurred in individuals with absolute glucose levels 70 mg/dL and 28% of these episodes occurred in individuals with absolute hyperglycemia levels ( $> 180$  mg/dL). Given the high inter-glucose variance present in these patients, we hypothesized that a relative definition of hypoglycemia may be more appropriate in identifying the relationship between extreme glucose levels and bradyarrhythmias. To test this hypothesis, we grouped each patient’s glucose levels into three relative tertiles and found that over 45% of bradycardias occurred in the lowest individual glucose tertile. Due to the small size of our population, we conducted a statistical bootstrap analysis to confirm the enrichment of bradycardia episodes in relatively low glucose levels.

Furthermore, we examined the effect of time on the occurrence of such events using similar statistical methods. Our results indicated a high accumulation of bradycardia episodes after midnight and in the morning, corresponding with declining relative glucose levels. Specifically, a significant enrichment of these events was observed between 06:00 and 09:00. Interestingly, we also observed an enrichment of bradycardia episodes at relatively low glucose tertile levels between 12:00 and 15:00, despite the overall increase in glucose levels during this time period. On the other hand, bradycardia episodes that occurred between 21:00 and 03:00 were pronounced notably at

relatively high glucose levels.

To the best of our knowledge, these results suggest for the first time that the threshold for hypo- and hyperglycemia-associated bradycardias correspondingly depends on individual glucose values and glucose control.

In addition, we examined the expression of various heart rate variability (HRV) features in the time-domain, frequency-domain, and nonlinear domain. Utilizing a statistical bootstrap test, we found a significant overexpression of RR interval standard deviation and spectral power, as well as Poincaré plot parameters. These findings indicate a deviation from typical HRV patterns during bradycardia events. Furthermore, we observed a statistically significant decrease in the mean, median, and minimum heart rate values, which is a predictable outcome in cases of bradycardia. Additionally, a decline in entropy parameters was observed, further supporting the presence of abnormal HRV patterns during these events.

Next, we aimed to evaluate the utility of heart rate variability (HRV) parameters, glucose values, and day-time features in predicting instances of bradyarrhythmias. To accomplish this, we employed a classification schema to separate data points that were in proximity to these events (referred to as "risk zones") from the remaining data points (referred to as "normal zones"). The results of our analysis indicated that the Random Forests classifier achieved an acceptable area under the curve (AUC) of 0.94 in classifying the risk data points. However, due to the imbalanced distribution of the data, with a small number of risk zones compared to normal zones, the computed precision (0.87) and sensitivity (0.88) of the model were found to be suboptimal for practical application. Specifically, increasing precision led to a deterioration in sensitivity and an increase in false alarms. Additionally, our findings revealed that the risk areas could be detected up to 15 minutes prior to the onset of bradycardia events.

The results of our study indicate that several parameters are crucial in the classification of transition into arrhythmia episodes. Specifically, we found that heart rate mean and lowest values, as well as RR intervals standard deviation, length of an ellipse fitted on Poincaré plot (SD2), sample entropy, and frequency power were highly dominant in identifying these transitions. Previous research has suggested that variance and autocorrelation are important parameters in detecting critical transitions in dynamic systems [24]. Our findings support this assumption, as we observed that standard deviation is derived from variance and SD2 represents the autocorrelation coefficient. Overall, these results highlight the importance of these parameters in the identification of arrhythmia episodes.



# Application Showcase: Critical Transition in Septic Patients

## 8.1 Introduction

Specific physiological systems show nonlinear dynamics that undergo sudden regime shifts at breaking points [107, 108]. In a medical setting, these shifts often exhibit instabilities in clinical states, e.g., disease-onset or disease stage transition [109]. Such phenomena have been studied mathematically using data-driven techniques to develop statistical indicators known as early-warning signals to model the dynamics of systems hovering shifts [24, 110, 111, 112, 113, 114, 115, 116]. Modeling such transitions is advantageous for several applications in systems medicine, such as monitoring health [117], forecasting disease-onset, and acquiring an improved knowledge of the underlying disease progression [118].

This chapter focus on sepsis, one of the primary causes of in-hospital mortality and costs the healthcare sector billions as a consequence of growing incidence rates and extended hospital stays [119, 120]. As stated in the Third International Consensus Definitions of Sepsis and Septic Shock (Sepsis-3), "sepsis is a life-threatening organ dysfunction caused by a dysregulated host response to infection," and "septic shock is a subset of sepsis in which underlying circulatory and cellular/metabolic abnormalities are profound enough to increase mortality substantially [121]."

The heterogeneity of infectious factors and the multiple accompanying comorbidities make accurate sepsis diagnosis challenging for physicians. Early, aggressive administration of antibiotics is vital, and delays in this treatment lead to increased mortality significantly [122, 123].

There exist several disease severity scores or early-warning signals, such as the Simplified Acute Physiology Score (SAPS II), the Acute Physiology and Chronic Health

Evaluation (APACHE II), the Sepsis-related Organ Failure Assessment Score (SOFA), the Modified Early Warning Score (MEWS), and the Simple Clinical Score [124]. These scores are acceptable predictors of general disease severity and mortality but unable to assess the risk of developing sepsis with reasonable accuracy and precision [124].

Multiple machine learning (ML) approaches were therefore introduced to predict sepsis onset [125, 126, 127]. Rothman et al. [128] employed structured information from electronic health records (EHRs) to identify sepsis on admission or to predict its onset during hospitalization. Ghosh et al. [129] presented a model integrating sequential contrast patterns with coupled hidden Markov models for septic shock prediction. Henry et al. [124] designed a targeted real-time early warning score (TREWScore) by training a Cox regression model to spot the patients at high risk of developing septic shock. These ML approaches showed the potential to improve diagnostic accuracy, while they involve time-consuming and domain-specific feature selection [130, 131]. This section proposed a strategy as a preprocessing step to select relevant data for further septic shock diagnosis analysis. Our strategy aimed to identify the critical transitions employing the concept of surprise loss (SL) introduced by Giacomini [132]. SL was initially presented in econometrics to evaluate forecast breakdown, i.e., instability in a time series model's forecasting ability. We believe that similar instabilities are present in clinical time series data due to medical intervention or disease progression. We employed SL to specify unstable regions in the time series. Besides, we introduced a critical-transition-based data-sampling approach to regions around critical transition.

The presented work in this chapter is also published in a peer-reviewed open-access scientific journal [37].

## 8.2 Data

We used the publicly available EHR database Medical Information Mart for Intensive Care (MIMIC)-III v1.4 [133], which includes multivariate time-series data of 46520 anonymized patients from 58976 separate ICU admission. We considered each ICU encounter as one distinct patient sample for ease of interpretation.

With the growth of pathophysiology and epidemiological over decades, rule-based retrospective annotation of sepsis syndrome has matured [134]. The latest characterization, generally known as Sepsis-3, stresses organ dysfunction as the primary outcome of the sepsis syndrome and suggests updating the International Classifica-

tion of Diseases (ICD) coding system [135, 136] (ICD-9: 995.92, 785.52; ICD-10: R65.20, R65.21). The organ dysfunction degree is graded commonly using the SOFA scoring system every 24 hours during a patient’s ICU stay [137, 138]. Due to the incompatibility of ICD-9 codes in the MIMIC-III dataset and the Sepsis-3 definition, we annotated the patient data based on the 2016 consensus definition. Figure 8-1 illustrates the schematic of the annotation framework, which was designed and coded by our colleague Mr. Jayesh Bhat.

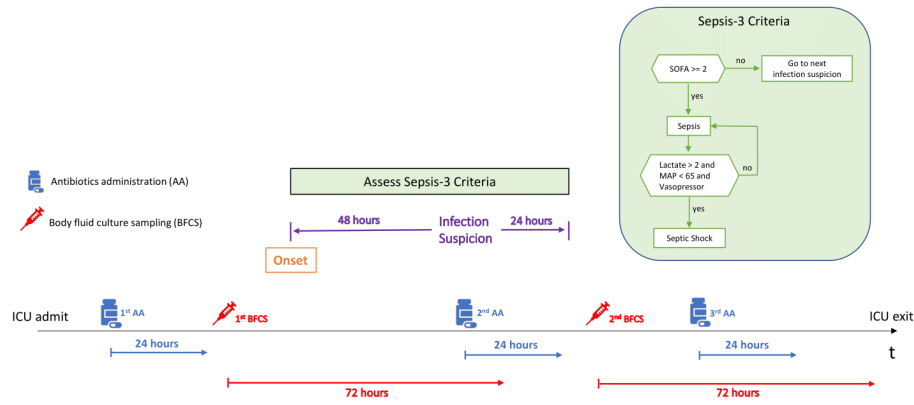


Figure 8-1: The data from the patient’s ICU stay was analyzed by retrieving the timestamps of body fluid sampling (including blood, urine, and cerebrospinal fluid) and antibiotic administration.

The criteria for determining suspected infections were established as the administration of antibiotics within 72 hours of any prior body fluid sampling or any body fluid sampling within 24 hours of prior antibiotic administration, regardless of culture results. The Sepsis-3 criteria were then assessed over a 48-hour period prior to and a 24-hour period after each suspected infection timestamp (IST). If the criteria were met within this time frame, the beginning of the time frame was recorded as the onset of the suspected infection. The schematic depicts an example where the second antibiotic administration falls within 72 hours of a prior body fluid sample, leading to a suspected infection [37]

Running the framework on all 58976 patient encounters lead to 22,547 (38.2%) sepsis patients and 3208 (5.4%) septic shock patients. For the sake of consistency and robustness of the study, we only considered adults (18+ years old at admission) who stayed between 36-hours and 144 hours in the ICU unit before the annotated onset of septic shock. According to these criteria, we identified a cohort of 630 septic patients and a non-sepsis cohort containing 6,236 patients who lacked Sepsis-3 annotation or sepsis-specific ICD-9 codes. Table D.1 in appendix D shows the demographic data of

the two cohorts.

The clinical staff prioritizes treatment procedures over on time and accurate documentation of their processes, reducing clinical annotation reliability for retrospective analysis. Therefore, we limited our data analysis to primarily machine-recorded relevant and available quantitative variables. Hence, we preselected sets of variables: the laboratory variables included bicarbonate, creatinine, blood urea nitrogen (BUN), hematocrit, hemoglobin, platelet count, white blood cell count (WBC), potassium, and sodium; the vital signs and physiological variables included body temperature, heart rate, respiratory rate, oxygen saturation (SpO2), arterial blood pressure (systolic, mean, and diastolic), and urine output. We added two generated septic markers suggested by Henry et al. [124] from the mentioned variables: the shock index (ratio of heart rate over systolic blood pressure), and the ratio of BUN to creatinine. The mean sampling rates of the selected variables in each patient cohorts can be seen in Table 8.1, and figures D-1-D-5 in appendix D exhibits their mean profile in respective cohorts.

| Variable         | Sampling Rate per Day |            |
|------------------|-----------------------|------------|
|                  | Septic                | Non-Sepsis |
| BUN              | 1.7                   | 1.1        |
| Creatinine       | 1.7                   | 2.6        |
| Hemoglobin       | 1.9                   | 5.0        |
| Bicarbonate      | 1.7                   | 1.5        |
| Respiratory Rate | 20.5                  | 4.4        |
| Heart Rate       | 20.2                  | 9.8        |
| Hematocrit       | 2.2                   | 8.2        |
| WBC              | 1.5                   | 9.2        |
| SpO2             | 20.1                  | 8.1        |
| Platelets        | 1.6                   | 5.3        |
| Systolic BP      | 19.8                  | 9.3        |
| Urine Output     | 12.1                  | 10.2       |
| Temperature      | 6.7                   | 9.1        |
| Sodium           | 2.0                   | 3.2        |
| Diastolic BP     | 19.8                  | 10.4       |
| Mean BP          | 20.0                  | 6.6        |
| Potassium        | 2.5                   | 3.8        |

Table 8.1: The mean sampling rate of pre-selected clinical variables in septic shock and non-sepsis patient populations was calculated by determining the ratio of the number of observations recorded in the ICU per patient to the length of each patient’s stay (in days) and then averaging the results across all patients in each cohort [37]

## 8.3 Methods

### 8.3.1 Time Binning

Table 8.1 shows the high heterogeneity and irregularity in the sampling frequency of the variables. We curated the uneven sampling intervals by applying a time binning approach explained in section 3.3. We transformed the time series into 30-minute bins by computing the median values (See Sect.4.1.1) over the bin windows.

### 8.3.2 Missing Value Imputation

Inadequate data collection and lack of documentation cause considerable missing values in the Longitudinal EHRs, making such data challenging to analyze. To handle the problem of missing data, we tested different imputation methods introduced in section 3.2, such as linear, polynomial, and Stineman. Comparing the results, we chose the Stineman strategy due to its exceptional performance in diminishing overshoots and handling acute changes in the imputed values.

### 8.3.3 Data Normalization

As explained in section 3.1, data normalization is a crucial step in data analysis. Due to our data's diverse scales and measurement units (See Table. 8.1), we used Z-score to normalize the data in each patient data.

### 8.3.4 Surprise Loss

The surprise loss ( $SL$ ) is a temporal-model-based feature [132], which calculates the difference between the forecast error, i.e., out-of-sample error, and the in-sample performance of a dynamic model.

The original paper [132] used a univariate linear dynamical model, whereas, in this study, we used a State-Space Model (SSM) [139, 140], as it allows us to calculate  $SL$  in a multivariate procedure. In other words, this class of model is able to unravel the hidden processes from the multiple observables/variables to model two sources of variability.

SSM hierarchically models the data with hidden states that generate observables. In the context of clinical time-series and bio-signals analysis, the hidden states can be regarded as the underlying biological processes representation, and the observables

denote the clinically measured variables. The observables in our SSM are defined as linear combinations of hidden random states. Such a model embodies the variations in the biological processes and a measurement noise term. The variations due to biology are modeled by adding a stochastic term to the hidden states, whereas the noise term is added to the observables. Both terms are presumed to follow a multivariate normal (MVN) distribution. Below, we can see the formulation of the described SSM model:

$$\begin{aligned} x_t &= x_{t-1} + w_t && \text{where } w_t \sim MVN(0, Q), y_0 \sim MVN(\pi, \Lambda) \\ y_t &= Zx_t + a + v_t && \text{where } v_t \sim MVN(0, R) \end{aligned} \quad (8.1)$$

Where  $y$  is an  $n \times T$  matrix of  $n$  observables (raw signals) and  $T$  is the length of the observables;  $x$  is an  $e \times T$  matrix of hidden states, where  $e$  is the number of hidden trends; and  $w$  is an  $e \times T$  matrix of the hidden states error. As a general rule,  $n \gg e$ . The hidden states error at time  $t$  follows an  $MVN$  distribution with mean 0 and  $e \times e$  covariance matrix  $Q$ ;  $v$  is an  $n \times T$  matrix of observation error. Similar to  $w$ , the observation error at time  $t$  follows an  $MVN$  distribution with mean 0 and  $n \times n$  covariance matrix  $R$ ;  $Z$  is an  $n \times e$  parameter matrix;  $a$  is a vector of offsets;  $\pi$  is a matrix of  $e \times 1$  means;  $\Lambda$  is an  $e \times e$  covariance matrix.

We symbolized the set of parameters in a packed form as  $\theta = (Q, R, Z, y_{1,...,T}, \pi, \Lambda)$ , and their estimate is  $\hat{\theta}$ .  $\hat{y}_t$  and  $\tilde{y}_{t+\lambda}$  are the estimate and  $\lambda$ -step-ahead forecast, respectively, of the given signals  $y_t$ .

As expressed at the beginning of this section, SL is the quantified dissimilarity between the dynamic model (here SSM) forecast error and the in-sample error of the model performance. The forecast error, i.e., the out-of-sample error, estimates the model's implementation quality in predicting the data points that are not used for fitting. While the in-sample error quantifies the overall deviation of the fitted values from the actual data used for model fitting. A high out-of-sample error compared to the in-sample error indicates an inconsistency of the model in forecasting the future, which suggests the presence of instability in the system's dynamic. In this sense, the model fitting accuracy is not a matter of importance but rather its consistency in predicting the future. We defined a loss function based on the characterized errors difference in a moving window to detect such unstable regions in the time series. In contrast to the original idea of SL [132], where the significance of such breakdowns in time series was statistically analyzed, our objective is to detect the intensified SL values as an early indicator of transitions in temporal data for further analysis. This idea proposes SL as a technique similar to structural breaks or tipping points analysis methods

[141, 142], with the additional advantage of being robust to model misspecification. For example, SSM model, like many other dynamic models, may not necessarily show an accurate fit and a good representation of the underlying disease dynamic; while SL by quantifying the deviation between in-sample and out-of-sample performance of the model and not depending on model parameters or error variances, address such misspecifications of the model.

