COLLECTIVE ANOMALY DETECTION IN TIME SERIES USING PITCH FREQUENCY AND DISSIMILARITY FEATURES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

EKIN CAN ERKUŞ

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
BIOMEDICAL ENGINEERING

JUNE 2023

Approval of the thesis:

COLLECTIVE ANOMALY DETECTION IN TIME SERIES USING PITCH FREQUENCY AND DISSIMILARITY FEATURES

submitted by **EKIN CAN ERKUŞ** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Biomedical Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Vilda Purutçuoğlu
Head of Department, **Biomedical Engineering**

Prof. Dr. Vilda Purutçuoğlu
Supervisor, **Department of Statistics, METU**

Assist. Prof. Dr. Aykut Eken
Co-supervisor, **Dept. of Biomedical Engineering, TOBB ETU**

Examining Committee Members:

Prof. Dr. Çağdaş Hakan Aladağ
Department of Statistics, Hacettepe University

Prof. Dr. Vilda Purutçuoğlu
Department of Statistics, METU

Assoc. Prof. Dr. Yeşim Serinağaoğlu Doğrusöz
Department of Electrical and Electronics Engineering, METU

Assoc. Prof. Dr. Tolga Çukur
Dept. of Electrical and Electronics Eng., Bilkent University

Assoc. Prof. Dr. Yeşim Aydın Son
Department of Health Informatics, METU

Date:12.06.2023

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Ekin Can Erkuş

Signature :

ABSTRACT

COLLECTIVE ANOMALY DETECTION IN TIME SERIES USING PITCH FREQUENCY AND DISSIMILARITY FEATURES

Erkuş, Ekin Can
Ph.D., Department of Biomedical Engineering
Supervisor: Prof. Dr. Vilda Purutçuoğlu
Co-Supervisor: Assist. Prof. Dr. Aykut Eken

June 2023, 203 pages

Collective anomalies appear in the majority of time series data modalities due to a variety of factors. They appear frequently in biomedical signals as a result of electrode displacement, motion, or faulty equipment. These anomalies have a negative impact on model and analysis performance and are frequently identified in order to be eliminated or detected in order to observe unwanted data behavior.

This thesis describes a novel method for detecting collective anomalies in quasi-periodic time series data. By leveraging pitch frequency estimation techniques commonly used in audio signal processing, the proposed algorithm combines the strengths of both the time and frequency domains. It provides a comprehensive view of anomalous patterns and can be customized and adapted to different domains and datasets, making it useful for a wide range of applications. By employing a sliding windows approach and utilizing previous data information to dynamically learn structural patterns, the proposed algorithm also excels in real-time anomaly detection. It is effective in detecting subject-specific anomalies, although it may not locate single-sample outliers that do not significantly affect window properties. The algorithm was developed specifically for quasi-periodic data and may be limited in its applicability to non-quasi-periodic time series data.

Both synthetically generated and benchmark electrocardiogram (ECG) datasets are used to assess the effectiveness of the proposed algorithm under a variety of conditions. The performance of the proposed approach is compared to other features commonly used in anomaly detection, as well as some benchmark time series anomaly detection algorithms. The findings show that the proposed method consistently outperforms the compared algorithms in detecting both outlier-like and inlier-like anomalies. It also outperforms other non-parametric approaches in terms of computational efficiency.

Keywords: anomaly detection, time series, pitch frequency, dissimilarity, sliding windows

ÖZ

PİTCH FREKANSI VE BENZEŞMEZLİK ÖZNİTELİKLERİ KULLANILARAK ZAMAN SERİLERİNDE KOLEKTİF ANOMALİ TESPİTİ

Erkuş, Ekin Can
Doktora, Biyomedikal Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Vilda Purutçuoğlu
Ortak Tez Yöneticisi: Dr. Öğr. Üyesi. Aykut Eken

Haziran 2023 , 203 sayfa

Kolektif anomaliler, çeşitli faktörler nedeniyle zaman serisi veri modalitelerinin çoğunda görülür. Biyomedikal sinyallerde elektrotların yer değiştirmesi, hareket veya hatalı ekipman nedeniyle sıklıkla görülürler. Bu anomaliler model ve analiz performansı üzerinde olumsuz bir etkiye sahiptir. Bu yüzden, onları veriden kaldırmak veya verideki farklı davranışları yakalamak için tespit edilmeleri önemlidir.

Bu tez, yarı periyodik zaman serisi verilerindeki kolektif anomalileri tespit etmek için yeni bir yaklaşım önermektedir. Önerilen algoritma, ses sinyali işlemede yaygın olarak kullanılan perde frekansı tahmin tekniklerinden yararlanarak hem zaman hem de frekans alanlarının güçlü yönlerini birleştirmektedir. Bu sayede, anomaliler için farklı açılardan değerlendirmeler ile daha iyi bir performans sunması hedeflenektedir. Kayan pencereler yaklaşımını kullanarak ve yapısal örüntüleri dinamik olarak öğrenmek için önceki veri bilgilerini kullanarak, önerilen algoritma gerçek zamanlı anomali tespitinde de kullanılabilmektedir. Pencere özelliklerini önemli ölçüde etkilemeyen tek örnekli aykırı değerleri tespit etmek için geliştirilmemiş olsa da, kolektif anormallikleri tespit etmede etkilidir. Algoritma özellikle yarı periyodik veriler için geliştirilmiştir ve yarı periyodik olmayan zaman serisi verilerine uygulanabilirliği sınırlı olabilir.

Önerilen algoritmanın etkinliğini çeşitli koşullar altında değerlendirmek için hem sentetik olarak oluşturulmuş hem de karşılaştırmalı elektrokardiyogram (EKG) veri kümeleri kullanılmıştır. Önerilen yaklaşımın performansı, anomali tespitinde yaygın olarak kullanılan diğer öznitelikler ve temel anomali tespit algoritmaları ile karşılaştırılmıştır. Bulgular, önerilen yöntemin kolektif anomalileri tespit etmede karşılaştırılan algoritmalardan genelde daha iyi performans gösterdiğini ve ayrıca hesaplama verimliliği açısından diğer parametrik olmayan yaklaşımlardan daha hızlı olduğunu ortaya koymaktadır.

Anahtar Kelimeler: anomali tespiti, zaman serileri, perde frekansı, benzemezlik, kayan pencereler

To my beloved family, whose infinite love and support have been the unwavering anchor that has steadied me throughout my life.

ACKNOWLEDGMENTS

I would like to express my deepest gratitude and appreciation to my thesis advisor, Prof. Dr. Vilda Purutçuoğlu, for her tenacious guidance, invaluable insights, endless patience, and relentless dedication throughout the course of this research. Her exceptional mentoring and encouragement have been critical in pushing the boundaries of my capabilities and providing me with incredible research opportunities. Her academic vision as a valuable researcher influenced my current academic direction. I am eternally grateful for her tireless support and an incredible source of inspiration.

I would also like to express my heartfelt gratitude to my co-advisor, Assist. Prof. Dr. Aykut Eken, for his valuable support with this thesis. I would also like to express my gratitude to Assoc. Prof. Dr. Didem Gökçay, whose visionary perspectives and profound understanding of various aspects have greatly influenced my development and steered me as a researcher. Furthermore, I would like to thank all of the professors who I have attended their lectures, listened to their valuable speeches, read their insightful papers and notes, and admired their academic behavior.

I extend my sincere appreciation to the members of the thesis committee, Assoc. Prof. Dr. Yeşim Serinağaoğlu and Assoc. Prof. Dr. Tolga Çukur, for their relentless support, valuable feedbacks and guidance throughout the entire duration of this thesis. Their expertise and long-standing dedication to their fields have been immensely valuable in shaping this thesis. I would also like to thank other jury members Prof. Dr. Çağdaş Hakan Aladağ and Assoc. Prof. Dr. Yeşim Aydın Son for their participation in the committee and their valuable feedback, which has significantly contributed to the enhancement of this work.

I am indebted to my family members for their immeasurable love and support. Firstly, I would like to express my deepest gratitude to my parents, Adnan and Reyhan Erkuş for their lifelong encouragement, sacrifices in raising me while they wane in strength, and their obstinate belief in me. Their constant support and motivation have been the

driving force behind my accomplishments. And of course, I am incredibly grateful to my spouse, Özlem Kahraman Erkuş for her unending patience, indefatigable support, and love throughout this tiring journey. Her radiant presence and unwavering commitment have provided me with the strength and encouragement to overcome challenges and persist in the troubles I encounter. I am truly grateful for the inexhaustible dedication and love she has bestowed upon me throughout this demanding endeavor. Her understanding and encouragement have been invaluable for this thesis, and my life. I would also like to thank our beloved daughter, Alina whose infinite love has been a constant source of motivation which has given me the strength to persevere. It is my deepest desire that our accomplishments will serve as a guiding light for her future endeavors and the path she chooses to embark upon. Last but not least, I cannot forget to express my appreciation to our energetic and ever-cheerful dog, Meelo.

Finally, I would like to extend my sincere thanks to all my colleagues and researchers who have made significant contributions to the literature in this field. Your work has laid the foundation upon which this thesis is built, and I am grateful for the insights and knowledge you have shared. I am humbled to have had the opportunity to collaborate and learn from such remarkable individuals, and your contributions have undoubtedly enriched my research journey.

I am deeply grateful to everyone who has played a role, whether big or small, in the realization of this thesis. Your guidance, support, and encouragement have been invaluable, and I am profoundly thankful for your contributions to my academic and personal growth.

TABLE OF CONTENTS

ABSTRACT v

ÖZ vii

ACKNOWLEDGMENTS x

TABLE OF CONTENTS xii

LIST OF TABLES xviii

LIST OF FIGURES xxii

CHAPTERS

1 INTRODUCTION 1
  - 1.1 Time Series 1
    - 1.1.1 Periodic Data Behavior 2
    - 1.1.2 Biomedical Time Series 3
  - 1.2 Outliers and Anomalies 3
    - 1.2.1 Types of Outliers and Anomalies 5
      - 1.2.1.1 Type I Anomalies (Point Outliers, High Magnitude Anomalies) 5
      - 1.2.1.2 Type II Anomalies (Contextual Anomalies, Spatial Outliers) 8
      - 1.2.1.3 Type III Anomalies (Collective Outliers) (Anomalies) . 11
    - 1.2.2 Anomalies in Biomedical Data 13