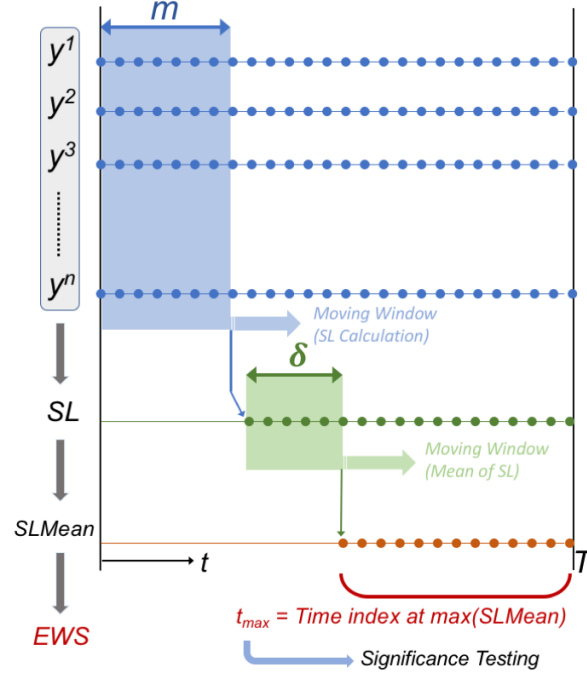


Figure 8-2: A schematic diagram for the calculation of  $SL$ ,  $SLMean$ , and  $t_{max}$ . The SSM was fitted with a moving time window of length  $m$  (as shown in blue) and the  $SL$  was computed. A second sliding window of length  $\delta$  was used to compute the  $SLMean$  (as illustrated in green). The  $T$  denotes disease onset in septic shock patients and it represents the time of discharge or death in non-sepsis patients. The  $t_{max}$  denotes the time index of the highest  $SLMean$  and it was used in our data-sampling approach [37]

Using a moving window with a length of  $m$ , we fitted the SSM model on the data within the window with the time indices  $tm + 1, \dots, t$ .  $y_t^{ic}$  represent the data variables of patient  $i$  with the clinical condition  $c$  at time index  $t$ , and  $T$  is the length of the whole patient time series data. Afterward, we used a quadratic loss function to calculate the averages of the squared differences between the estimated and the actual values, as follows:

$$L_j(\hat{\theta}_t^{ic}) = \frac{1}{n} \sum_{k=1}^n (y(k)_j^{ic} - \hat{y}(k)_j^{ic})^2 \quad (8.2)$$

where  $x(k)_j^{i_c}$  is the  $k^{th}$  element of column vector  $x_j^{i_c}$ . Likewise, the out-of-sample error is a quadratic loss function that averages the squared dissimilarities between the  $\lambda$ -step-ahead forecast and the actual future data value:

$$L_{t+\lambda}(\hat{\theta}_t^{i_c}) = \frac{1}{n} \sum_{k=1}^n (y(k)_{t+\lambda}^{i_c} - \tilde{y}(k)_{t+\lambda}^{i_c})^2 \quad (8.3)$$

Consequently, SL can be formulated as the subtraction of  $L_{t+\lambda}(\hat{\theta}_t^{i_c})$  and the average of  $L_j(\hat{\theta}_t^{i_c})$  along the window axis  $m$ :

$$SL_{t+\lambda}^{i_c} = L_{t+\lambda}(\hat{\theta}_t^{i_c}) - \frac{1}{m} \sum_{j=t-m+1}^t L_j(\hat{\theta}_t^{i_c}) \quad \text{for } t = m, \dots, T^{i_c} - \lambda \quad (8.4)$$

In addition, we applied a moving average filter (with a size delta) to make calculated  $SL$  less sensitive to short-term fluctuations due to sudden changes like monitoring errors. As a result,  $SLMean$  can be defined as:

$$SLMean_t^{i_c} = \sum_{j=t-(m+\delta)+1}^t \frac{SL_j^{i_c}}{j} \quad \text{for } t = m + \delta, \dots, T^{i_c} \quad (8.5)$$

Given patient  $i$  with the clinical condition  $c$  (in our case, septic shock), a relatively high  $SLMean_t^{i_c}$  can imply a temporal transition to the disease state. We can determine the transition time by capturing the occurrence time of  $SLMean^{i_c}$  maximum and denote it as  $t_{max}^{i_c}$ . A schematic of  $SL^{i_c}$ ,  $SLMean^{i_c}$ , and  $t_{max}^{i_c}$  calculation is illustrated in figure 8-2. Figure 8-7 depicts a simulated example of  $SLMean$  using synthetic data.

### 8.3.5 Data-sampling strategy with SLMean

One of the main steps in implementing machine learning models to study sepsis is proper feature/variable selection.

This section shows how sampling data at tipping points derived from the  $SL$  calculation can distinguish septic patients from non-sepsis groups and can be used as a strategy for selecting the proper variables for further machine learning-based analysis. It should be noted that the data sampling is merely based on the  $SL$  calculation and is independent of the patient's clinical condition. The clinical condition is taken into account in the successive statistical test.

For this purpose, we picked the variables' data at the critical transition points

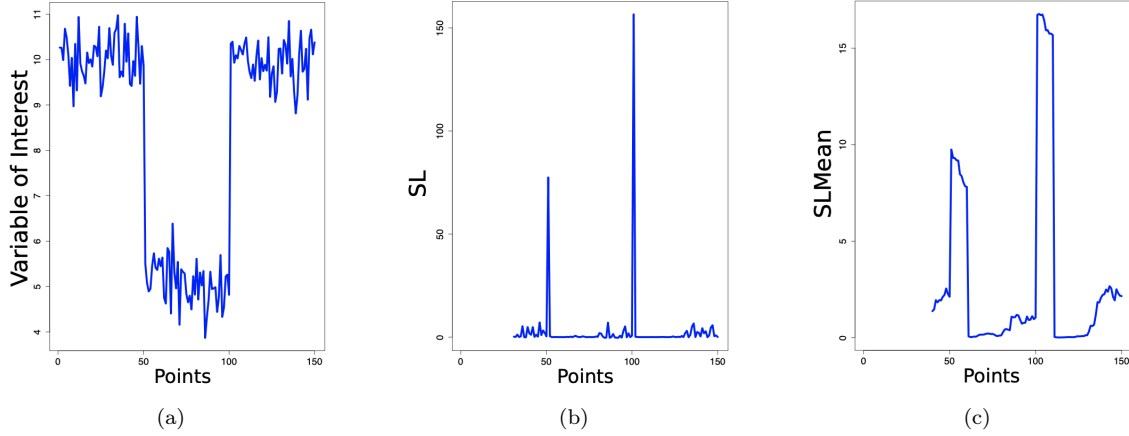


Figure 8-3: An example of *SLMean* calculation from a synthetic dataset that was generated by concatenating 50 points, drawn independently from three univariate normal distributions with different means (5, 10, 15) and a standard deviation of 0.5. Computed with a moving time window of length 30 and the number of hidden states set to 1, the magnitude of SL intensified at the 50<sup>th</sup> and 100<sup>th</sup> time-points, where the parameters of the data-generating process changed [37].

( $t_{max}$ ) and defined them as an  $n \times v$  matrix  $S^c = [y_{t_{max}}^{1c}, \dots, y_{t_{max}}^{vc}]$  where  $c \in \{0, 1\}$  denotes the non-sepsis and septic shock conditions, and  $v$  is the whole number of patients. After that, we performed a Wilcoxon rank-sum test (See Sect. 5.1.1) on each variable, which resulted in the corresponding p-value as shown below:

$$p = \left( pval(S_1^0, S_1^1), \dots, pval(S_n^0, S_n^1) \right) \quad (8.6)$$

The computed p-value can be described as the significance of the variable in differentiating the two patients groups (septic shock and non-sepsis). Figure 8-4 demonstrates the schematic of the statistical test workflow.

### 8.3.6 Bootstrapping

We presented a bootstrap test to assess the superiority of the SL-based sampling strategy over random data sampling. In this sense, the bootstrap test compares the p-values obtained from the statistical test performed on the critical transition points' data to the p-values resulting from random points in the time series. For a randomly picked time index  $t$  with its corresponding observation  $y_t^{ic}$ , where  $t \in (1, T^{ic})$ , the  $t_{random}$  p-values were computed by replacing  $t_{max}$  with  $t$ . The bootstrap test was repeated  $R$  times.

After  $R$  times iteration of the bootstrap test, we calculated the frequency of times where  $t_{max}$  p-values were more significant  $t_{random}$  p-values and expressed it as boot-

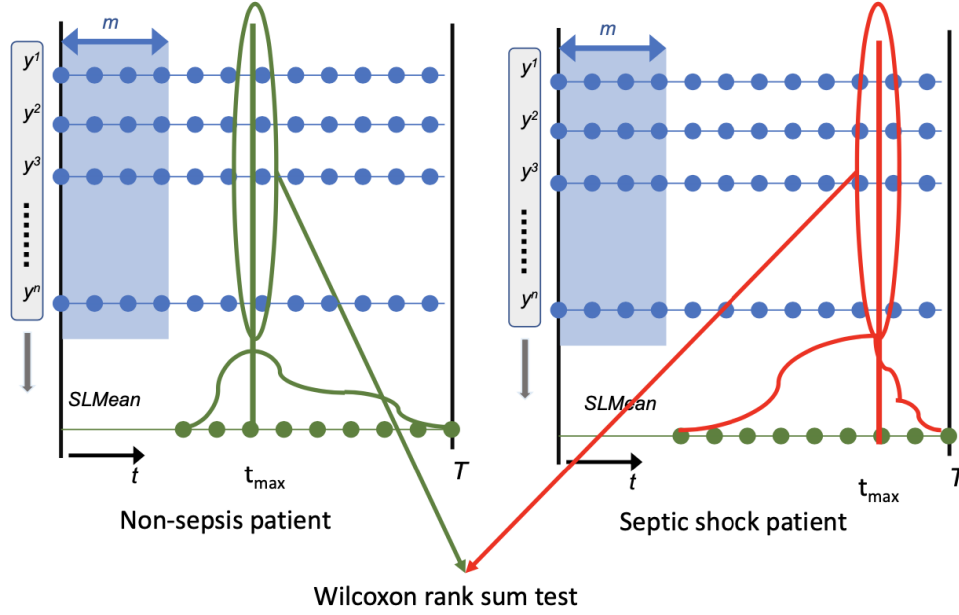


Figure 8-4: A schematic representation of our data-sampling approach utilizing our method has been presented. Critical transition points, defined as the time of the highest *SLMean*, were selected from both septic shock and non-sepsis patients. The statistical significance of the changes in variables was assessed through the application of the Wilcoxon rank-sum test [37].

strap frequency (*BF*). Therefore, A high *BF* implies that the SL-based data sampling outperforms the random approach. Additionally, to evaluate the impact of random sampling from the proximity of the septic shock onset, we calculated *BF* on randomly selected data of septic patients from two time intervals, 36 hours and 18 hours, before the annotated onset of the disease.

### 8.3.7 Autocorrelation and variance as early-warning signals

There exist other indicators of state transition in the dynamics of the systems. For example, Scheffer et al. [24] hypothesized that increasing autocorrelation and variance indicate the proximity to such transitions. To assess the performance of SL concept, we computed these two warning signals and compared the results with the outcomes from SL method. These measures are both univariate (See section X); therefore, to implement them on multivariate time series, we formulated them as follows:

$$AC1_{t'}^{i_c} = \frac{1}{N} \sum_{k=1}^N AC(y(k)_{t'})^{i_c} \quad \text{for } t' = m, \dots, T^{i_c} \quad (8.7)$$

$$VAR_{t'}^{i_c} = \frac{1}{N} \sum_{k=1}^N Var(y(k)_{t'})^{i_c} \quad \text{for } t' = m, \dots, T^{i_c} \quad (8.8)$$

where  $AC$  and  $Var$  are autocorrelation and variance functions.  $y(k)$  denotes the value of variable  $k$  in a moving time window with length of  $m$ , where  $t'$  shows the time indices in the window from  $t - m + 1$  to  $t$ . The first coefficient of auto-correlation  $AC1_{t'}^{i_c}$  and variance  $VAR_{t'}^{i_c}$  were calculated by averaging over  $N$  variables.  $i$  denotes the patient  $i$  with clinical condition  $c$ , and  $T^{i_c}$  is the length of the corresponding time series.

## 8.4 Results

### 8.4.1 SLMean as an early-warning indicator

We computed the  $SLMean^{i_c}$  in a moving window manner as described in section X, with the following parameters:  $m = 36$ , i.e., 18 hours;  $e = 3$ ;  $\lambda$ -step-ahead = 1, i.e., 30 minutes;  $\delta = 6$ , i.e., 3 hours. As to illustrate, Figure 8-5 demonstrates the calculated  $SL$  and  $SLMean$  in one patient. A higher out-sample error than in-sample error is reflected in a positive  $SLMean_{t'}^{i_c}$  value and indicates potential state transitions in the underlying dynamics.

To assess the overall  $SLMean$  trend in the septic cohort, we computed the average value and standard deviation of  $SLMean^{1_c=1}, \dots, SLMean^{N_c=1}$  (where  $N$  is the total number of septic shock patient encounters and  $c = 1$  is the septic-shock cohort index) along the time axis before the onset of the septic shock (See Fig. 8-6), where an intensification of the mean value  $SLMean$  in the proximity of the annotated onset time (time zero) can be seen. As described in section X, we calculated the mean value of  $VAR^{i_c}, \dots, VAR^{N_c}$  and  $AC1^{1_c}, \dots, AC1^{N_c}$  and similarly calculated the average and standard deviation values across time. As shown in figures 8-7a and 8-7b, in contrast to  $SLMean$ , we could not observe any substantial variation in these two indicators prior to the disease onset.

To estimate the power of  $SLMean$  in identifying the critical transition to the disease state, we located the time of the of  $SLMean$  maximum value (denoted as  $t_{max}$ ) (See section X). Figure 8-8 displays the distribution of  $t_{max}$  in septic patients. In a few cases, the maximum value of  $SLMean$  arose precisely at the onset time, indicating the  $SL$  model's inability to predict onset time before rule-based methods. However, in most cases,  $t_{max}$  occurred from 30minutes to several hours before the disease onset.

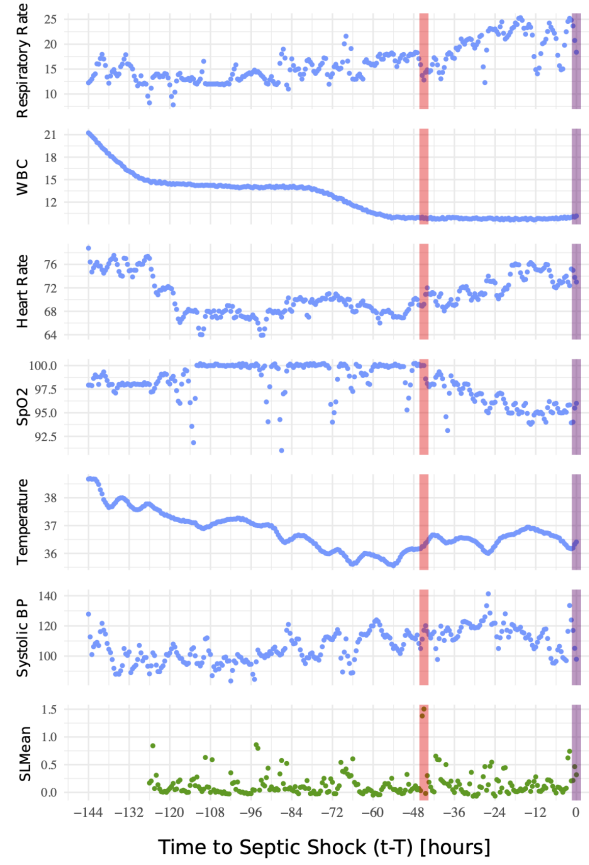


Figure 8-5: The changes over time in a set of variables used in the present study and the corresponding calculated  $SLMean^{tc}$  of a sample septic patient before the onset of septic shock (violet line). The  $SLMean^{tc}$  is computed over a moving time window ( $m = 36$ , i.e., 18 hours;  $e = 3$ ;  $\lambda$ -step-ahead = 1, i.e., 30 minutes;  $\delta = 6$ , i.e., 3 hours). The red line exhibits the time location ( $t_{max}^{tc}$ ) of the largest  $SLMean^{tc}$  (i.e., the critical transition point) [37].

In 60% of the patients, the maximum value of the  $SLmean$  appeared less than 48 hours before the onset time, while in other patients, the  $t_{max}$  was detected beyond the onset time. It can be explained by the lack of records or a low sampling rate of variables a few days before the onset of septic shock, resulting in a weak  $SLMean$  signal. On average, the highest value of  $SLMean$  happened 46 hours (median of 35.6 hours) prior to the septic shock annotated onset time. In comparison, Henry et al.[124] reported a median of 28.2 hours.

A visual assessment of the mean and standard deviation of  $SLMean$  along the time axis, reveals a notable upward trend starting 24 hours before the septic shock onset. Although it looks contradictory to the 35.6 hours median of  $t_{max}$ , elevated  $SLMeans$  (not necessarily highest) in the proximity of onset time in most patients caused such intensification in the average trend.

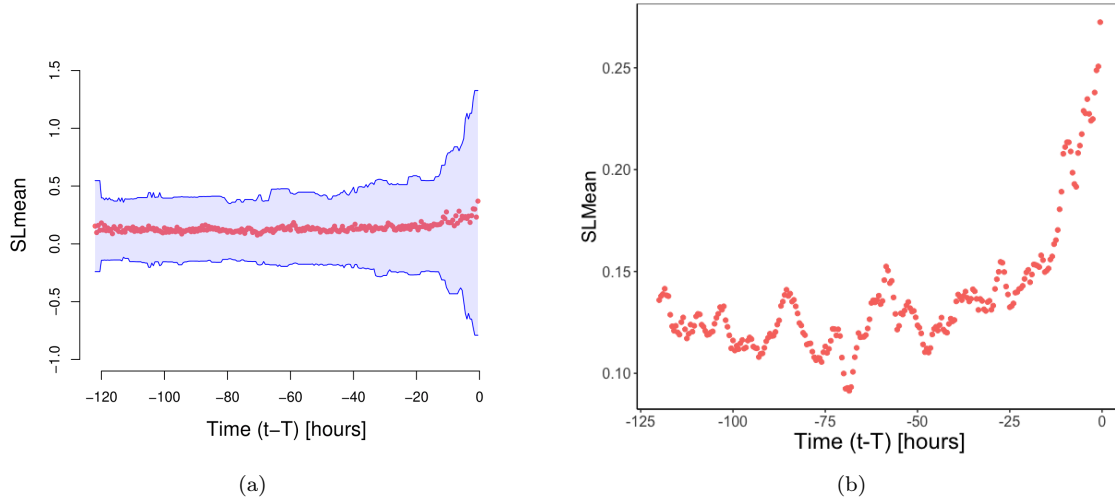


Figure 8-6: a) Componentwise mean (red dots) and  $\pm$  standard deviation (blue lines) of  $SLMean$  of all septic shock patients prior to disease-onset, b) Componentwise mean of  $SLMean$  for all septic shock patients prior to disease-onset [37]

### 8.4.2 Robustness of the SL calculation

We assessed the uncertainty in  $SL$  calculation due to the SSM model parameter alteration. We examined two other moving window lengths ( $m \in (24, 30)$ ) considering average variables sampling rate (see Table 8.1). We also picked the two values of ( $e \in (4, 5)$ ) based on the presumptions described in Sect. A similar procedure was performed to calculate  $SLMeans$  and  $t_{max}$  using the new parameters, respectively, and the obtained results were compared to the results of the reference parameters ( $m = 36$  and  $e = 3$ ) described in the previous section. Our results exhibit slight divergence in  $t_{max}$ , where the median deviation is 0, and the interquartile range (IQR) is 5.4 hours. Figure 8-9 shows the distributions of differences in calculated  $t_{max}$  for each parameter compared to the reference parameters. Although the near-zero interquartile range indicates a negligible effect of the parameter selection on the  $SL$  calculation, we can observe several outliers resulting from multiple high values of  $SLMeans$  in some patients, which contributes to the change of  $t_{max}$  due to the parameter perturbation.