1.3 Anomaly Detection 15
    1.3.1 Real-time Anomaly Detection 16
    1.3.2 Subject-specific Anomaly Detection 18
    1.3.3 Features for Anomaly Detection 19
    1.3.4 Detection of Quasi-periodic Outliers 21
1.4 Motivation, and Possible Contributions to the Literature 22
2 TECHNICAL BACKGROUND 25
2.1 Some Properties of Time Series 25
    2.1.1 Stationarity 26
    2.1.2 Periodicity 27
        2.1.2.1 Quasi-periodicity 27
    2.1.3 Sampling Rate 28
2.2 Frequency Domain and Spectral Analysis 29
    2.2.1 Quefrency 30
    2.2.2 Pitch (Fundamental) Frequency 32
    2.2.3 Pitch Frequency Estimation Methods 33
        2.2.3.1 Normalized Correlation Function (NCF) 33
        2.2.3.2 Pitch Estimation Filter (PEF) 35
        2.2.3.3 Cepstrum Pitch Determination (CEP) 36
        2.2.3.4 Log-Harmonic Summation (LHS) 37
        2.2.3.5 Summation of Residual Harmonics (SRH) 38
2.3 Dissimilarity Metrics and Temporal Similarity Measures 40
    2.3.1 Linear Dissimilarity Metrics 40

2.3.2 Dynamic Time Warping (DTW) 41

2.3.3 Temporal Similarity Measures 42

2.4 Machine Learning Classifiers 43

2.4.1 K-Means Clustering 44

2.4.2 K-Nearest Neighbour (k-NN) Classifier 44

2.4.3 Decision Trees 45

2.4.4 Support Vector Machines (SVM) 46

2.4.5 Naive Bayes Classifier 47

2.5 Validation and Performance Evaluation 47

2.5.1 Confusion Matrix and Common Performance Metrics 48

2.5.1.1 Sensitivity (Recall or True Positive Rate) 48

2.5.1.2 Specificity (True Negative Rate) 49

2.5.1.3 Precision 49

2.5.1.4 Accuracy 50

2.5.1.5 F1 Score 50

2.5.1.6 Matthew's Correlation Coefficient (MCC) 51

2.5.2 K-Fold Cross Validation 51

3 RELATED ANOMALY DETECTION STUDIES AND RECENT DEVELOPMENTS 53

3.1 Common Outlier and Anomaly Detection Methods in Literature . . . 53

3.1.1 Statistical Approaches 53

3.1.1.1 z-score Test 54

3.1.1.2 Box-plot Test 55

3.1.1.3 Grubbs Method 56

3.1.2 Frequency Domain Related Approaches 56

3.1.2.1 Spectral Residual (SR) Method 57

3.1.2.2 Singular Spectrum Analysis (SSA) 57

3.1.2.3 Autoencoder-based Anomaly Detection 58

3.1.2.4 Power Spectral Density (PSD) Analysis 59

3.1.2.5 Hilbert Transform-based Methods 59

3.1.2.6 Cepstrum Analysis 60

3.1.2.7 Wavelet-based Methods 60

3.1.3 Dissimilarity-based Approaches 61

3.1.3.1 Higher Order Time Series Symbolic Aggregate Approximation (HOTSAX) 61

3.1.3.2 LDOF (Local Distance-based Outlier Factor) 62

3.1.3.3 Autonomous Anomaly Detection (AAD) Method 63

3.2 Frequency Domain Based Outlier Detection (FOD) Algorithm 64

3.2.1 Single-step FOD Algorithm and Applications 65

3.2.2 Simulations to measure FOD algorithm performances 67

3.2.3 Conclusion and Problems of the FOD Algorithm 71

3.2.3.1 Problem with the FOD Algorithm: Thresholding 71

3.3 Windowed Frequency Domain Based Outlier Detection (WFOD) . . 72

3.3.1 Evaluation of the WFOD Feature Performances and Results . . 76

3.3.2 Conclusion and Problems of the WFOD Algorithm 80

3.3.3 Problem with the WFOD Algorithm: Grand Average Alignment 83

4 THE PROPOSED APPROACH 85

4.1 Initialization of the Sliding Window Approach 85

4.2 The Pitch Frequency Estimation 88

4.3 The Relationship Between Frequency and Time Domain 88

4.4 Data Partition and Grand Averaging 90

4.5 Dissimilarity Computation as Feature Extraction 92

4.6 Clustering for the Decision of Anomaly 95

4.7 The Proposed Algorithm Pseudocode 96

4.8 The Time Complexity 96

5 APPLICATIONS, RESULTS AND DISCUSSION 101

5.1 Synthetic Data Analyses 101

5.1.1 Synthetic Data Analysis 1: Comparison of Feature Extraction Methods 102

5.1.1.1 Synthetic Dataset 1 Generation 102

5.1.1.2 Other Benchmark Features to Compare 103

5.1.1.3 Results and Discussion for Synthetic Data Analysis 1 . 106

Univariate Classification Accuracies: 109

Multivariate Classification Accuracies: 113

5.1.2 Synthetic Data Analysis 2: Comparing the Moving Windows Parameters 114

5.1.2.1 Synthetic Dataset 2 Generation 114

5.1.2.2 Experimental Setup and Feature Extraction 117

5.1.2.3 Clustering and Decision Making 119

5.1.2.4 Results and Discussion for Synthetic Data Analysis 2 . 120

5.2 Real Data Analyses 125

5.2.1 Real Data Analysis 1: Introducing Displacement Noise to an Interval 125

5.2.1.1 Real Datasets and Experimental Design 125

5.2.1.2 Results and Discussion for Real Data Analysis 1 129

5.2.2 Real Data Analysis 2: Frequency Search Range Effect on Pre-annotated Benchmark ECG Datasets 134

5.2.2.1 Real Datasets and the Usage of the Annotated Data . . . 135

5.2.2.2 Results and Discussion for Real Data Analysis 2 136

6 CONCLUSION AND FUTURE WORKS 139

6.1 Conclusion 139

6.2 Future Works 141

REFERENCES 143

A APPENDIX 177

A.1 Extension Works 177

A.1.1 Comparison of Dissimilarity Metrics as Features 177

A.1.2 Comparison of Pitch Frequency Algorithms 189

A.2 The Classification Graphical User Interface (GUI) 198

B CURRICULUM VITAE 199

LIST OF TABLES

TABLES

Table 2.1 Common dissimilarity metrics and their formulation. 40

Table 2.2 Confusion matrix for two classes 48

Table 3.1 The average F1-scores for the effects of data length on the performance of the outlier detection algorithms 68

Table 3.2 The average F1-score values for the effects of baseline data distribution and outlier fold change. 69

Table 3.3 The average F1-scores for the effects of outlier percentage in the data. 70

Table 3.4 The average computational time (milliseconds) for each iteration for data length effect. 70

Table 3.5 Classification accuracy of statistical feature pairs with and without WFOD feature for standing - walking experiment under distinct classification methods. 80

Table 3.6 Classification accuracy of statistical feature pairs with and without WFOD feature for standing - jumping experiment under distinct classification methods. 81

Table 3.7 Classification accuracy of statistical feature pairs with and without WFOD feature for walking - jumping experiment under distinct classification methods. 82

Table 4.1 The dissimilarity metrics applied in the proposed algorithm and their formulations. 93

Table 4.2 The formulations of the time-dependent similarity metrics applied in the proposed algorithm under the WSS assumption. 93

Table 4.3 Preliminary F1-score values of the proposed algorithm for different univariate dissimilarity metrics, obtained using the toy data. 94

Table 4.4 The average computational times for randomly generated datasets consisting of 100 subjects in each with fixed length under Gaussian Normal Distribution. 98

Table 4.5 The average computational times for real benchmark dataset MIT-BIH Malignant Ventricular Ectopy with cropped data lengths. 98

Table 5.1 The list of simulated datasets. 103

Table 5.2 The compilation of feature groups and their corresponding features utilized in this investigation. 108

Table 5.3 The accuracy results of the proposed algorithm with the NFC pitch frequency detection method. Abbreviations: G.: Gaussian, E.: Exponential, L.: Log-normal, S.: Sinusoidal, A.: Amplitude. 109

Table 5.4 The accuracy results of the proposed algorithm with the PEFAC pitch frequency detection method. Abbreviations: G.: Gaussian, E.: Exponential, L.: Log-normal, S.: Sinusoidal, A.: Amplitude. 110

Table 5.5 The accuracy results of the statistical features. Abbreviations: G.: Gaussian, E.: Exponential, L.: Log-normal, S.: Sinusoidal, A.: Amplitude. 110

Table 5.6 The accuracy results of the spectral features. Abbreviations: G.: Gaussian, E.: Exponential, L.: Log-normal, S.: Sinusoidal, A.: Amplitude. 111

Table 5.7 The accuracy results of the DCT coefficient features. Abbreviations: G.: Gaussian, E.: Exponential, L.: Log-normal, S.: Sinusoidal, A.: Amplitude. 112

Table 5.8 The multivariate classification accuracies of feature groups. Abbreviations: G.: Gaussian, E.: Exponential, L.: Log-normal, S.: Sinusoidal, A.: Amplitude. 114

Table 5.9 Comparison of accuracies for window size using the Euclidean dissimilarity metric, with a slide size of 1000. 'B' refers to baseline, 'Sinus' abbreviates sinusoidal, 'A' stands for anomalous windows, 'SNR' refers to signal-to-noise ratio, 'Fs' is the sampling rate. 122

Table 5.10 Comparison of accuracies for different slide sizes using the Euclidean dissimilarity metric, with a window size of 1000. 'B' refers to baseline, 'Sinus' abbreviates sinusoidal, 'A' stands for anomalous windows, 'SNR' refers to signal-to-noise ratio, 'Fs' is the sampling rate. . . . 122

Table 5.11 Comparison of accuracies for different dissimilarity metrics with a window size of 1000 and a slide size of 1000. 'B' refers to baseline, 'Sinus' abbreviates sinusoidal, 'A' stands for anomalous windows, 'SNR' refers to signal-to-noise ratio, 'Fs' is the sampling rate. 124

Table 5.12 Comparison of average computational times (in s) of dissimilarity metrics for the number of windows. 124

Table 5.13 The performance results of the methods for the modified MIT-BIT Long Term ECG dataset. *MCC* stands for Matthew's Correlation Coefficient. 131

Table 5.14 The performance results of the methods for the modified MIT-BIT Normal Sinus ECG dataset. *MCC* stands for Matthew's Correlation Coefficient. 132

Table 5.15 The performance results of the methods for the modified European ST-T Change ECG dataset. *MCC* stands for Matthew's Correlation Coefficient. 133

Table 5.16 The relative average computational times corresponding to the proposed algorithm through the simulations. 134

Table 5.17 The frequency range effect of the pitch frequency detection using the CEP algorithm on real annotated benchmark ECG datasets. 136

Table A.1 The classification results of synthetic ECG database with 60 to 110 bpm. 187

Table A.2 The classification results of synthetic ECG database with R peak amplitude of 1000 to 2000 units. 188

Table A.3 The classification results of synthetic ECG database with QRS width of 0.07 seconds to 0.12 seconds. 189