### 8.4.3 SLMean-based data-sampling strategy

We performed the statistical test described in section X between the variables value samples at  $t_{max}$  in septic patients and non-septic cohorts and calculated each variable's corresponding p-values. Then in a bootstrap approach, we compared the obtained

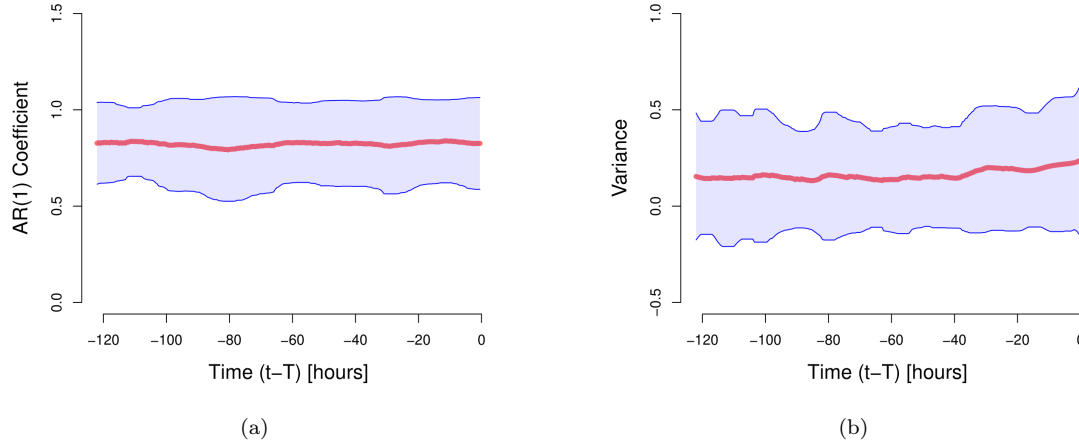


Figure 8-7: Componentwise mean (red dots) and  $\pm$  standard deviation (blue lines) of a) *AC1* and b) *VAR* of all septic shock patients prior to disease-onset [37]

p-values with p-values resulting from the sampling of variables from random time points in 1000 iterations. Next, according to the formulation in section X, the ratio of times (Bootstrap frequency) that the p-values from  $t_{max}$  data sampling are more significant than the random sampling p-values was calculated and denoted as BFSL. Similarly, we found the bootstrap frequency of all variables for *AC1* and *VAR*. Table 8.2 demonstrates the superiority of BFSL in 14 out of 19 variables.

Additionally, to test whether the random sampling in the proximity of septic shock onset outperforms *SLMean* in cohorts differentiation, we performed the bootstrap test by sampling the data from 36 and 18 hours windows prior to the disease onset. The obtained bootstrap frequencies were denoted as  $BF(18hours)$  and  $BF(36hours)$  and compared with the bootstrap frequency of the whole data, denoted as  $BF(Full)$ . Figure 8-10 shows large  $BF(Full)$  in most of the variables (15 out of 19), which indicates the data sampling at the highest *SLMean* lead to more significant p-values in discriminating septic groups from non-septic. However, by random sampling close to the annotated onset time, we observe a notable drop in the bootstrap frequency of six variables (WBC, diastolic blood pressure, Hemoglobin, SpO2, creatinine, and BUN). On the other hand, nine variables (respiratory rate, heart rate, potassium, mean blood pressure, hematocrit, shock index, temperature, BUN-creatinine, and systolic blood pressure) show no remarkable change in their Bootstrap frequencies.

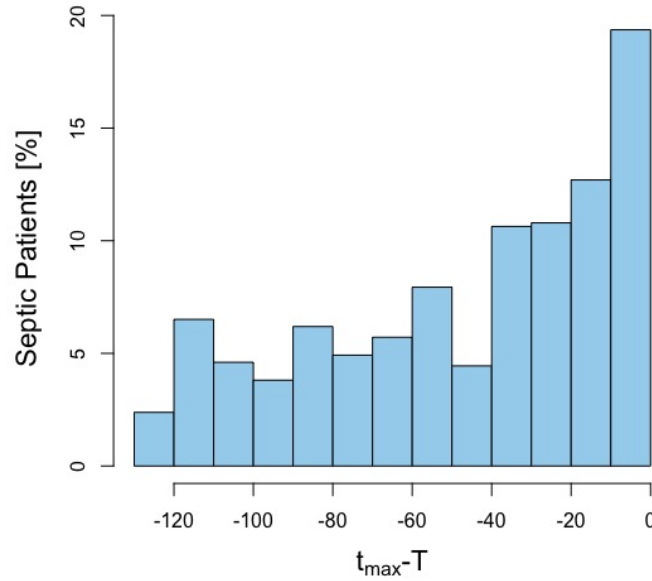


Figure 8-8: Distribution of the critical transitions times from the onset time of septic shock, i.e.  $t_{\max}^{1_c} - T^{1_c}, \dots, t_{\max}^{N_c} - T^{N_c}$ , where  $c = 1$  represents septic shock condition and  $N$  is the total number of septic shock patients.  $SLMean^{i_c}$  reaches a maximum at  $t_{\max}^{i_c}$  [37].

## 8.5 Discussions

In this clinical time series analysis application, we focused on the prediction and identification of unstable regions and critical transitions in a complex dynamic system. We believe that our findings complement the established scoring techniques in classifying the stable states as they offer a personalized assessment of the health status. In contrast to such scores based on absolute variable values, critical transitions markers are determined through the evolution of individual time series and are relatively robust to normalization and data missing values problems.

To identify such critical transitions in clinical time series, we employed the concept of surprise loss (SL), which was initially devised for instability identification of econometrics univariate forecast models based on the indifference between the in-sample error of the fitted model and the forecast error. In order to make the concept compatible with multivariate time series data of ICU patients, we utilized a multivariate SSM model as a representation of two sources of variability, i.e., the hidden underlying biological trends and the observed data. Given the calculated SL from SSM model, we introduced an early warning signal, namely, moving average SL, which showed an increasing sign of instability roughly 24 hours before the rule-based annotated onset

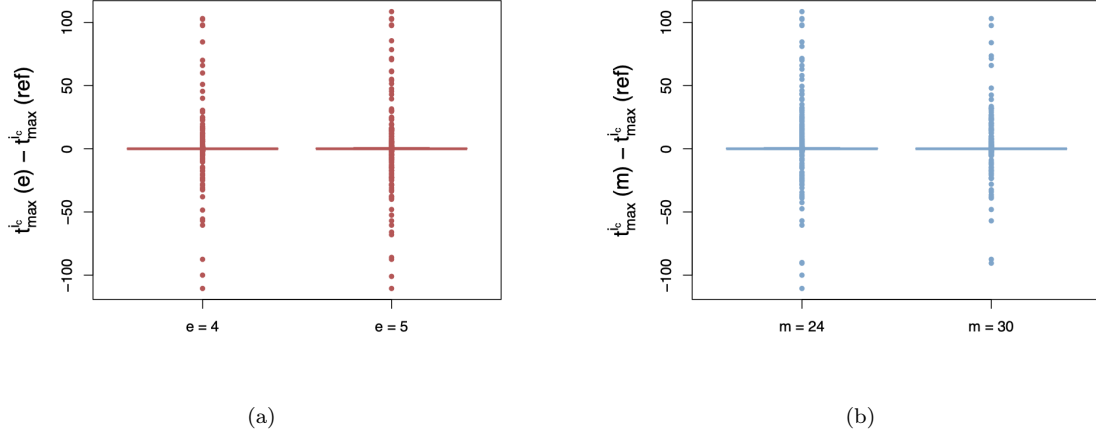


Figure 8-9: The robustness of our procedure was examined by calculating SL of septic patients ( $c = 1$ ) using different settings of moving time-window ( $m \in (24, 30)$  and  $e \in (4, 5)$ ). The corresponding  $t_{max}^c$  for each set of parameters, along with their difference from the reference parameter set ( $m = 36, e = 3$ ), were obtained. a) The difference between  $t_{max}^c(m)$  and  $t_{max}^c(ref)$  shows the effect of moving time-window length on  $t_{max}^c$ . b) The difference between  $t_{max}^c(e)$  and  $t_{max}^c(ref)$  shows the effect of number of trends on  $t_{max}^c$  [37].

of the septic shock.

Moreover, we designed a critical-transition-based data-sampling strategy for distinguishing septic shock from non-sepsis patients using the average SL. We showed through a statistical bootstrap test the advantage of our method compared to a random data selection strategy, where in general, the SL-based technique leads to more acceptable p-values and  $BF(Full)$  than the random strategy. To estimate whether sampling data in proximity of the septic shock onset surpasses our approach in discriminating sepsis from non-sepsis, we applied the bootstrap sampling approach on selectively sampled random data from 36 hours and 18 hours before the beginning of septic to calculate  $BF(36h)$  and  $BF(18h)$ , respectively. The  $BF$  results showed the consistency of the former result (random selection from the whole time series) for most of the variables. Specifically, some variables, such as blood pressure and body temperature, displayed significant differentiation p-values compared to random sampling. This finding can be explained from a systems theory point of view, where we expect the collapse of mechanisms that control the system in homeostasis close to a critical transition or unstable point. Hence, variables under tight control within stable states may show more sensitivity to systemic volatility around a tipping point.

Additionally, we compared our method outperformed two other univariate early-warning indicators of critical transitions in dynamic systems: autocorrelation and variance [24, 143, 117]., and demonstrated that SL model outperformed these two

| Variable         | BF (SL) | BF (AC1) | BF (VAR) |
|------------------|---------|----------|----------|
| BUN              | 0.98    | 0.88     | 0.95     |
| BUN-Creatinine   | 1.00    | 0.20     | 0.43     |
| Creatinine       | 0.94    | 0.98     | 1.00     |
| Hemoglobin       | 0.86    | 0.85     | 0.07     |
| Hematocrit       | 0.99    | 0.81     | 0.09     |
| Shock Index      | 0.99    | 0.03     | 0.72     |
| Respiratory Rate | 0.75    | 0.82     | 0.61     |
| Heart Rate       | 0.75    | 0.09     | 0.62     |
| Systolic BP      | 1.00    | 0.03     | 0.40     |
| Bicarbonate      | 0.1     | 0.89     | 1.00     |
| Platelets        | 0.35    | 0.15     | 0.08     |
| Temperature      | 0.99    | 0.86     | 0.93     |
| Urine Output     | 0.20    | 0.97     | 0.66     |
| WBC              | 0.75    | 0.97     | 1.00     |
| Mean BP          | 0.97    | 0.05     | 0.37     |
| SpO2             | 0.91    | 0.89     | 0.71     |
| Diastolic BP     | 0.85    | 0.06     | 0.57     |
| Sodium           | 0.42    | 0.34     | 0.16     |
| Potassium        | 0.82    | 0.04     | 0.13     |

Table 8.2: A statistical significance test (see Sect. 8.3.5) was performed to test whether the values of clinical variables at largest *SLMean*, *AC1* and *VAR* were able to statistically differentiate septic shock patients from non-sepsis patients, and a bootstrap test was performed to calculate the fraction of replications where the  $t_{max}$  p-values were less than the p-values from a random sampling. The columns of the table show the result achieved by each variable: BF (SL), BF (AC1) and BF (VAR). The highest BF for each clinical variable is shown in red [37].

measures in signaling the critical transition in the patients with septic shock. We also performed the statistical bootstrap test on the signals from these two indicators and showed their underperformance compared to our model.

Conceptually, we hypothesized that the underlying system has a stationary state. Therefore, all observed variations in the computed SL are due to the reactions to stochastic perturbations, where the system's robustness controls the deviation at the time of SL estimation. Consequently, we can describe SL deviations as a "loss of stability" indicator in a temporal system, even before a state shift, and compare it to the critical slowing down analysis indicators [143]. However, it should be noted that such deviations in SL value may neither indicate a transition to another state nor an already developed new state. This is due to the fact that an increased SL might appear by any impulsive event impacting the data, either the monitoring system aberrations or health-related covariates. Therefore, using a high SL value as the only threshold for building a warning system might lead to false alarms. Hence, robust

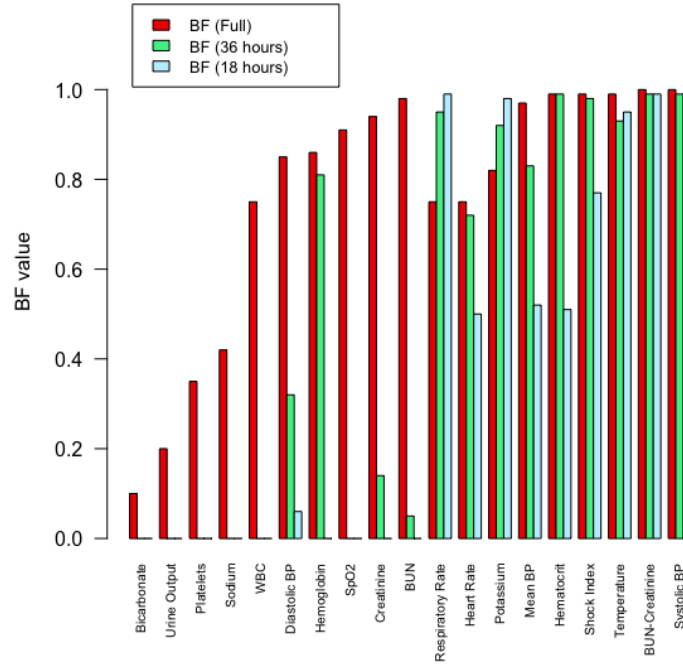


Figure 8-10: A bootstrap test, performed by randomly sampling points from 18 hours and 36 hours windows prior to the onset of septic shock, tested whether low p-values at  $t_{max}$  are an effect of time or a characteristic of regions with high  $SLMean$  values [37].

characterization of the standard SL patterns using the Machine learning models is crucial for transferring the technique to clinical applications. Another restrictive aspect of this strategy is the necessity of collecting sufficient data to perform the computation and obtain reliable results.

# Summary and Outlooks

With the advances in data recording, collection, and storage technologies, we can name the new century the era of Big Data. From transport to financial markets, advanced research to computer games, enormous data generation and analysis, and drawing meaningful findings are revolutionizing their respective industry. The medical area is no different, with big data improving all aspects of healthcare, including drug development, treatment, and patient monitoring. Continuous monitoring systems by means of wearable or clinical monitoring systems record millions of data points per hour which are impossible to be monitored or processed manually. In addition, longitudinal clinical data such as prescriptions history and laboratory test results are collected extensively. Consequently, Computational models and machine learning algorithms have enabled learning mechanisms of biomedical processes from such temporal data, which can benefit individuals' health and comfort. Nevertheless, the analysis of clinical temporal data is enormously affected by data acquisition and quality issues such as heterogeneous sampling rates, systematic and random errors, and missing values.

Data preprocessing and feature extraction are the standard steps that must be followed in time series data analysis to address these issues and perform reliable consecutive modeling. In this thesis, we introduced and developed a Python package named CMDA as an easy-to-configure platform for time series data research, enabling various data wrangling methods execution and delivering ready-to-use data for further machine learning analysis. The package is built in a modular fashion, which offers tools for importing multiple time-series data, filtering, and feature extraction using several built-in functions, which can be implemented in a pipeline. Additionally, its capability to embed user-defined functions into the pipeline distinguishes it from similar packages.

After describing the core components of the package and defining the built-in functions used in the denoising and feature extraction modules, we showed three use

cases where we implemented different modules of the package in the time series data analysis.

The first use case addresses the performance of the workflow in reading thousands of high-resolution ECG signals and transforming them into various features, including entropy parameters, spectral features, and wavelet features. A classification model followed the feature extraction step. The performance of the two sets of feature data (spectral and entropy features as one set and stationary wavelet features as another set) was explored in classifying the attributed diagnostic classes of ECG records using a random forest classification approach. Moreover, we discovered the features that significantly differentiate one diagnosis class from the rest of the diagnosis labels. We used an ensemble clustering method to explain the labels' misclassification of one of the diagnosis classes in the classification process. Finally, we employed the same clustering approach to find groups within the data and fit the individual classification model to each cluster. This hybrid strategy could improve the classification performance and make it comparable to the state-of-the-art complex deep neural network methods but with a remarkably lower effort and computation time.

The second section of the use cases results showed the heart rate variability (HRV) study of a small group of type 2 diabetic patients who underwent simultaneous continuous interstitial glucose (IG) and electrocardiogram monitoring for seven days. This case represents an example of multivariate time series data with different sampling frequencies of heart rate and IG level, as well as the irregular sampling rate issue of the heart rate. The workflow was utilized to denoise the data and extract time- and spectral-domain heart rate features in a sliding window manner to synchronize the heart rate sampling frequency with the IG level. The generated feature data suggest that relative hypoglycemic and hyperglycemic IG levels in patients with type 2 diabetes may be more relevant than absolute glucose levels in explaining the occurrence of bradyarrhythmia episodes in these patients. Additionally, we discovered the differentially expressed HRV features during the bradycardia events. Finally, we built a classifier to predict the risk of bradycardia in these patients by classifying the feature windows in the proximity of the arrhythmia episodes (risk zones). The results showed a satisfactory accuracy of the classifier in contrasting the risk zones from the normal points; however, as arrhythmias are represented as rare events, the achieved precision is not adequate for employing the model in practice, indicating the necessity of adding other factors (e.g., insulin level, diet) in the model.