Table A.4 The classification results of synthetic ECG database with t wave amplitude of 500 to 1000 units. 190

Table A.5 Synthetic Dataset: S1, Univariate and multivariate results. 'A' stands for accuracy, and 't' is for computational time. 192

Table A.6 Synthetic Dataset: S2, Univariate and multivariate results. 'A' stands for accuracy, and 't' is for computational time. 193

Table A.7 Synthetic Dataset: S3, Univariate and multivariate results. 'A' stands for accuracy, and 't' is for computational time. 194

Table A.8 Synthetic Dataset: S4, Univariate and multivariate results. 'A' stands for accuracy, and 't' is for computational time. 196

Table A.9 Real Dataset: MIT-BIH Long Term ECG, Univariate and multivariate results. 'A' stands for accuracy, and 't' is for computational time. . . . 197

LIST OF FIGURES

FIGURES

Figure 1.1 Example representations of type I anomalies. 7

Figure 1.2 Example data generated using Gaussian Normal distribution, and a low-amplitude outlier placed at the 80th sample. 8

Figure 1.3 ECG data with a motion artifact in the form of a type II outlier sequence. 9

Figure 1.4 ECG data with ventricular tachyarrhythmia as a type III outlier sequence (red). 12

Figure 1.5 Randomly generated time-domain data with periodic outliers placed per 50 samples with random amplitudes. 21

Figure 1.6 The frequency domain estimation of the same data in Figure 1.5, using DFT transformation. 22

Figure 2.1 An illustration of the pitch frequency at 225.684Hz (bottom graph), obtained by the DFT of the upper sound data. 32

Figure 3.1 A representation of the box-plot graph. The red crosses denote the outliers. 55

Figure 3.2 A representation of the windowed data (Blue) and their WFOD Transform (Red) on real ECG data from BIDMC Congestive Heart Failure database 75

Figure 3.3 A representation of the windowing process and the WFOD features on real ECG data from BIDMC Congestive Heart Failure database. 77

Figure 3.4 Visualization of standing-jumping condition pairs in a concatenated dataset from the motion artifact-contaminated ECG dataset. . . . 79

Figure 3.5 Computation of the grand average by cropping the structures as shown in Figure 4.4. 83

Figure 3.6 Grand average pattern for quasi-periodic data consists of two epochs with the same amplitudes and different periods. 84

Figure 3.7 Grand average pattern for quasi-periodic data consists of two epochs with the same periods and different amplitudes. 84

Figure 4.1 Flowchart illustrating the proposed approach. 86

Figure 4.2 Illustration of window size and slide size. 87

Figure 4.3 The relationship between f_0 and t_{int} values. 89

Figure 4.4 Formation of initial sequential partitions with a length of t_{int}. . . 91

Figure 4.5 Computation of the grand average of partitions obtained from Figure 4.4. 92

Figure 5.1 Simulated datasets, part 1. 104

Figure 5.2 Simulated datasets, part 2. 105

Figure 5.3 Simulated datasets, part 3. 106

Figure 5.4 Simulated datasets, part 4. 107

Figure 5.5 Generated dataset 1 (upper) and zoomed in (lower). Baseline: Gaussian with $\mu = 0$ and $\sigma^2 = 1$. Anomalous: Gaussian with $\mu = 3$ and $\sigma^2 = 1$. 115

Figure 5.6 Generated dataset 2 (upper) and zoomed in (lower). Baseline: Sinusoidal with amplitude = 1 and sampling rate = 5 Hz. Anomalous: Sinusoidal with amplitude = 3 and sampling rate = 5 Hz. 116

Figure 5.7 Generated dataset 3 (upper) and zoomed in (lower). Baseline: Sinusoidal with amplitude = 1 and sampling rate = 5 Hz. Anomalous: Sinusoidal with amplitude = 1 and sampling rate = 15 Hz. 117

Figure 5.8 Snapshot of the real-time operation in a single MATLAB figure. 121

Figure 5.9 Snapshot of the real-time operation in a single MATLAB figure where a pattern observed in the grand average analysis. 123

Figure 5.10 Sample data from the MIH-BIH Long Term ECG dataset and examples of the modified data from this dataset. 127

Figure 5.11 Sample data from the MIT-BIH normal sinus rhythm dataset and examples of the modified data from this dataset. 128

Figure 5.12 Sample data from the European ST-T change dataset and examples of the modified data from this dataset. 128

Figure A.1 Synthetic ECG data, illustrating arrhythmia with 60 and 110 bpm values, generated under SNR of 10dB. 179

Figure A.2 Synthetic ECG data, illustrating type I outliers with 1000 and 2000 units in R peak amplitudes, generated under SNR of 10dB. 180

Figure A.3 Synthetic ECG data, illustrating narrowed and extended duration of QRS structures with 0.07 and 0.12 seconds values, generated under SNR of 10dB. 181

Figure A.4 Synthetic ECG data, illustrating t-wave abnormality with 500 and 1000 units in amplitude, generated under SNR of 10dB. 182

Figure A.5 Synthetic ECG data, illustrating arrhythmia with 60 and 110 bpm values, generated under SNR of -10dB. 183

Figure A.6 Synthetic ECG data, illustrating type I outliers with 1000 and 2000 units in R peak amplitudes, generated under SNR of -10dB. 184

Figure A.7 Synthetic ECG data, illustrating narrowed and extended duration of QRS structures with 0.07 and 0.12 seconds values, generated under SNR of -10dB. 185

Figure A.8 Synthetic ECG data, illustrating t-wave abnormality with 500 and 1000 units in amplitude, generated under SNR of -10dB. 186

Figure A.9 A snapshot of the GUI that is made for analyses. 198

CHAPTER 1

INTRODUCTION

This chapter provides a brief introduction to the general concepts, describes the main problems, and provides the motivation along with the aims of the thesis with a possible contribution to the literature.

1.1 Time Series

Time series refers to a sequence of consecutive samples of data that are collected at regular intervals over a span of time, and usually recorded at a fixed sampling rate [1]. Each data sample has a specific timestamp, which reveals the relationship between different samples and allows the analysis of the behavior and patterns exhibited by the data as time progresses [2]. This temporal dependence indicates that each data sample is impacted by its preceding values and may have a causal dependence on them [3]. Consequently, the posterior patterns in the data can be analyzed and by using this information, new models can be developed that can forecast future values [4]. By utilizing time series analyses, the hidden patterns, trends, and causal relationships within the data, and the related systems can also be examined.

While numerous disciplines such as engineering, medicine, finance, economics, and meteorology heavily rely on time series data, this thesis specifically concentrates on the field of biomedical engineering. In this domain, time series data are typically obtained from biological sources or subjects, using specialized equipment or software that adheres to a fixed sampling rate [5]. Analyzing these biomedical time series modalities can help with disease or disorder diagnosis, patient monitoring for anomalous body responses, and prediction of overall health conditions [6, 7]. Moreover, biomedical time series data often display periodic behavior, referred to as seasonality, where recurring patterns occur in a predictable manner [8].

1.1.1 Periodic Data Behavior

The existence of repeating patterns that appear at regular intervals is known as periodicity, and it is a fundamental property of many time series data [9]. Periodicity enables the identification and analysis of cyclic phenomena which uncovers underlying patterns, trends, and relationships within the data [2]. The process of accurate forecasting, anomaly detection, and decision-making is made possible by the analysis of the periodicity in time series, which includes the discovery of dominant frequencies, harmonics, and seasonality patterns [10, 11]. Apart from the ideal periodic behavior, biomedical time series data frequently exhibits quasi-periodicity, an intriguing phenomenon that displays patterns resembling periodicity but lacking strict regularity [12, 13]. In other words, contrary to periodicity, where patterns repeat at regular intervals, quasi-periodic patterns happen at irregular or varying intervals. Other than biological data, ecological systems, mechanical systems, and time-dependent data from social sciences are also some examples that frequently exhibit this quasi-periodic characteristic because their ideal periodicity is also influenced by a variety of factors and interactions [14, 15, 16].

Quasi-periodic anomalies, which appear in datasets exhibiting quasi-periodic behavior, are commonly observed in various domains, including biology, mechanics, and seasonal measurements [17]. Detecting such anomalies in time series data has been the subject of extensive research, leading to a variety of approaches. These approaches involve techniques like frequency domain transformation, modeling, and the use of sliding windows [18]. While some studies claim superior performance of their methods, it is important to recognize that the effectiveness of these approaches depends on factors such as dataset characteristics, application settings, and parameter selection [19, 20]. Therefore, choosing the right approach to detect quasi-periodic anomalies requires careful consideration of these contextual elements [21].

1.1.2 Biomedical Time Series

Biological data contributes to a wide range of time series data types, which are typically collected using specialized equipment with unique specifications [22]. As a result, biomedical signals take on various forms to represent different aspects of an organism's biological conditions. These signals often exhibit quasi-periodic behavior and have distinct characteristics across different data modalities [23]. However, working with quasi-periodic time series data presents challenges due to issues like device malfunctions, noise interference, or missing data intervals. Therefore, detecting anomalies in quasi-periodic medical data is essential for identifying deviations from expected patterns, which may indicate underlying health issues or abnormalities requiring further investigation and intervention [24].

Biomedical signals encompass a diverse range of types, serving as representations of various aspects of an organism's biological conditions [25]. These signals often exhibit regular periodic behavior, exhibiting distinctive characteristics across different data modalities [26]. It is crucial to acknowledge that the behavior of biological systems can vary due to factors such as disorders, age, gender, and physical conditions of the subject while excluding environmental factors and artifacts [27]. When controlling for other factors, accurately diagnosing disorders within subject groups becomes a critical component of the treatment process [28]. Consequently, achieving precise diagnoses relies on the recognition and detection of variations in the data behavior [29]. In the field of time series biomedical data analysis, most data types demonstrate quasi-periodic behavior, influenced by body rhythms like heartbeats, circadian rhythm, temperature dependence, and electromagnetic interference [30]. Analyzing and comprehending these quasi-periodic patterns can yield valuable information about the functioning of biological systems and facilitate the identification of abnormalities or anomalies [31].

1.2 Outliers and Anomalies

Outliers are data samples that deviate considerably from the anticipated or standard range of values in a dataset, while anomalies encompass any atypical or abnormal behavior or pattern in the data that does not adhere to the anticipated or standard patterns [32, 33]. Although the majority of the researchers claim to know the concept of anomalies and outliers, their definitions drastically differ across different research fields, and they are often categorized by different names based on their types. Statistically, an outlier is defined as a sample with a significantly larger amplitude than the rest of the data [34]. Therefore, in definition, outliers are generally considered as the point anomalies with significantly different amplitudes than the rest of the data baseline, which may statistically change the instantaneous characteristics of the data [35]. On the other hand, anomalies can appear in various forms, such as individual samples, groups of samples, or specific data intervals, exhibiting distinct characteristics compared to the rest of the data [36].