As the final part of the results and the third use case, a state-space model-based feature, known as surprise loss (SL), was proposed to discover data-driven indicators of

critical transitions in a multivariate time series dataset of septic shock and non-sepsis patient cohorts in an ICU setting. In this study, our tool can be used for windowing and handling enormous missing values in the data. The core idea of SL is to train a mathematical model on time series in an unsupervised fashion and quantify the deterioration of the model’s forecast (out-of-sample) performance relative to its past (in-sample) performance. The retrospective analysis revealed that critical transitions occurred hours before the onset of septic shock, suggesting the method’s applicability as an early-warning indicator. Furthermore, the model demonstrated its applicability as a sampling strategy method by identification of highly expressed features (here, lab parameters) at the transition points.

Combined, the findings presented in this thesis showed the capacity of the introduced package in temporal data abstraction using a methodical feature extraction scheme. The developed workflow offers a robust structure to achieve fast and reliable results. Nevertheless, like any other analysis tool, the package has its limitations, and we believe there is room for further development and improvement. However, with its modular design, we encourage scientists and developers to add components and broaden the package’s usability in various aspects of time-series analysis. For example, the package offers no tool for preparing the required data structure that can be utilized in recurrent neural networks. Furthermore, with the evolution of convolutional neural networks (CNN) concepts and their application in image processing, the idea of transforming high-resolution temporal data into images and applying CNNs has become prevalent in the deep learning community [144, 145, 146, 147]. Therefore, we suggest embedding a signal to image conversion module to the package as one of the future development scopes of the package.



# Appendix A

## A.1 Data Importer Module

Data importer (`cmda.import_data`), the first component in the pipeline, offers tools for loading multiple CSV files from a server or local disk into the python coding environment (`cmda.import_data.ImportCSV`). It must be take into account that the CSV files must contain signals as columns and temporal instances as rows.

The importer module also supports importing DAT files directly from Physionet databases (`cmda.import_data.ImportWFDB`).

The importer functions can save computation time by parallelizing the data loading process. Moreover, in the case of a very long time series, if windowing a segmentation is required, A rolling window importer function performs the time binning with a defined window length and step size and distributes each window's importing process on available nodes in parallel mode. Users can access the rolling window via `cmda.import_data.RollingWindowCSV`. library has handful modules to import continuous data from local disk or Physionet databases.

## A.2 Preprocessing Module

`cmda.preprocessing.Filters` is a tool for applying different filters and outlier removal functions on a numeric signal. User can select from a number of built-in filters, through a dynamic attribute of the `Filters` object, called `.add` that gives the user access to the following functions:

| Feature function        | Feature        | Description                                                       |
|-------------------------|----------------|-------------------------------------------------------------------|
| <code>rm_outlier</code> | Remove Outlier | Removing outliers based on a <i>low</i> and <i>high</i> threshold |

|                                   |                    |                                                                                                             |
|-----------------------------------|--------------------|-------------------------------------------------------------------------------------------------------------|
| <code>rm_outliers_quantile</code> | Remove Quantiles   | Removing the values in an array based on <i>low</i> and <i>high</i> defined quantiles                       |
| <code>butter_filter</code>        | Butterworth Filter | A predesigned Butterworth filter that can be used as a band-pass denoising tool                             |
| <code>interpolate</code>          | Interpolation      | A <i>linear</i> or <i>cubic</i> interpolation tool that can be used for imputing missing values in an array |

Table A.1: Built-in preprocessing features

Moreover, user-defined filters can be added easily to the `Filters` object instance via `.udf.add`. This functionality makes it possible to tailor the filters based on the application.

## A.3 Feature Extraction Module

`cmda.feature_extraction.Features` is a tool for extracting multiple features from a numeric time series array. This library provides several ready-to-use built-in time domain, spectral domain, nonlinear, and wavelet features that can be selected and added to a `Features` object instance via `.add` function.

Additionally, Users can define and add their own functions to the `Features` object instance via `.udf.add`. The available built-in function in the current version of the package are described in the following sections.

### A.3.1 Time-domain Features

Time-domain features are the signal descriptors in a time analysis span. The list of all built-in time-domain features is shown in Table A.2. These functions are defined using the `NumPy` package.

| Feature function    | Feature            | Description     |
|---------------------|--------------------|-----------------|
| <code>mean</code>   | Mean               | See Sect. 4.1.1 |
| <code>max</code>    | Maximum            | See Sect. 4.1.1 |
| <code>min</code>    | Minimum            | See Sect. 4.1.1 |
| <code>median</code> | Median             | See Sect. 4.1.1 |
| <code>std</code>    | Standard Deviation | See Sect. 4.1.1 |

|                          |                         |                                                                              |
|--------------------------|-------------------------|------------------------------------------------------------------------------|
| <a href="#">skewness</a> | Skewness                | See Sect. <a href="#">4.1.1</a>                                              |
| <a href="#">kurtosis</a> | Kurtosis                | See Sect. <a href="#">4.1.1</a>                                              |
| <a href="#">p2p</a>      | Peak-to-Peak            | See Sect. <a href="#">4.1.1</a>                                              |
| <a href="#">rms</a>      | Root Means Square       | See Sect. <a href="#">4.1.3</a>                                              |
| <a href="#">zcr</a>      | Zero Crossing Rate      | See Sect. <a href="#">4.1.4</a>                                              |
| <a href="#">mad</a>      | Mean Absolute Deviation | The median of the absolute deviations an array. Also called median of slopes |
| <a href="#">mns</a>      | Mean Slope              | The mean differences of an array                                             |

Table A.2: Built-in time-domain features

### A.3.2 Frequency-domain Features

Frequency-domain features are signal characteristics in the frequency domain, which are extracted from the Fourier transform of the signal. The list of all built-in frequency-domain features is shown in Table [A.3](#). These functions are defined using the [NumPy](#) and [SciPy](#) packages.

| Feature function            | Feature                       | Description                     |
|-----------------------------|-------------------------------|---------------------------------|
| <a href="#">periodogram</a> | Periodogram                   | See Sect. <a href="#">4.2.1</a> |
| <a href="#">welch</a>       | Welch Periodogram             | See Sect. <a href="#">4.2.1</a> |
| <a href="#">mnf</a>         | Mean frequency                | See Sect. <a href="#">4.2.2</a> |
| <a href="#">mdf</a>         | Median frequency              | See Sect. <a href="#">4.2.3</a> |
| <a href="#">vcf</a>         | Variance of central frequency | See Sect. <a href="#">4.2.4</a> |
| <a href="#">peaks</a>       | Peak Frequency                | See Sect. <a href="#">4.2.5</a> |
| <a href="#">psr</a>         | Power spectrum ratio          | See Sect. <a href="#">4.2.6</a> |
| <a href="#">band_power</a>  | Band Power                    | See Sect. <a href="#">4.2.7</a> |
| <a href="#">band_std</a>    | Band Standard Deviation       | See Sect. <a href="#">4.2.7</a> |
| <a href="#">band_mnf</a>    | Band Mean Frequency           | See Sect. <a href="#">4.2.7</a> |
| <a href="#">band_mdf</a>    | Band Median Frequency         | See Sect. <a href="#">4.2.7</a> |

Table A.3: Built-in frequency-domain features

## A.4 Entropy Features

Shannon entropy and its derivations are methods for estimating the complexity of dynamic systems. Table A.4 shows the available entropy built-in functions in the CMDA package.

| Feature function                 | Feature             | Description     |
|----------------------------------|---------------------|-----------------|
| <a href="#">entropy</a>          | Shannon Entropy     | See Sect. 4.3   |
| <a href="#">sample_entropy</a>   | Sample Entropy      | See Sect. 4.3.1 |
| <a href="#">perm_entropy</a>     | Permutation Entropy | See Sect. 4.3.2 |
| <a href="#">spectral_entropy</a> | Spectral Entropy    | See Sect. 4.3.3 |

Table A.4: Built-in entropy features

## A.5 Wavelet Features

[swt\\_features](#) function that can return a set of time-domain and frequency-domain of the decomposition levels coefficients (See Sect. 4.4.2). The following parameters can be given to the function:

- Mean Frequency ([mnf](#))
- Power Spectrum Ratio ([psr](#))
- Peak Frequency ([peak](#))
- Median Absolute Deviation ([mds](#))
- Mean Slope ([mns](#))
- Shannon Entropy ([see](#))
- Permutation Entropy ([perm-ent](#))
- Normalized Energy ([nse](#))

The explanation of each parameter can be found in Tables A.2, A.4, and A.3.

## A.6 Feature Engineering Pipeline

[cmda.pipeline.Pipeline](#) is a tool for importing, pre-processing and feature extraction of multiple records (files) in an automated manner.

User must create an importer object instance ([cmda.import\\_data](#)), a feature object instance [cmda.preprocessing.Features](#), and if required a filter object instance ([cmda.preprocessing.Filters](#)) and feed these instances into the pipeline. The process can be run either consecutively or in parallel using [.run](#) function.

# Appendix B

## B.1 Feature Engineering

### B.1.1 Feature Object

`cmda.feature_extraction.Features` is a tool for extracting multiple features at the same time from a numeric signal `x`. One can select from a wide variety of built-in features, as well as adding user-defined features. This functionality makes it possible to tailor the features based on the application. The following subsections show examples of time domain, frequency domain, and user-defined feature extraction.

#### Import the data

We start by using sample data contains an ECG signal and ABP (Arterial Blood Pressure) signal with a length of 10 seconds and 125 Hz sampling rate. The toy data can be accessed from `cmda.data.ecg_abp_sample`.

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 from cmda.data import ecg_abp_sample
5
6 data = ecg_abp_sample()
7
8 ecg = data["ECG"]
9 abp = data["ABP"]
10
11 fs = 125
12 t = np.linspace(1, 1 / fs, len(ecg))
13 fig, (ax1, ax2) = plt.subplots(2, sharex=True)
14 ax1.plot(t, ecg)
15 ax1.set_ylabel("ECG [mV]")
```

```

16 ax2.plot(t, abp, c="red")
17 ax2.set_ylabel("ABP [mmHg]")
18 ax2.set_xlabel("Time [seconds]")
19 plt.show()

```

Listing B.1: Importing a sample data

Output:

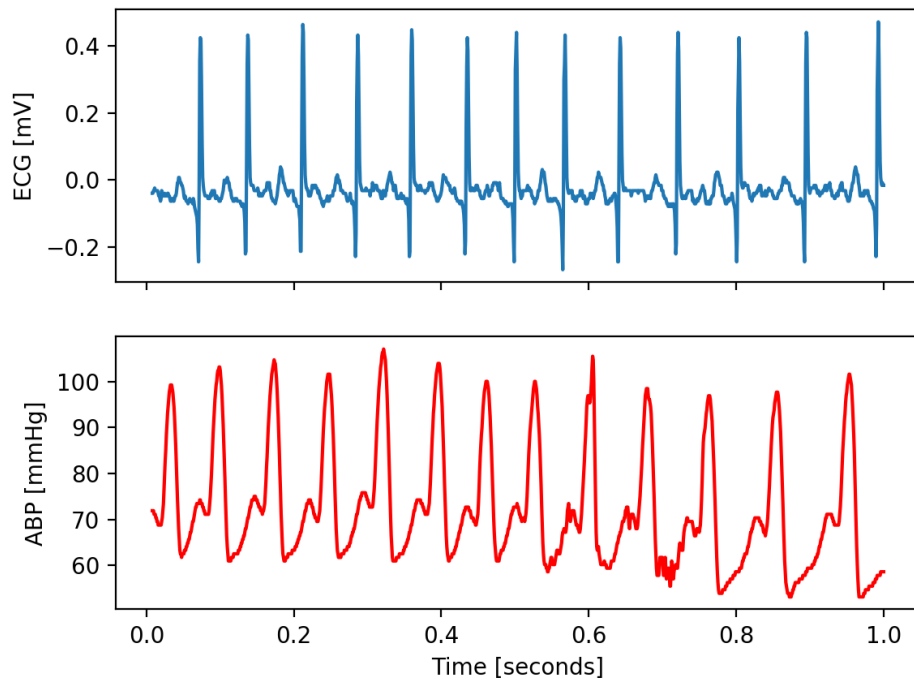


Figure B-1: Imported ECG and ABP sample signals

### Time-domain feature extraction

In this example, we extract some built-in time-domain features of the ABP signal. To start, we must call the `cmda.feature_extraction.Features` object and add the desired features using the `add` function.

After adding the features, we apply the `transform` function to the signal and get the features. The `transform` function returns the extracted features as a python dictionary.

```

1 from cmda.feature_extraction import Features
2 from cmda.data import ecg_apb_sample
3

```

```
4 data = ecg_apb_sample()
5 x = data["ABP"]
6
7 feature = Features()
8
9 # add built-in time-domain features
10 feature.add.mean()
11 feature.add.max()
12 feature.add.min()
13 feature.add.median()
14 feature.add.skewness()
15 feature.add.kurtosis()
16 feature.add.std()
17 feature.add.p2p()
18 feature.add.zcr(center=True)
19
20 feature.transform(x=x, fs=125)
```

Listing B.2: Time-domain feature extraction

Output:

```
1 {'mean': 72.385,
2  'max': 107.03125,
3  'min': 53.125,
4  'median': 69.53125,
5  'skewness': 0.9154627781784881,
6  'kurtosis': -0.15299373030467667,
7  'std': 13.268216310510619,
8  'p2p': 53.90625,
9  'zcr': 42}
```

Listing B.3: Time-domain feature extraction output

Frequency-domain feature extraction In this part, we extract some built-in frequency-domain features of the ECG signal. Similar to the previous example, we call the `cmda.feature_extraction.Features` object, and append the desired features using the `add` function.

Using the `transform` function returns the extracted features as a python dictionary.

```
1 from cmda.feature_extraction import Features
2 from cmda.data import ecg_apb_sample
3
4 data = ecg_apb_sample()
5 x = data["ECG"]
6
```

```

7 feature = Features()
8
9 # add frequency-domain built-in features
10 feature.add.mnf(spectrum="ps")
11 feature.add.mdf(spectrum="ps")
12 feature.add.std(f(spectrum="ps")
13 feature.add.psr(spectrum="welch", int_limit_ratio=0.01)
14 feature.add.peaks(spectrum="welch", n_peaks=1, height=True, width=
    True)
15 feature.add.band_power(spectrum="ps", low=1, high=7)
16 feature.add.band_mnf(spectrum="ps", low=1, high=7)
17
18 feature.transform(x=x, fs=125)

```

Listing B.4: Frequency-domain feature extraction

Output:

```

1 {'mnf': 9.857968968908676,
2  'mdf': 9.0,
3  'stdf': 6.657686374206612,
4  'psr_0.01': 0.1086823980353686,
5  'peak_freq_1': 3.90625,
6  'peak_height_1': 0.0009058970155242925,
7  'peak_width_1': 4.386465062242469,
8  'power_[1,7]Hz': 0.3921784430303598,
9  'mnf_[1,7]Hz': 4.277888519896743}

```

Listing B.5: Frequency-domain feature extraction output

## User-defined feature extraction

In this section, we derive some features, including a user-defined feature from the ABP signal. Similar to the previous example, we call the `cmda.feature_extraction.Features` object, Define the feature and add them to the feature object, using `add_udf` function. The `transform` function returns the extracted features as a python dictionary.

```

1 import numpy as np
2 from cmda.feature_extraction import Features
3 from cmda.data import ecg_apb_sample
4
5 data = ecg_apb_sample()
6 x = data["ABP"]
7
8 # Define the feature

```

```

9  # Please take note, that the output must be a dictionary
10 # ,where the keys are the extracted features
11 def quantile(x,low = 0.1,high =0.9):
12     q_low = np.quantile(x,q=low)
13     q_high = np.quantile(x,q=high)
14
15     res = {"q_low":q_low, "q_high":q_high}
16     return res
17
18
19 # Call the Feature object
20 feature = Features()
21
22 # add built-in features
23 feature.add.mean()
24 feature.add.max()
25 feature.add.zcr(center=True)
26 feature.add.mnf(spectrum="ps")
27
28 # add user-defined feature
29 feature.udf.add(quantile)
30
31 feature.transform(x=x, fs=125)

```

Listing B.6: User-defined feature extraction

Output:

```

1 {'mean': 72.385,
2  'max': 107.03125,
3  'zcr': 42,
4  'mnf': 2.0056974184541025,
5  'q_low': -13.79125,
6  'q_high': 23.70875}

```

Listing B.7: User-defined feature extraction output

## B.1.2 Multivariate Feature Extraction

In this tutorial, we show how to extract features from a multivariate signal. In the first example, we extract similar features from all the variables, and in the second part, we show how to extract different features from each variable. Please read [B.1.1](#) on how to create a feature object first. We use the same data as used in the Feature

Object tutorial (B.1.1). The data contain two signals (ECG, ABP) with a sampling frequency = 500.

### Feature extraction with one feature object

To extract features from a dataset, we use `cmda.feature_extraction.extract_features` function. First, we create a feature object with `cmda.feature_extraction.Features`.