Anomalies disrupt the expected behavior of time series data and can obscure valuable information within the dataset [37]. These anomalies can arise from various sources such as motion artifacts, non-systematic noises, missing values, or the combination of multiple data sources, often impacting specific time intervals and diminishing the effectiveness of data analysis techniques [38]. Therefore, it is crucial to identify and remove these anomalies prior to conducting the main analyses. Extensive research has been conducted on anomaly and outlier detection, resulting in the development of specialized approaches for time series data [39, 40, 41, 42]. Anomalies can manifest within single intervals or exhibit irregular or quasi-periodic patterns in the data. Quasi-periodic anomalies, in particular, may exhibit different probabilistic distributions compared to the baseline data [43]. Consequently, detecting anomalies often requires prior knowledge of the examined data, emphasizing the importance of understanding the behavior of the baseline data. While many studies focus on directly detecting anomalies from the given data, some approaches leverage the distributional information of the baseline data to enhance the accuracy of anomaly detection [44, 45, 46]. Each data modality demonstrates its unique behavior and distinct types of anomalies, underscoring the significance of selecting the most appropriate outlier detection algorithm with minimal false positive rates [47].

In the context of biomedical data modalities, both outliers, and anomalies can arise due to instrumentation errors, environmental noise, subject conditions, or artifacts caused by motion and the characteristics of the data [48]. Detecting and managing outliers is crucial during the preprocessing stage of biomedical data as they can significantly impact the efficiency of subsequent analyses [49]. Furthermore, in computer vision, artificial intelligence, and biomedical applications, anomaly detection plays a pivotal role in offline and real-time data processing, contributing to decision-making, disorder diagnosis, and live monitoring [50].

1.2.1 Types of Outliers and Anomalies

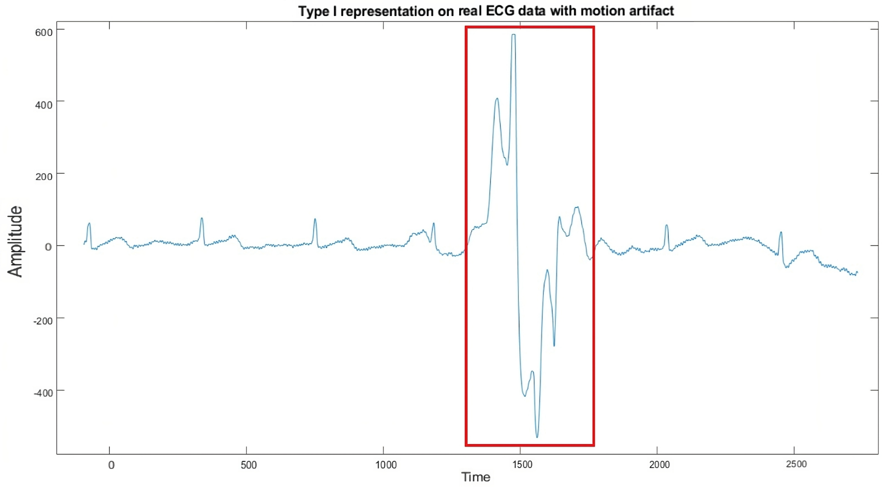
The presence of faults in data manifests in various forms, including point outliers or collective anomalies comprised of a series of outlying samples [51]. Additionally, certain anomalies can manifest as unexpected sequences within the data, creating a distinctive outlying structure within time series data [52]. These outliers or anomalies can be categorized into three types: type I, type II, and type III, based on their distinctive behavior within the data [52]. Furthermore, another categorization considers whether the outliers appear as single-point anomalies, sequential outliers consisting of a continuous series of samples, or periodic outliers [53]. In many studies, type II and type III anomalies are commonly referred to as anomalies. Given the variation in outlier types, it becomes essential to employ appropriate outlier detection methods tailored to detect each specific type of outlier. Thus, the initial step in the outlier and anomaly detection process should involve identifying the outlier type present in the data. This enables the selection of suitable approaches and optimal parameter settings.

1.2.1.1 Type I Anomalies (Point Outliers, High Magnitude Anomalies)

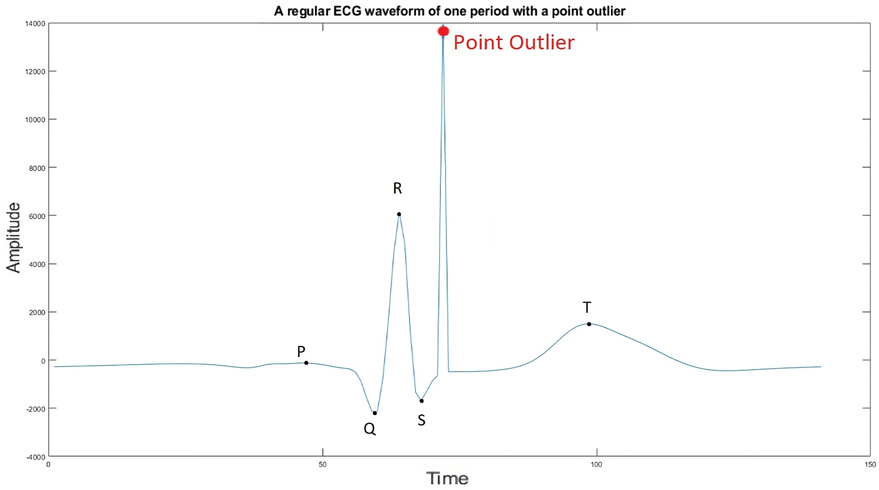
Type I anomalies encompass various statistical properties or features that distinguish them from the rest of the data, regardless of the data modality. Point, sequential, and periodic outliers all may fall under the category of type I anomalies as long as their amplitudes are comparably greater than the baseline data range [52]. Type I anomalies are commonly defined across different research fields. For example, in engineering studies, outliers are defined as samples that exceed a predefined set of thresholds, while statistical studies consider samples as outliers when their standardized values reach a predetermined quantile [54, 55].