```

1 from cmda.feature_extraction import Features
2 from cmda.feature_extraction import extract_features
3 from cmda.data import ecg_apb_sample
4
5 # load the data
6 data = ecg_apb_sample()
7
8 # Create the feature object
9 features = Features()
10
11 # Add built-in features
12 features.add.std()
13 features.add.mnf()
14 features.add.stdf()
15
16 # extract features
17 res = extract_features(data=data, feature_obj=features)
18
19 print(res)

```

Listing B.8: Multivariate feature extraction using one feature object

Output:

```

1 {'ECG_std': 0.0912508348533301,
2  'ECG_mnf': 0.0788637517512694,
3  'ECG_stdf': 0.053261490993652905,
4  'ABP_std': 13.268216310510619,
5  'ABP_mnf': 0.01604557934763282,
6  'ABP_stdf': 0.012480758701313835}

```

### Feature extraction with multiple feature objects

In various real applications, extracting similar features from different signals may lead to wrong data compression and unnecessary information. To avoid that, we show how to create and apply signal-specific feature objects.

```
1 from cmda.feature_extraction import Features
2 from cmda.feature_extraction import extract_features
3 from cmda.data import ecg_apb_sample
4
5 # load the data
6 data = ecg_apb_sample()
7
8 # Create the feature object for ECG signal
9 ecg_features = Features()
10
11 # Add built-in features
12 ecg_features.add.mnf()
13 ecg_features.add.stdf()
14
15
16 # Create the feature object for ECG signal
17 abp_features = Features()
18
19 # Add built-in features
20 abp_features.add.mean()
21 abp_features.add.std()
22 abp_features.add.max()
23 abp_features.add.min()
24
25 # Create the feature object as a dict
26 features = {'ECG':ecg_features, 'ABP':abp_features}
27
28 # extract eatures
29 res = extract_features(data=data, feature_obj=features)
30
31 print(res)
```

Listing B.9: Multivariate feature extraction using one feature object

Output:

```
1 {'ECG_mnf': 0.0788637517512694,
2  'ECG_stdf': 0.053261490993652905,
3  'ABP_mean': 72.385,
4  'ABP_std': 13.268216310510619,
5  'ABP_max': 107.03125,
6  'ABP_min': 53.125}
```

### B.1.3 Feature Extraction Pipeline

This tutorial shows how to implement `cmda.pipeline.Pipeline` to extract features from a Physionet WFDB. We extract features from the ECGRDVQ database. The database contains 5232 extracted 10-second standard 12 lead ECG segments. The meta-data of the database (`ecgrdvq_metadata`), which is a dictionary consisting of the record names `record_name` and their corresponding public directories `public_dir`, is available in `cmda.data.wfdb`. The pipeline includes importing the data, applying filters to the records, and feature extraction from records. To import the data from Physionet WFDB, we use the `cmda.read_data.ReadWFDB`. We create the filtering object by `cmda.filter.Filters` and create the feature object by `cmda.feature_extraction.Features`. A detailed tutorial on creating feature objects can be found in B.1.1.

```

1 from cmda.data.wfdb import ecgrdvq_metadata
2 from cmda.read_data import ReadWFDB
3 from cmda.preprocessing import Filters
4 from cmda.feature_extraction import Features
5 from cmda.pipeline import Pipeline
6
7 # Load the meta-data
8 metadata = ecgrdvq_metadata()
9
10 # get the record names and their corresponding public directories
11 record_names = metadata['record_name']
12 public_dir = metadata['public_dir']
13
14 # Build the importer object using ReadWFDB
15 # Set the channels to ["II",'V1'] to import these channels
    exclusively.
16 importer = ReadWFDB(record_names=record_names, public_dir=public_dir
    , channels=['II','V1'])
17
18 # Create the filter object and add a buuterworth low-pass filter
19 filter_obj = Filters()
20 filter_obj.add.butter_filter(cutoff=60, btype="lowpass")
21
22 # Create the feature object
23 feature_obj = Features()
24 feature_obj.add.mnf()
25 feature_obj.add.mdf()
26 feature_obj.add.psr()

```

```

27 feature_obj.add.stdf()
28 feature_obj.add.peaks(n_peaks=1,spectrum='welch',height=False,width=
    False,nperseg=512)
29 feature_obj.add.band_power(low=0.6, high=2,spectrum='welch')
30 feature_obj.add.band_power(low=2, high=4,spectrum='welch')
31 feature_obj.add.band_power(low=4, high=6,spectrum='welch')
32 feature_obj.add.band_power(low=6, high=10,spectrum='welch')
33 feature_obj.add.band_power(low=10, high=15,spectrum='welch')
34 feature_obj.add.band_power(low=15, high=30,spectrum='welch')
35
36 # Build the pipeline
37 ecgrdvq_pipeline = Pipeline(importer=importer,features=feature_obj,
    filters=filter_obj)
38
39 # Run the pipeline
40 # Set the dataframe_output to True, to get the extracted as a Pandas
    dataframe
41 # n_jobs set the number of cpu cores for parallel feature extraction
42 res = ecgrdvq_pipeline.run(n_jobs=4, dataframe_output = True)

```

Listing B.10: Multivariate feature extraction using one feature object

Output:

```

1 Running the pipeline on 5232 instances...
2
3 100% [||||||||||||||||] 5232/5232 [00:05<00:00, 2.23it/s]
4
5 finished!

```

The required time for parallel running the above pipeline was around 2640 seconds for a PC with a 3,5 GHz Dual-Core Intel Core i7 processor and 16 GB 2133 MHz LPDDR3 memory. The process included retrieving the data from Physionet data servers too.



# Appendix C

## C.1 NORM classification

| Cluster          | Metric   |      |           |        |      |
|------------------|----------|------|-----------|--------|------|
|                  | Accuracy | AUC  | Precision | Recall | F1   |
| 0                | 0.93     | 0.96 | 0.91      | 0.96   | 0.94 |
| 1                | 0.87     | 0.94 | 0.83      | 0.92   | 0.87 |
| 2                | 0.92     | 0.97 | 0.90      | 0.94   | 0.92 |
| 3                | 0.94     | 0.98 | 0.91      | 0.98   | 0.94 |
| 4                | -        | -    | -         | -      | -    |
| Weighted Average | 0.92     | 0.96 | 0.89      | 0.95   | 0.92 |

Table C.1: ]

Extracted SWT features data of ECG records were clustered using the consensus clustering method. A One-vs-All Random Forests model was employed in a bootstrapping fashion to classify NORM records. Classification metrics of NORM prediction in individual clusters, as well as the average weighted metrics were computed, given K=5.

Accuracy 0.923917 AUC 0.972727 Precision 0.899996 Recall 0.955063 F1 0.926470

## C.2 HYP classification

| Cluster          | Metric   |      |           |        |      |
|------------------|----------|------|-----------|--------|------|
|                  | Accuracy | AUC  | Precision | Recall | F1   |
| 0                | 0.81     | 0.87 | 0.77      | 0.88   | 0.82 |
| 1                | 0.84     | 0.91 | 0.81      | 0.89   | 0.85 |
| 2                | 0.84     | 0.92 | 0.80      | 0.92   | 0.86 |
| 3                | 0.93     | 0.97 | 0.89      | 0.98   | 0.94 |
| 4                | 0.98     | 1.00 | 0.97      | 0.99   | 0.98 |
| Weighted Average | 0.87     | 0.93 | 0.84      | 0.93   | 0.88 |

Table C.2: ]

Extracted SWT features data of ECG records were clustered using the consensus clustering method. A One-vs-All Random Forests model was employed in a bootstrapping fashion to classify CD records. Classification metrics of CD prediction in individual clusters, as well as the average weighted metrics were computed, given K=5.

| Cluster          | Metric   |      |           |        |      |
|------------------|----------|------|-----------|--------|------|
|                  | Accuracy | AUC  | Precision | Recall | F1   |
| 0                | 0.87     | 0.93 | 0.88      | 0.85   | 0.86 |
| 1                | 0.88     | 0.95 | 0.87      | 0.89   | 0.88 |
| 2                | 0.87     | 0.94 | 0.84      | 0.91   | 0.87 |
| 3                | 0.92     | 0.96 | 0.89      | 0.95   | 0.92 |
| 4                | 0.95     | 0.97 | 0.92      | 0.99   | 0.95 |
| Weighted Average | 0.89     | 0.95 | 0.88      | 0.92   | 0.89 |

Table C.3: Extracted SWT features data of ECG records were clustered using the consensus clustering method. A One-vs-All Random Forests model was employed in a bootstrapping fashion to classify HYP records. Classification metrics of HYP prediction in individual clusters, as well as the average weighted metrics were computed, given K=5.

| Cluster          | Metric   |      |           |        |      |
|------------------|----------|------|-----------|--------|------|
|                  | Accuracy | AUC  | Precision | Recall | F1   |
| 0                | 0.84     | 0.92 | 0.82      | 0.86   | 0.84 |
| 1                | 0.87     | 0.94 | 0.83      | 0.93   | 0.88 |
| 2                | 0.86     | 0.93 | 0.82      | 0.93   | 0.87 |
| 3                | 0.92     | 0.95 | 0.88      | 0.98   | 0.93 |
| 4                | 0.94     | 0.97 | 0.90      | 0.99   | 0.94 |
| Weighted Average | 0.88     | 0.94 | 0.84      | 0.93   | 0.88 |

Table C.4: Extracted SWT features data of ECG records were clustered using the consensus clustering method. A One-vs-All Random Forests model was employed in a bootstrapping fashion to classify MI records. Classification metrics of MI prediction in individual clusters, as well as the average weighted metrics were computed, given K=5.

| Cluster          | Metric   |      |           |        |      |
|------------------|----------|------|-----------|--------|------|
|                  | Accuracy | AUC  | Precision | Recall | F1   |
| 0                | 0.83     | 0.92 | 0.85      | 0.80   | 0.82 |
| 1                | 0.87     | 0.94 | 0.83      | 0.93   | 0.88 |
| 2                | 0.88     | 0.94 | 0.83      | 0.95   | 0.89 |
| 3                | 0.93     | 0.97 | 0.89      | 0.97   | 0.93 |
| 4                | 0.95     | 0.98 | 0.91      | 0.99   | 0.95 |
| Weighted Average | 0.88     | 0.94 | 0.85      | 0.94   | 0.89 |

Table C.5: ]

Extracted SWT features data of ECG records were clustered using the consensus clustering method. A One-vs-All Random Forests model was employed in a bootstrapping fashion to classify STTC records. Classification metrics of STTC prediction in individual clusters, as well as the average weighted metrics were computed, given  $K=5$ .



# Appendix D

|                                                                       | Non-Sepsis (6236)       |                        | Septic Shock (630)    |                       |
|-----------------------------------------------------------------------|-------------------------|------------------------|-----------------------|-----------------------|
|                                                                       | Female                  | Male                   | Female                | Male                  |
| Number of patients                                                    | 2700 (43.29 %)          | 3536 (56.70 %)         | 250 (39.68 %)         | 380 (60.31 %)         |
| Age                                                                   | 63.18 $\pm$ 18.76 (185) | 60.91 $\pm$ 16.03 (86) | 65.56 $\pm$ 15.18 (4) | 64.74 $\pm$ 15.38 (4) |
| In-hospital mortality                                                 | 629 (23.30 %)           | 705 (19.94 %)          | 160 (64.00 %)         | 248 (65.26 %)         |
| Infectious and parasitic diseases                                     | 243 (9.00 %)            | 305 (8.63 %)           | 163 (65.20 %)         | 241 (63.42 %)         |
| Neoplasms                                                             | 484 (17.93 %)           | 549 (15.53 %)          | 49 (19.60 %)          | 91 (23.95 %)          |
| Endocrine, nutritional and metabolic diseases, and immunity disorders | 1783 (66.04 %)          | 2283 (64.56 %)         | 189 (75.60 %)         | 290 (76.32 %)         |
| Diseases of the blood and blood-forming organs                        | 815 (30.19 %)           | 758 (21.44 %)          | 151 (60.40 %)         | 210 (55.26 %)         |
| Mental disorders                                                      | 864 (32.00 %)           | 920 (26.02 %)          | 81 (32.40 %)          | 98 (25.79 %)          |
| Diseases of the nervous system                                        | 659 (24.41 %)           | 701 (19.82 %)          | 84 (33.60 %)          | 124 (32.63 %)         |
| Diseases of the sense organs                                          | 223 (8.26 %)            | 181 (5.12 %)           | 24 (9.60 %)           | 23 (6.05 %)           |
| Diseases of the circulatory system                                    | 2139 (79.22 %)          | 2931 (82.89 %)         | 225 (90.00 %)         | 345 (90.79 %)         |
| Diseases of the respiratory system                                    | 1076 (39.85 %)          | 1118 (31.62 %)         | 205 (82.00 %)         | 313 (82.37 %)         |
| Diseases of the digestive system                                      | 885 (32.78 %)           | 1052 (29.75 %)         | 164 (65.60 %)         | 243 (63.95 %)         |
| Diseases of the genitourinary system                                  | 705 (26.11 %)           | 880 (24.89 %)          | 186 (74.40 %)         | 278 (73.16 %)         |
| Complications of pregnancy, childbirth, and the puerperium            | 17 (0.63 %)             | 0 (0.00 %)             | 0 (0.00 %)            | 0 (0.00 %)            |
| Diseases of the skin and subcutaneous tissue                          | 144 (5.33 %)            | 147 (4.16 %)           | 50 (20.00 %)          | 88 (23.16 %)          |
| Diseases of the musculoskeletal system and connective tissue          | 661 (24.48 %)           | 405 (11.45 %)          | 46 (18.40 %)          | 61 (16.05 %)          |
| Congenital anomalies                                                  | 128 (4.74 %)            | 147 (4.16 %)           | 13 (5.20 %)           | 13 (3.42 %)           |
| Certain conditions originating in the perinatal period                | 0 (0.00 %)              | 0 (0.00 %)             | 0 (0.00 %)            | 0 (0.00 %)            |
| Symptoms, signs, and ill-defined conditions                           | 779 (28.85 %)           | 835 (23.61 %)          | 145 (58.00 %)         | 227 (59.74 %)         |
| Injury and poisoning                                                  | 824 (30.52 %)           | 1107 (31.31 %)         | 197 (78.80 %)         | 289 (76.05 %)         |
| External causes of injury and supplemental classification             | 1662 (61.56 %)          | 2114 (59.79 %)         | 170 (68.00 %)         | 269 (70.79 %)         |

Table D.1: Demographic data of our septic shock and non-sepsis cohorts. According to MIMIC-III documentation, date of birth has been shifted for patients older than 89. The median age for such *very old* patients was 91.4. When calculating mean age of the individual groups, *very old* patients' age (count of such patients indicated in parenthesis) was set to 91.4. The major categories of disease was based on ICD9

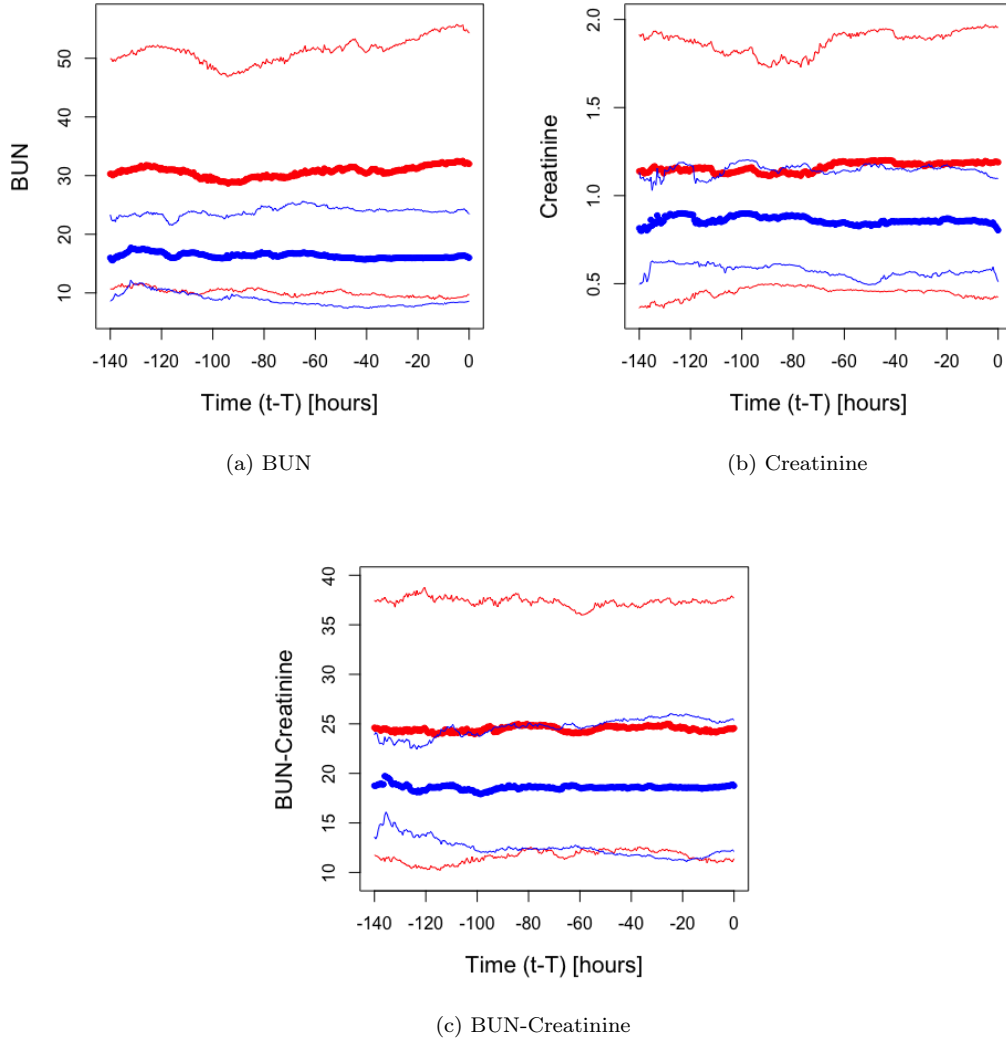


Figure D-1: Median (*dots*) and median deviation (*lines*) of variables in septic shock patients (**red**) and non-sepsis patients (**blue**).  $T$  is the length of time series, and  $t - T$  is the time before the onset of septic shock in septic patients or the time before discharge/decease in non-sepsis patients.

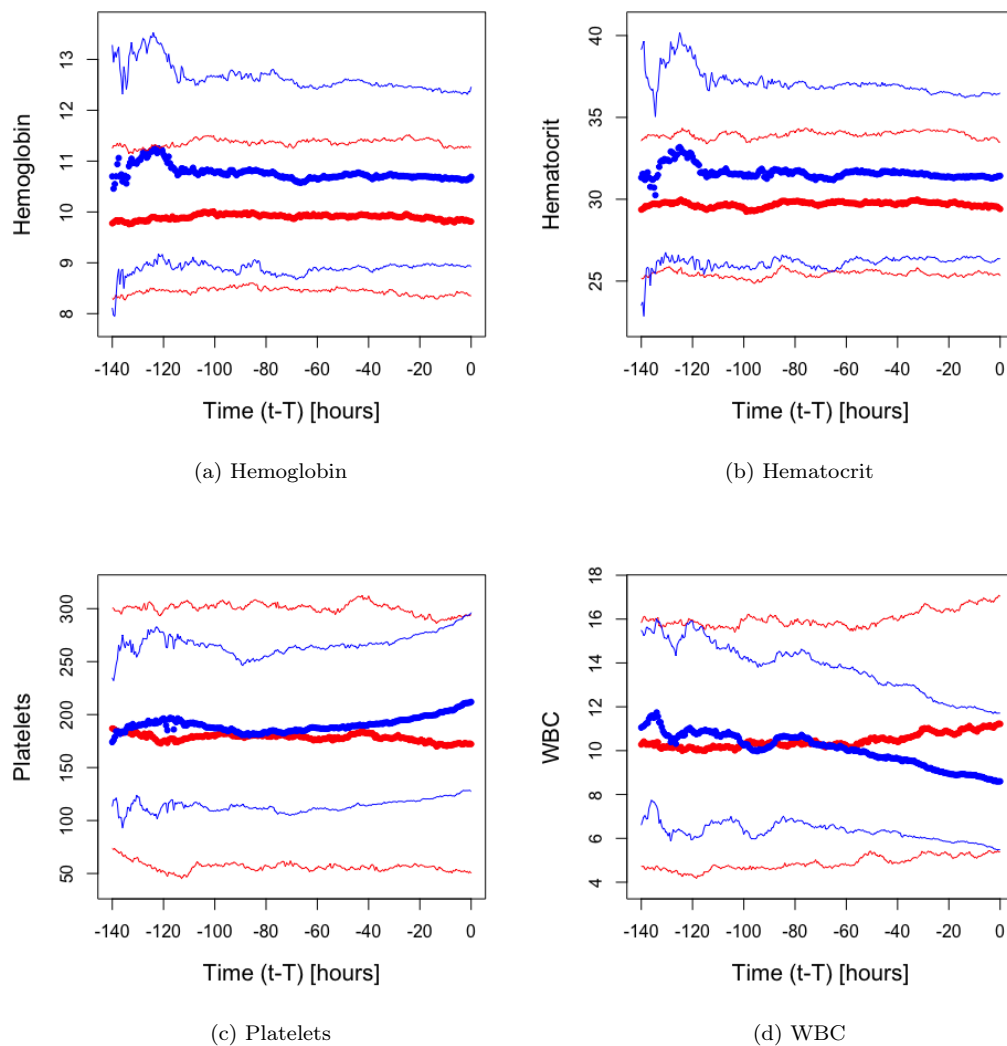


Figure D-2: Median (*dots*) and median deviation (*lines*) of variables in septic shock patients (*red*) and non-sepsis patients (*blue*).  $T$  is the length of time series, and  $t - T$  is the time before the onset of septic shock in septic patients or the time before discharge/decease in non-sepsis patients.

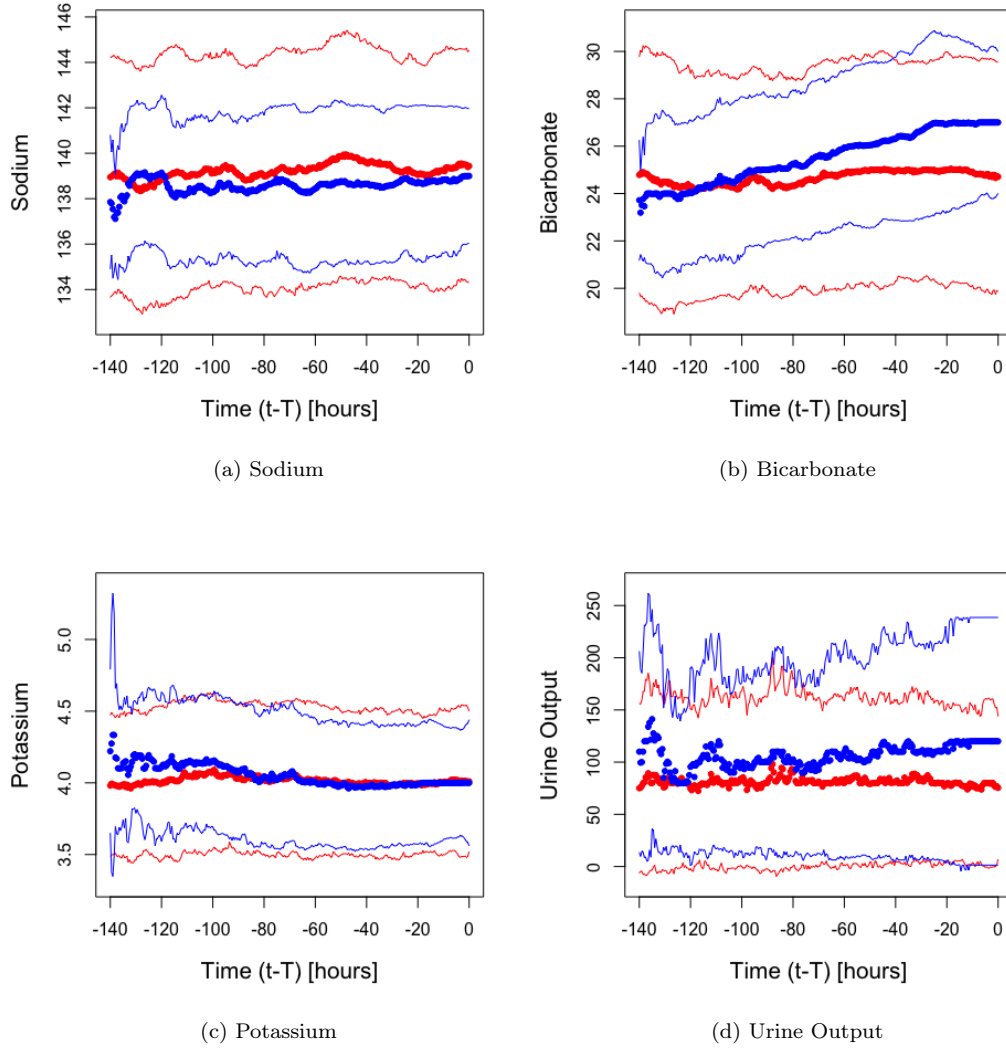


Figure D-3: Median (*dots*) and median deviation (*lines*) of variables in septic shock patients (*red*) and non-sepsis patients (*blue*).  $T$  is the length of time series, and  $t - T$  is the time before the onset of septic shock in septic patients or the time before discharge/decease in non-sepsis patients.

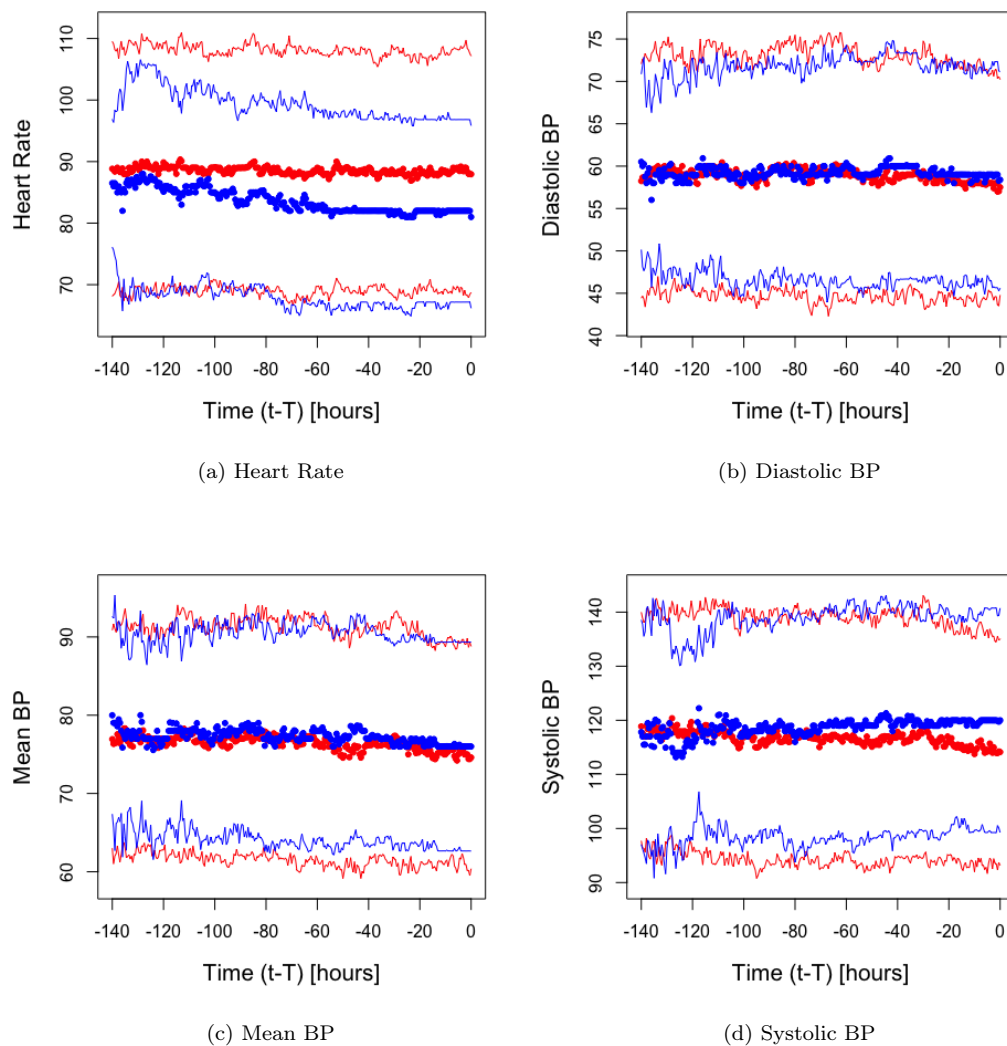


Figure D-4: Median (*dots*) and median deviation (*lines*) of variables in septic shock patients (*red*) and non-sepsis patients (*blue*).  $T$  is the length of time series, and  $t - T$  is the time before the onset of septic shock in septic patients or the time before discharge/decease in non-sepsis patients.

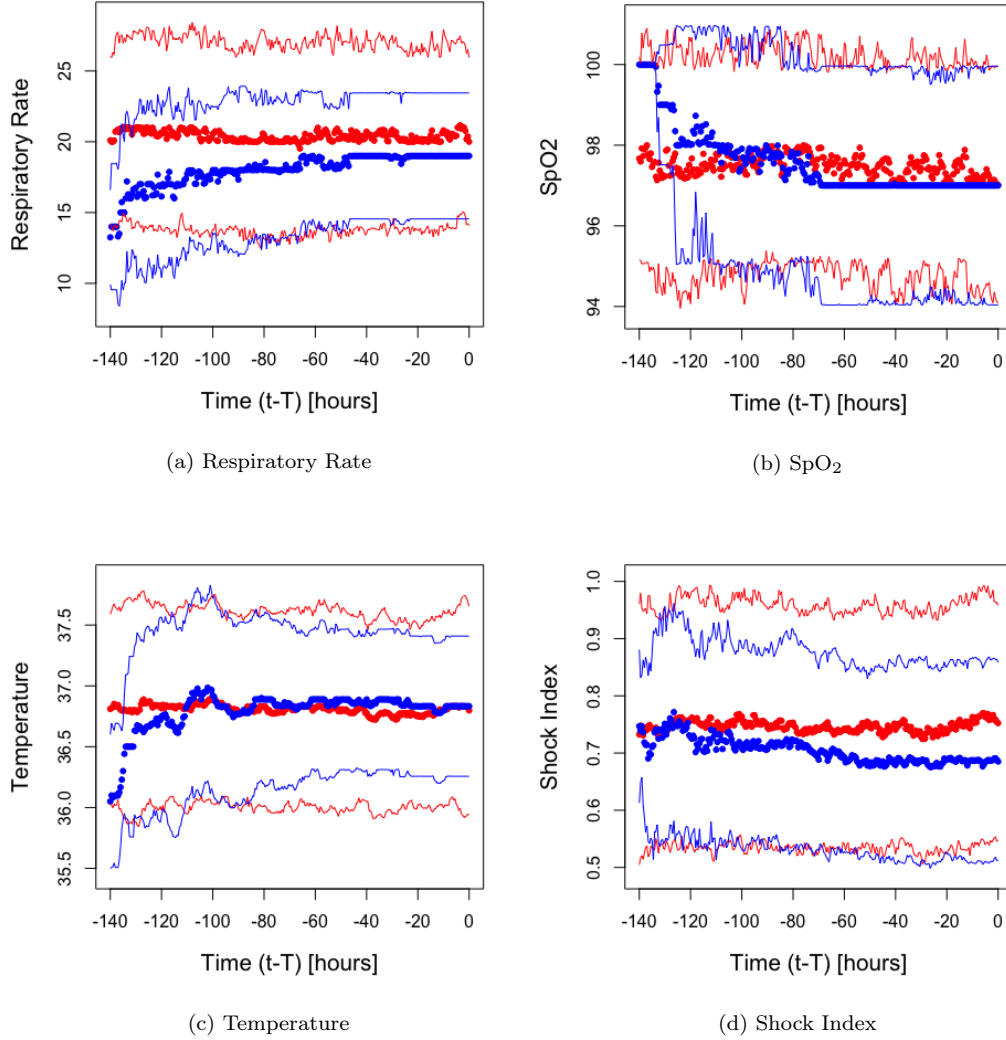


Figure D-5: Median (*dots*) and median deviation (*lines*) of variables in septic shock patients (*red*) and non-sepsis patients (*blue*).  $T$  is the length of time series, and  $t - T$  is the time before the onset of septic shock in septic patients or the time before discharge/decease in non-sepsis patients.

# Bibliography

- [1] Jin Yang, Yuanjie Li, Qingqing Liu, Li Li, Aozi Feng, Tianyi Wang, Shuai Zheng, Anding Xu, and Jun Lyu. Brief introduction of medical database and data mining technology in big data era. *Journal of Evidence-Based Medicine*, 13(1):57–69, February 2020.
- [2] Riccardo Bellazzi. Big Data and Biomedical Informatics: A Challenging Opportunity. *Yearbook of Medical Informatics*, 23(01):08–13, August 2014.
- [3] Choong Ho Lee and Hyung-Jin Yoon. Medical big data: promise and challenges. *Kidney Research and Clinical Practice*, 36(1):3–11, March 2017.
- [4] Sarah B. Scruggs, Karol Watson, Andrew I. Su, Henning Hermjakob, John R. Yates, Merry L. Lindsey, and Peipei Ping. Harnessing the Heart of Big Data. *Circulation Research*, 116(7):1115–1119, March 2015.
- [5] A. Sinha, G. Hripcsak, and M. Markatou. Large Datasets in Biomedicine: A Discussion of Salient Analytic Issues. *Journal of the American Medical Informatics Association*, 16(6):759–767, November 2009.
- [6] Xinzhi Zhang, Eliseo J. Pérez-Stable, Philip E. Bourne, Emmanuel Peprah, O. Kenrik Duru, Nancy Breen, David Berrigan, Fred Wood, James S. Jackson, David W.S. Wong, and Joshua Denny. Big Data Science: Opportunities and Challenges to Address Minority Health and Health Disparities in the 21st Century. *Ethnicity & Disease*, 27(2):95, April 2017.
- [7] James M. Snyder, Jacob A. Pawloski, and Laila M. Poisson. Developing Real-world Evidence-Ready Datasets: Time for Clinician Engagement. *Current Oncology Reports*, 22(5):45, May 2020.
- [8] Richard Hodson. Precision medicine. *Nature*, 537(7619):S49–S49, September 2016.
- [9] Benedikt E. Maissenhaelter, Ashley L. Woolmore, and Peter M. Schlag. Real-world evidence research based on big data: Motivation—challenges—success factors. *Der Onkologe*, 24(S2):91–98, November 2018.
- [10] Holger Fröhlich, Rudi Balling, Niko Beerenwinkel, Oliver Kohlbacher, Santosh Kumar, Thomas Lengauer, Marloes H. Maathuis, Yves Moreau, Susan A. Murphy, Teresa M. Przytycka, Michael Rebhan, Hannes Röst, Andreas Schuppert,

- Matthias Schwab, Rainer Spang, Daniel Stekhoven, Jimeng Sun, Andreas Weber, Daniel Ziemek, and Blaz Zupan. From hype to reality: data science enabling personalized medicine. *BMC Medicine*, 16(1):150, December 2018.
- [11] Dominik Lautsch, Kristy Iglay, Lingfeng Yang, Neha Bansal, Riddhi Markan, Niranjana Kathe, and Swapnil Rajpathak. Level of glycemic control among US type 2 diabetes mellitus patients on dual therapy of metformin and sodium–glucose cotransporter 2 inhibitor: a retrospective database study. *Current Medical Research and Opinion*, 36(10):1583–1589, October 2020.
- [12] Jasmanda Wu, Cunlin Wang, Sengwee Toh, Federica Edith Pisa, and Larry Bauer. Use of real-world evidence in regulatory decisions for rare diseases in the United States—Current status and future directions. *Pharmacoepidemiology and Drug Safety*, 29(10):1213–1218, October 2020.
- [13] M. Di Rienzo, F. Rizzo, G. Parati, G. Brambilla, M. Ferratini, and P. Castiglioni. MagIC System: a New Textile-Based Wearable Device for Biological Signal Monitoring. Applicability in Daily Life and Clinical Setting. In *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, pages 7167–7169, Shanghai, China, 2005. IEEE.
- [14] Eugenijus Kaniusas. *Biomedical signals and sensors*. Biological and medical physics, biomedical engineering. Springer, Heidelberg, 2012.
- [15] Duarte Dias and João Paulo Silva Cunha. Wearable Health Devices—Vital Sign Monitoring, Systems and Technologies. *Sensors*, 18(8):2414, July 2018.
- [16] Yashodhan Athavale and Sridhar Krishnan. Biosignal monitoring using wearables: Observations and opportunities. *Biomedical Signal Processing and Control*, 38:22–33, September 2017.
- [17] Fernando Seoane, Inmaculada Mohino-Herranz, Javier Ferreira, Lorena Alvarez, Ruben Buendia, David Ayllón, Cosme Llerena, and Roberto Gil-Pita. Wearable Biomedical Measurement Systems for Assessment of Mental Stress of Combatants in Real Time. *Sensors*, 14(4):7120–7141, April 2014.
- [18] Suzanne M. Cox, Ashley Lane, and Samuel L. Volchenbourn. Use of Wearable, Mobile, and Sensor Technology in Cancer Clinical Trials. *JCO clinical cancer informatics*, 2:1–11, December 2018.
- [19] James Williamson, Qi Liu, Fenglong Lu, Wyatt Mohrman, Kun Li, Robert Dick, and Li Shang. Data sensing and analysis: Challenges for wearables. In *The 20th Asia and South Pacific Design Automation Conference*, pages 136–141, Chiba, Japan, January 2015. IEEE.
- [20] Lukas Smital, Clifton R. Haider, Martin Vitek, Pavel Leinveber, Pavel Jurak, Andrea Nemcova, Radovan Smisek, Lucie Marsanova, Ivo Provaznik, Christopher L. Felton, Barry K. Gilbert, and David R. Holmes. Real-Time Quality

- Assessment of Long-Term ECG Signals Recorded by Wearables in Free-Living Conditions. *IEEE Transactions on Biomedical Engineering*, 67(10):2721–2734, October 2020.
- [21] Yasser Khan, Aminy E. Ostfeld, Claire M. Lochner, Adrien Pierre, and Ana C. Arias. Monitoring of Vital Signs with Flexible and Wearable Medical Devices. *Advanced Materials*, 28(22):4373–4395, June 2016.
- [22] Scott L. Zeger, Rafael Irizarry, and Roger D. Peng. On Time Series Analysis of Public Health and Biomedical Data. *Annual Review of Public Health*, 27(1):57–79, April 2006.
- [23] Tak-chung Fu. A review on time series data mining. *Engineering Applications of Artificial Intelligence*, 24(1):164–181, February 2011.
- [24] Marten Scheffer, Jordi Bascompte, William A. Brock, Victor Brovkin, Stephen R. Carpenter, Vasilis Dakos, Hermann Held, Egbert H. van Nes, Max Rietkerk, and George Sugihara. Early-warning signals for critical transitions. *Nature*, 461(7260):53–59, September 2009.
- [25] Irfan Pratama, Adhistya Erna Permanasari, Igi Ardiyanto, and Rini Indrayani. A review of missing values handling methods on time-series data. In *2016 International Conference on Information Technology Systems and Innovation (ICITSI)*, pages 1–6, 2016.
- [26] Chao Tang, G. Chen, M. Fu, and Rui-jin Liao. Space charge behavior in multi-layer oil-paper insulation under different DC voltages and temperatures. *IEEE Transactions on Dielectrics and Electrical Insulation*, 17(3):775–784, June 2010.
- [27] Fabian Mörchen. Time series feature extraction for data mining using DWT and DFT. 2003. Publisher: Citeseer.
- [28] Can Wang, Mitra Baratchi, Thomas Bäck, Holger H. Hoos, Steffen Limmer, and Markus Olhofer. Towards Time-Series Feature Engineering in Automated Machine Learning for Multi-Step-Ahead Forecasting. In *ITISE 2022*, page 17. MDPI, June 2022.
- [29] Saeed Aghabozorgi, Ali Seyed Shirkhorshidi, and Teh Ying Wah. Time-series clustering – A decade review. *Information Systems*, 53:16–38, October 2015.
- [30] Sven Festag and Cord Spreckelsen. Semantic Anomaly Detection in Medical Time Series. *Studies in Health Technology and Informatics*, 278:118–125, May 2021.
- [31] L. Glass and D. Kaplan. Time series analysis of complex dynamics in physiology and medicine. *Medical Progress Through Technology*, 19(3):115–128, 1993.

- [32] Andreas K. Maier, Christopher Syben, Bernhard Stimpel, Tobias Würfl, Mathis Hoffmann, Frank Schebesch, Weilin Fu, Leonid Mill, Lasse Kling, and Silke Christiansen. Learning with known operators reduces maximum error bounds. *Nature Machine Intelligence*, 1(8):373–380, August 2019.
- [33] Maximilian Christ, Andreas W. Kempa-Liehr, and Michael Feindt. Distributed and parallel time series feature extraction for industrial big data applications. 2016. Publisher: arXiv Version Number: 3.
- [34] Rollin Mccraty and Fred Shaffer. Heart Rate Variability: New Perspectives on Physiological Mechanisms, Assessment of Self-regulatory Capacity, and Health Risk. *Global Advances in Health and Medicine*, 4(1):46–61, January 2015.
- [35] Angkoon Phinyomark, Pornchai Phukpattaranont, and Chusak Limsakul. Feature reduction and selection for EMG signal classification. *Expert Systems with Applications*, 39(8):7420–7431, June 2012.
- [36] Zhengzheng Xing, Jian Pei, Philip S. Yu, and Ke Wang. Extracting Interpretable Features for Early Classification on Time Series. In *Proceedings of the 2011 SIAM International Conference on Data Mining*, pages 247–258. Society for Industrial and Applied Mathematics, April 2011.
- [37] Pejman F. Ghalati, Satya S. Samal, Jayesh S. Bhat, Robert Deisz, Gernot Marx, and Andreas Schuppert. Critical Transitions in Intensive Care Units: A Sepsis Case Study. *Scientific Reports*, 9(1):12888, September 2019.
- [38] Maximilian Christ, Nils Braun, Julius Neuffer, and Andreas W. Kempa-Liehr. Time Series Feature Extraction on basis of Scalable Hypothesis tests (tsfresh – A Python package). *Neurocomputing*, 307:72–77, September 2018.
- [39] Gary Kee Khoon Lee, Henry Kasim, Rajendra Prasad Sirigina, Shannon Shi Qi How, Stephen King, and Terence Gih Guang Hung. Smart Robust Feature Selection (SoFt) for imbalanced and heterogeneous data. *Knowledge-Based Systems*, 236:107197, January 2022.
- [40] Romain Tavenard, Johann Faouzi, Gilles Vandewiele, Felix Divo, Guillaume Androz, Chester Holtz, Marie Payne, Roman Yurchak, Marc Rußwurm, Kushal Kolar, and others. Tslearn, a machine learning toolkit for time series data. *J. Mach. Learn. Res.*, 21(118):1–6, 2020.
- [41] Johann Faouzi and Hicham Janati. pyts: A Python Package for Time Series Classification. *J. Mach. Learn. Res.*, 21:46–1, 2020.
- [42] David M Burns and Cari M Whyne. Seglearn: A python package for learning sequences and time series. *The Journal of Machine Learning Research*, 19(1):3238–3244, 2018. Publisher: JMLR. org.

- [43] Brett Naul, Stéfan van der Walt, Arien Crellin-Quick, Joshua S. Bloom, and Fernando Pérez. cesium: Open-Source Platform for Time-Series Inference. 2016. Publisher: arXiv Version Number: 1.
- [44] James Max Kanter and Kalyan Veeramachaneni. Deep feature synthesis: Towards automating data science endeavors. In *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–10, Campus des Cordeliers, Paris, France, October 2015. IEEE.
- [45] Raoul A. Collenteur, Mark Bakker, Ruben Caljé, Stijn A. Klop, and Frans Schaars. Pastas: Open Source Software for the Analysis of Groundwater Time Series. *Groundwater*, 57(6):877–885, November 2019.
- [46] Markus Löning, Anthony Bagnall, Sajaysurya Ganesh, Viktor Kazakov, Jason Lines, and Franz J. Király. sktime: A Unified Interface for Machine Learning with Time Series. 2019. Publisher: arXiv Version Number: 1.
- [47] Lisa-Katrin Turnhoff, Ali Hadizadeh Esfahani, Maryam Montazeri, Nina Kusch, and Andreas Schuppert. FORESEE: a tool for the systematic comparison of translational drug response modeling pipelines. *Bioinformatics*, 35(19):3846–3848, October 2019.
- [48] Ary L. Goldberger, Luis A. N. Amaral, Leon Glass, Jeffrey M. Hausdorff, Plamen Ch. Ivanov, Roger G. Mark, Joseph E. Mietus, George B. Moody, Chung-Kang Peng, and H. Eugene Stanley. PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. *Circulation*, 101(23), June 2000.
- [49] Russel W Stineman. A Consistently Well Behaved Method of Interpolation. *Creative Computing*, 6(7):54–57, 1980.
- [50] Gerardo M.E. Perillo and María Cintia Piccolo. An interpolation method for estuarine and oceanographic data. *Computers & Geosciences*, 17(6):813–820, January 1991.
- [51] Angkoon Phinyomark, Sirinee Thongpanja, Huosheng Hu, Pornchai Phukpat-taranont, and Chusak Limsakul. The Usefulness of Mean and Median Frequencies in Electromyography Analysis. In Ganesh R. Naik, editor, *Computational Intelligence in Electromyography Analysis - A Perspective on Current Applications and Future Challenges*. InTech, October 2012.
- [52] Leon Cohen. *Time-frequency analysis*. Prentice Hall signal processing series. Prentice Hall PTR, Englewood Cliffs, N.J, 1995.
- [53] Joshua S. Richman and J. Randall Moorman. Physiological time-series analysis using approximate entropy and sample entropy. *American Journal of Physiology-Heart and Circulatory Physiology*, 278(6):H2039–H2049, June 2000.

- [54] Alfonso Delgado-Bonal and Alexander Marshak. Approximate Entropy and Sample Entropy: A Comprehensive Tutorial. *Entropy*, 21(6):541, May 2019.
- [55] Christoph Bandt and Bernd Pompe. Permutation Entropy: A Natural Complexity Measure for Time Series. *Physical Review Letters*, 88(17):174102, April 2002.
- [56] Y N Pan, J Chen, and X L Li. Spectral entropy: A complementary index for rolling element bearing performance degradation assessment. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 223(5):1223–1231, May 2009.
- [57] Ali. N Akansu and Richard. A Hadad. *Multiresolution Signal Decomposition*. Elsevier, 2001.
- [58] Nasser Kehtarnavaz. *Digital Signal Processing System Design*. Elsevier, 2008.
- [59] J.E. Fowler. The redundant discrete wavelet transform and additive noise. *IEEE Signal Processing Letters*, 12(9):629–632, 2005.
- [60] Student. The Probable Error of a Mean. *Biometrika*, 6(1):1, March 1908.
- [61] Henry B. Mann. Nonparametric Tests Against Trend. *Econometrica*, 13(3):245, July 1945.
- [62] Andrew M. McIntosh, Michael Sharpe, and Stephen M. Lawrie. Research methods, statistics and evidence-based practice. In *Companion to Psychiatric Studies*, pages 157–198. Elsevier, 2010.
- [63] Robert V. Hogg, Elliot A. Tanis, and Dale L. Zimmerman. *Probability and statistical inference*. Pearson, Boston, ninth edition edition, 2015.
- [64] R. A. Fisher. On the Interpretation of  $\chi^2$  from Contingency Tables, and the Calculation of P. *Journal of the Royal Statistical Society*, 85(1):87, January 1922.
- [65] D. R. Cox. The Regression Analysis of Binary Sequences. *Journal of the Royal Statistical Society. Series B (Methodological)*, 20(2):215–242, 1958. Publisher: [Royal Statistical Society, Wiley].
- [66] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. Linear Methods for Classification. In *The Elements of Statistical Learning*, pages 101–137. Springer New York, New York, NY, 2009. Series Title: Springer Series in Statistics.
- [67] Raoof Gholami and Nikoo Fakhari. Support Vector Machine: Principles, Parameters, and Applications. In *Handbook of Neural Computation*, pages 515–535. Elsevier, 2017.

- [68] Hanan Al-Hadeethi, Shahab Abdulla, Mohammed Diykh, Ravinesh C. Deo, and Jonathan H Green. Adaptive boost LS-SVM classification approach for time-series signal classification in epileptic seizure diagnosis applications. *Expert Systems with Applications*, 161:113676, December 2020.
- [69] W. A. Chaovalitwongse and P. M. Pardalos. On the time series support vector machine using dynamic time warping kernel for brain activity classification. *Cybernetics and Systems Analysis*, 44(1):125–138, January 2008.
- [70] A. Kampouraki, G. Manis, and C. Nikou. Heartbeat Time Series Classification With Support Vector Machines. *IEEE Transactions on Information Technology in Biomedicine*, 13(4):512–518, July 2009.
- [71] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. Support Vector Machines and Flexible Discriminants. In *The Elements of Statistical Learning*, pages 417–458. Springer New York, New York, NY, 2009. Series Title: Springer Series in Statistics.
- [72] Leo Breiman. [No title found]. *Machine Learning*, 45(1):5–32, 2001.
- [73] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [74] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. Random Forests. In *The Elements of Statistical Learning*, pages 587–604. Springer New York, New York, NY, 2009. Series Title: Springer Series in Statistics.
- [75] Stefano Monti. Consensus Clustering: A Resampling-Based Method for Class Discovery and Visualization of Gene Expression Microarray Data. *Machine Learning*, 52(1/2):91–118, 2003.
- [76] Emelia J. Benjamin, Salim S. Virani, Clifton W. Callaway, Alanna M. Chamberlain, Alexander R. Chang, Susan Cheng, Stephanie E. Chiuve, Mary Cushman, Francesca N. Delling, Rajat Deo, Sarah D. de Ferranti, Jane F. Ferguson, Myriam Fornage, Cathleen Gillespie, Carmen R. Isasi, Monik C. Jiménez, Lori Chaffin Jordan, Suzanne E. Judd, Daniel Lackland, Judith H. Lichtman, Lynda Lisabeth, Simin Liu, Chris T. Longenecker, Pamela L. Lutsey, Jason S. Mackey, David B. Matchar, Kunihiro Matsushita, Michael E. Mussolino, Khurram Nasir, Martin O’Flaherty, Latha P. Palaniappan, Ambarish Pandey, Dilip K. Pandey, Mathew J. Reeves, Matthew D. Ritchey, Carlos J. Rodriguez, Gregory A. Roth, Wayne D. Rosamond, Uchechukwu K.A. Sampson, Gary M. Satou, Svati H. Shah, Nicole L. Spartano, David L. Tirschwell, Connie W. Tsao, Jenifer H. Voeks, Joshua Z. Willey, John T. Wilkins, Jason HY. Wu, Heather M. Alger, Sally S. Wong, and Paul Muntner. Heart Disease and Stroke Statistics—2018

- Update: A Report From the American Heart Association. *Circulation*, 137(12), March 2018.
- [77] World Health Organization. *Global status report on noncommunicable diseases 2014*. Number WHO/NMH/NVI/15.1. World Health Organization, Geneva, 2014. Section: Summary in French, Spanish, Arabic, Russian and Chinese published as technical document WHO/NMH/NVI/15.1.
- [78] Stephen M. Salerno, Patrick C. Alguire, and Herbert S. Waxman. Competency in Interpretation of 12-Lead Electrocardiograms: A Summary and Appraisal of Published Evidence. *Annals of Internal Medicine*, 138(9):751, May 2003.
- [79] Patrick Wagner, Nils Strodthoff, Ralf-Dieter Bousseljot, Dieter Kreiseler, Fatima I. Lunze, Wojciech Samek, and Tobias Schaeffter. PTB-XL, a large publicly available electrocardiography dataset. *Scientific Data*, 7(1):154, December 2020.
- [80] M. S. Spach and J. M. Kootsey. The nature of electrical propagation in cardiac muscle. *American Journal of Physiology-Heart and Circulatory Physiology*, 244(1):H3–H22, January 1983.
- [81] Eduardo José da S. Luz, William Robson Schwartz, Guillermo Cámara-Chávez, and David Menotti. ECG-based heartbeat classification for arrhythmia detection: A survey. *Computer Methods and Programs in Biomedicine*, 127:144–164, April 2016.
- [82] Tracy Barill. The six second ECG: A practical guidebook to basic ECG interpretation. *Nursecom Educational Technologies*, 2003.
- [83] David Park and Glenn Fishman. Development and Function of the Cardiac Conduction System in Health and Disease. *Journal of Cardiovascular Development and Disease*, 4(2):7, June 2017.
- [84] Kim E. Barrett and William F. Ganong, editors. *Ganong’s review of medical physiology*. A Lange medical book. McGraw-Hill Med, New York, 24. ed edition, 2012.
- [85] Wai-Kai Chen, editor. *The electrical engineering handbook*. Elsevier Academic Press, Amsterdam ; Boston, 2005. OCLC: ocm56953259.
- [86] V.K. Murthy, T.M. Grove, G.A. Harvey, and L.J. Haywood. Clinical Usefulness Of ECG Frequency Spectrum Analysis. In *The Second Annual Symposium on Computer Application in Medical Care, 1978. Proceedings.*, pages 610–612, 1978.
- [87] N Larburu, T Lopetegi, and I Romero. Comparative study of algorithms for Atrial Fibrillation detection. In *2011 Computing in Cardiology*, pages 265–268, 2011.

- [88] Nitish V. Thakor. From Holter Monitors to Automatic Defibrillators: Developments in Ambulatory Arrhythmia Monitoring. *IEEE Transactions on Biomedical Engineering*, BME-31(12):770–778, December 1984.
- [89] Larisa G. Tereshchenko and Mark E. Josephson. Frequency content and characteristics of ventricular conduction. *Journal of Electrocardiology*, 48(6):933–937, November 2015.
- [90] Paul H. Langner. The Value of High Fidelity Electrocardiography Using the Cathode Ray Oscillograph and an Expanded Time Scale. *Circulation*, 5(2):249–256, February 1952.
- [91] David J. Callans and Mark E. Josephson. *Josephson’s clinical cardiac electrophysiology: techniques and interpretations*. Wolters Kluwer, Philadelphia, 6th edition edition, 2021.
- [92] Hongpo Zhang, Renke He, Honghua Dai, Mingliang Xu, and Zongmin Wang. SS-SWT and SI-CNN: An Atrial Fibrillation Detection Framework for Time-Frequency ECG Signal. *Journal of Healthcare Engineering*, 2020:1–11, May 2020.
- [93] Sandra Śmigiel, Krzysztof Pałczyński, and Damian Ledziński. ECG Signal Classification Using Deep Learning Techniques Based on the PTB-XL Dataset. *Entropy*, 23(9):1121, August 2021.
- [94] Nils Strodthoff, Patrick Wagner, Tobias Schaeffter, and Wojciech Samek. Deep Learning for ECG Analysis: Benchmarks and Insights from PTB-XL. 2020. Publisher: arXiv Version Number: 1.
- [95] Vani P. Sanon, Saurabh Sanon, Rushit Kanakia, Hu Yu, Faris Araj, Rene Oliveros, and Robert Chilton. Hypoglycemia From a Cardiologist’s Perspective: Cardiovascular effects of hypoglycemia. *Clinical Cardiology*, 37(8):499–504, August 2014.
- [96] The ORIGIN Trial Investigators, Linda G. Mellbin, Lars Rydén, Matthew C. Riddle, Jeffrey Probstfield, Julio Rosenstock, Rafael Díaz, Salim Yusuf, and Hertz C. Gerstein. Does hypoglycaemia increase the risk of cardiovascular events? A report from the ORIGIN trial. *European Heart Journal*, 34(40):3137–3144, October 2013.
- [97] Sophia Zoungas, Anushka Patel, John Chalmers, Bastiaan E. de Galan, Qiang Li, Laurent Billot, Mark Woodward, Toshiharu Ninomiya, Bruce Neal, Stephen MacMahon, Diederick E. Grobbee, Andre Pascal Kengne, Michel Marre, and Simon Heller. Severe Hypoglycemia and Risks of Vascular Events and Death. *New England Journal of Medicine*, 363(15):1410–1418, October 2010.
- [98] D. E Bonds, M. E Miller, R. M Bergenstal, J. B Buse, R. P Byington, J. A Cutler, R J. Dudl, F. Ismail-Beigi, A. R Kimel, B. Hoogwerf, K. R Horowitz,

- P. J. Savage, E. R. Seaquist, D. L. Simmons, W. I. Sivitz, J. M. Speril-Hillen, and M. E. Sweeney. The association between symptomatic, severe hypoglycaemia and mortality in type 2 diabetes: retrospective epidemiological analysis of the ACCORD study. *BMJ*, 340(jan08 1):b4909–b4909, January 2010.
- [99] Elaine Chow, Alan Bernjak, Scott Williams, Robert A. Fawdry, Steve Hibbert, Jenny Freeman, Paul J. Sheridan, and Simon R. Heller. Risk of Cardiac Arrhythmias During Hypoglycemia in Patients With Type 2 Diabetes and Cardiovascular Risk. *Diabetes*, 63(5):1738–1747, May 2014.
- [100] Peter Novodvorsky, Alan Bernjak, Elaine Chow, Ahmed Iqbal, Lianne Sellors, Scott Williams, Robert A. Fawdry, Bhavin Parekh, Richard M. Jacques, Jefferson L.B. Marques, Paul J. Sheridan, and Simon R. Heller. Diurnal Differences in Risk of Cardiac Arrhythmias During Spontaneous Hypoglycemia in Young People With Type 1 Diabetes. *Diabetes Care*, 40(5):655–662, May 2017.
- [101] Pejman. Farhadi Ghalati, Moein. E. Samadi, Paul. Balfanz, Dirk Müller-Wieland, Stephan Jonas, Andreas Napp, Christoph Wanner, Markus Ketteler, Athina Vassiliadou, Stefan Heidenreich, Thomas Deserno, Gudrun Hetzel, Danilo Fliser, Malte Kelm, Jürgen Floege, Nikolaus Marx, and Andreas Schuppert. Monitoring Individualized Glucose Levels Predicts Risk for Bradycardia in Type 2 Diabetes Patients with Chronic Kidney Disease: a Pilot Study. *Scientific Reports*, 14:30290, December 2024.
- [102] David E Vaillancourt and Karl M Newell. Changing complexity in human behavior and physiology through aging and disease. *Neurobiology of Aging*, 23(1):1–11, January 2002.
- [103] Fred Shaffer and J. P. Ginsberg. An Overview of Heart Rate Variability Metrics and Norms. *Frontiers in Public Health*, 5:258, September 2017.
- [104] Fred Shaffer, Rollin McCraty, and Christopher L. Zerr. A healthy heart is not a metronome: an integrative review of the heart’s anatomy and heart rate variability. *Frontiers in Psychology*, 5, September 2014.
- [105] Marcus Karlsson, Rolf Hörnsten, Annika Rydberg, and Urban Wiklund. Automatic filtering of outliers in RR intervals before analysis of heart rate variability in Holter recordings: a comparison with carefully edited data. *BioMedical Engineering OnLine*, 11(1):2, December 2012.
- [106] Ko Keun Kim, Jung Soo Kim, Yong Gyu Lim, and Kwang Suk Park. The effect of missing RR-interval data on heart rate variability analysis in the frequency domain. *Physiological Measurement*, 30(10):1039–1050, October 2009.
- [107] Mitra Mojtahedi, Alexander Skupin, Joseph Zhou, Ivan G. Castaño, Rebecca Y. Y. Leong-Quong, Hannah Chang, Kalliopi Trachana, Alessandro Giuliani,

- and Sui Huang. Cell Fate Decision as High-Dimensional Critical State Transition. *PLOS Biology*, 14(12):1–28, December 2016. Publisher: Public Library of Science.
- [108] Satya Swarup Samal, Jeyashree Krishnan, Ali i Hadizadeh Esfahan, Christoph Lueders, Andreas Weber, and Ovidiu Radulescu. Metastable regimes and tipping points of biochemical networks with potential applications in precision medicine. *bioRxiv*, 2018. Publisher: Cold Spring Harbor Laboratory.
- [109] Christophe Trefois, Paul MA Antony, Jorge Goncalves, Alexander Skupin, and Rudi Balling. Critical transitions in chronic disease: transferring concepts from ecology to systems medicine. *Current Opinion in Biotechnology*, 34:48–55, August 2015.
- [110] Vishwesha Guttal and Ciriya Jayaprakash. Changing skewness: an early warning signal of regime shifts in ecosystems. *Ecology Letters*, 11(5):450–460, May 2008.
- [111] V. Dakos, M. Scheffer, E. H. van Nes, V. Brovkin, V. Petoukhov, and H. Held. Slowing down as an early warning signal for abrupt climate change. *Proceedings of the National Academy of Sciences*, 105(38):14308–14312, September 2008.
- [112] Vasilis Dakos, Egbert H. van Nes, Raúl Donangelo, Hugo Fort, and Marten Scheffer. Spatial correlation as leading indicator of catastrophic shifts. *Theoretical Ecology*, 3(3):163–174, August 2010.
- [113] S. R. Carpenter and W. A. Brock. Rising variance: a leading indicator of ecological transition: Variance and ecological transition. *Ecology Letters*, 9(3):311–318, March 2006.
- [114] Luonan Chen, Rui Liu, Zhi-Ping Liu, Meiyi Li, and Kazuyuki Aihara. Detecting early-warning signals for sudden deterioration of complex diseases by dynamical network biomarkers. *Scientific reports*, 2:342, 2012. Publisher: Nature Publishing Group.
- [115] T. M. Lenton, V. N. Livina, V. Dakos, E. H. Van Nes, and M. Scheffer. Early warning of climate tipping points from critical slowing down: Comparing methods to improve robustness. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 370(1962):1185–1204, 2012.
- [116] Pei Chen, Rui Liu, Yongjun Li, and Luonan Chen. Detecting critical state before phase transition of complex biological systems by hidden Markov model. *Bioinformatics*, 32(14):2143–2150, July 2016.
- [117] Sanne M W Gijzel, Ingrid A van de Leemput, Marten Scheffer, Geert E A van Bon, Vivian Weerdesteyn, Thijs M H Eijsvogels, Maria T E Hopman, Marcel G M Olde Rikkert, and René J F Melis. Dynamical Indicators of Resilience in Postural Balance Time Series Are Related to Successful Aging in

- High-Functioning Older Adults. *The Journals of Gerontology: Series A*, July 2018.
- [118] Rui Liu, Pei Chen, Kazuyuki Aihara, and Luonan Chen. Identifying early-warning signals of critical transitions with strong noise by dynamical network markers. *Scientific Reports*, 5(1), December 2015.
- [119] C Fleischmann, M Hartmann, Cs Hartog, T Welte, S Heublein, D Thomas-Rueddel, U Dennler, and K Reinhart. EPIDEMIOLOGY OF SEPSIS IN GERMANY: INCIDENCE, MORTALITY AND ASSOCIATED COSTS OF CARE 2007-2013. *Intensive Care Medicine Experimental*, 3(Suppl 1):A50, 2015.
- [120] Poushali Bhattacharjee, Dana P. Edelson, and Matthew M. Churpek. Identifying Patients With Sepsis on the Hospital Wards. *Chest*, 151(4):898–907, April 2017.
- [121] Mervyn Singer, Clifford S. Deutschman, Christopher Warren Seymour, Manu Shankar-Hari, Djillali Annane, Michael Bauer, Rinaldo Bellomo, Gordon R. Bernard, Jean-Daniel Chiche, Craig M. Coopersmith, Richard S. Hotchkiss, Mitchell M. Levy, John C. Marshall, Greg S. Martin, Steven M. Opal, Gordon D. Rubenfeld, Tom van der Poll, Jean-Louis Vincent, and Derek C. Angus. The Third International Consensus Definitions for Sepsis and Septic Shock (Sepsis-3). *JAMA*, 315(8):801, February 2016.
- [122] David F. Gaieski, Mark E. Mikkelsen, Roger A. Band, Jesse M. Pines, Richard Massone, Frances F. Furia, Frances S. Shofer, and Munish Goyal. Impact of time to antibiotics on survival in patients with severe sepsis or septic shock in whom early goal-directed therapy was initiated in the emergency department\*. *Critical Care Medicine*, 38(4):1045–1053, April 2010.
- [123] Anand Kumar, Daniel Roberts, Kenneth E. Wood, Bruce Light, Joseph E. Parrillo, Satendra Sharma, Robert Suppes, Daniel Feinstein, Sergio Zanotti, Leo Taiberg, David Gurka, Aseem Kumar, and Mary Cheang. Duration of hypotension before initiation of effective antimicrobial therapy is the critical determinant of survival in human septic shock\*. *Critical Care Medicine*, 34(6):1589–1596, June 2006.
- [124] Katharine E. Henry, David N. Hager, Peter J. Pronovost, and Suchi Saria. A targeted real-time early warning score (TREWScore) for septic shock. *Science Translational Medicine*, 7(299):299ra122–299ra122, August 2015.
- [125] Shannon Mitchell, Katherine Schinkel, Yifeng Song, Yuanbo Wang, John Ainsworth, Travis Halbert, Stephen Strong, Jinghe Zhang, Christopher C. Moore, and Laura E. Barnes. Optimization of sepsis risk assessment for ward patients. pages 107–112. IEEE, April 2016.

- [126] M. Carrara, G. Baselli, and M. Ferrario. Mortality prediction in septic shock patients: Towards new personalized models in critical care. pages 2792–2795. IEEE, August 2015.
- [127] Collin H H Tang, Paul M Middleton, Andrey V Savkin, Gregory S H Chan, Sarah Bishop, and Nigel H Lovell. Non-invasive classification of severe sepsis and systemic inflammatory response syndrome using a nonlinear support vector machine: a preliminary study. *Physiological Measurement*, 31(6):775–793, June 2010.
- [128] Michael Rothman, Mitchell Levy, R. Philip Dellinger, Stephen L. Jones, Robert L. Fogerty, Kirk G. Voelker, Barry Gross, Albert Marchetti, and Joseph Beals. Sepsis as 2 problems: Identifying sepsis at admission and predicting onset in the hospital using an electronic medical record-based acuity score. *Journal of Critical Care*, 38:237–244, April 2017.
- [129] Shameek Ghosh, Jinyan Li, Longbing Cao, and Kotagiri Ramamohanarao. Septic shock prediction for ICU patients via coupled HMM walking on sequential contrast patterns. *Journal of Biomedical Informatics*, 66:19–31, February 2017.
- [130] R. Dennis Cook. Detection of Influential Observation in Linear Regression. *Technometrics*, 19(1):15–18, 1977. Publisher: Taylor & Francis.
- [131] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.
- [132] Raffaella Giacomini and Barbara Rossi. Detecting and predicting forecast breakdowns. *Review of Economic Studies*, 76(2):669–705, 2009.
- [133] Alistair E. W. Johnson, Tom J. Pollard, Lu Shen, Li-wei H. Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G. Mark. MIMIC-III, a freely accessible critical care database. *Scientific Data*, 3:sdata201635, May 2016.
- [134] Edward Abraham. New definitions for sepsis and septic shock: continuing evolution but with much still to be done. *Jama*, 315(8):757–759, 2016.
- [135] World Health Organization. ICD-10: The ICD-10 Classification of Mental and Behavioural Disorders: diagnostic criteria for research. In *ICD-10: the ICD-10 classification of mental and behavioural disorders: diagnostic criteria for research*. 1993.
- [136] National Center for Health Statistics (US). *The International Classification of Diseases: 9th Revision, Clinical Modification: ICD-9-CM*. 1991.
- [137] Jean-Louis Vincent, Arnaldo De Mendonça, Francis Cantraine, Rui Moreno, Jukka Takala, Peter M. Suter, Charles L. Sprung, Francis Colardyn, and Serge

- Blecher. Use of the SOFA score to assess the incidence of organ dysfunction/failure in intensive care units: results of a multicenter, prospective study. *Critical care medicine*, 26(11):1793–1800, 1998.
- [138] J.-L. Vincent, Rui Moreno, Jukka Takala, Sheila Willatts, Arnaldo De Mendonça, Hajo Bruining, C. K. Reinhart, Peter M Suter, and L. G. Thijs. *The SOFA (Sepsis-related Organ Failure Assessment) score to describe organ dysfunction/failure*. Springer, 1996.
- [139] Andrew C Harvey. *Forecasting, structural time series models and the Kalman filter*. Cambridge university press, 1990.
- [140] Marie Auger-Méthé, Chris Field, Christoffer M. Albertsen, Andrew E. Derocher, Mark a. Lewis, Ian D. Jonsen, and Joanna Mills Flemming. State-space models’ dirty little secrets: even simple linear Gaussian models can have estimation problems. *Scientific Reports*, 6(May):26677, 2016.
- [141] Donald W. K. Andrews. Tests for Parameter Instability and Structural Change With Unknown Change Point. *Econometrica*, 61(4):821–856, 1993. Publisher: [Wiley, Econometric Society].
- [142] Gregory C. Chow. Tests of Equality Between Sets of Coefficients in Two Linear Regressions. *Econometrica*, 28(3):591–605, 1960. Publisher: [Wiley, Econometric Society].
- [143] Marten Scheffer, J. Elizabeth Bolhuis, Denny Borsboom, Timothy G. Buchman, Sanne M. W. Gijzel, Dave Goulson, Jan E. Kammenga, Bas Kemp, Ingrid A. van de Leemput, Simon Levin, Carmel Mary Martin, René J. F. Melis, Egbert H. van Nes, L. Michael Romero, and Marcel G. M. Olde Rikkert. Quantifying resilience of humans and other animals. *Proceedings of the National Academy of Sciences*, 115(47):11883–11890, November 2018.
- [144] Elif Izci, Mehmet Akif Ozdemir, Murside Degirmenci, and Aydin Akan. Cardiac Arrhythmia Detection from 2D ECG Images by Using Deep Learning Technique. In *2019 Medical Technologies Congress (TIPTEKNO)*, pages 1–4, Izmir, Turkey, October 2019. IEEE.
- [145] Jodie Ashford, Jordan J. Bird, Felipe Campelo, and Diego R. Faria. Classification of EEG Signals Based on Image Representation of Statistical Features. In Zhaojie Ju, Longzhi Yang, Chenguang Yang, Alexander Gegov, and Dalin Zhou, editors, *Advances in Computational Intelligence Systems*, volume 1043, pages 449–460. Springer International Publishing, Cham, 2020. Series Title: Advances in Intelligent Systems and Computing.
- [146] Isaak Kavasidis, Simone Palazzo, Concetto Spampinato, Daniela Giordano, and Mubarak Shah. *Brain2Image: Converting Brain Signals into Images*. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1809–1817, Mountain View California USA, October 2017. ACM.

- 
- [147] Muhammet Fatih Aslan, Kadir Sabanci, and Akif Durdu. A CNN-based novel solution for determining the survival status of heart failure patients with clinical record data: numeric to image. *Biomedical Signal Processing and Control*, 68:102716, July 2021.