In one-dimensional biomedical data, type I anomalies often arise from noise and motion artifacts, abnormal readings associated with subjects such as disorders or markers, instrument defects, and environmental changes during measurements [56, 57, 6]. However, in two and three-dimensional biomedical data, additional outlier-causing phenomena may include abnormal readings related to subjects such as lesions, tumorous tissues, or sudden changes in heat [58, 59]. Motion artifacts and device-related artifacts are particularly common in ECG data, leading to type I anomalies. Motion artifacts tend to persist for a longer duration and can cause trending or level shifts in the data, while noise artifacts, with their shorter appearances, can mimic individual structures and hinder automated algorithms from detection [60, 61].

Figure 1.1a provides an example of a type I outlier sequence in ECG data caused by a motion artifact, while Figure 1.1b illustrates a point outlier of type I within a regular PQRST structure of ECG data.



(a) ECG data with a motion artifact in the form of a type I outlier sequence.



(b) A regular PQRST structure of ECG data with an added point outlier of type I.

Figure 1.1: Example representations of type I anomalies.

Type I outlier samples typically exhibit statistically different behavior, often manifested as deviations in the mean value of the data. Compared to type II and type III anomalies, type I anomalies are relatively easier to detect using statistical outlier detection algorithms [40]. After basic preprocessing steps specific to the data modality, such as noise elimination and detrending operations, parametric outlier detection algorithms can be employed.

Figure 1.2 provides another example of a type I outlier. It showcases a randomly generated data set with a normal baseline distribution (N(0,1)), with an outlier synthetically placed at the 80th sample.

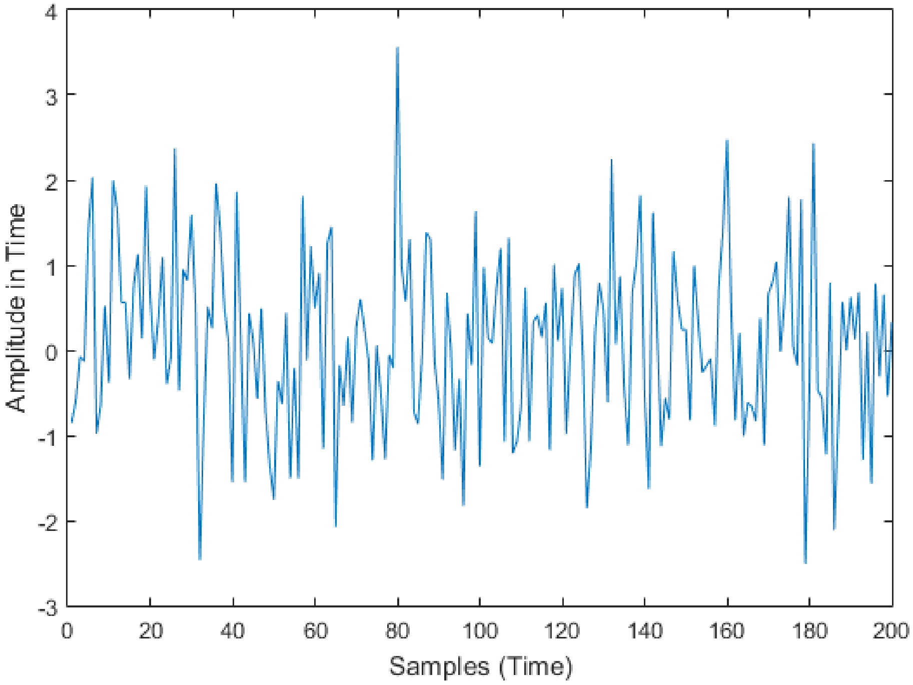


Figure 1.2: Example data generated using Gaussian Normal distribution, and a low-amplitude outlier placed at the 80th sample.

1.2.1.2 Type II Anomalies (Contextual Anomalies, Spatial Outliers)

Type II anomalies, also known as contextual outliers, are the most commonly observed outliers in datasets, exhibiting irregular behavior within a regular period of data behavior [52]. While the data in these datasets display periodic patterns with local peaks or sinks, the outlying samples deviate from this periodic behavior, making their detection dependent on analyzing the temporal neighbors of suspected data samples rather than considering the entire dataset [62]. Although most type II anomalies can be considered as inliers according to the statistical definition of outliers, meaning that they do not significantly deviate from the overall data mean [55], standard statistical outlier detection approaches may fail to identify them unless specific preliminary steps are taken.

Contextual outliers in time series data can sometimes resemble type I anomalies, either as individual samples or sequential samples. However, the key distinction lies in the fact that type II anomalies have amplitude values within the normal range of the data rather than representing extreme data points. Consequently, type II anomalies do not dramatically alter the statistical properties of the data, but they can hinder the effectiveness of processing algorithms applied to the data. Furthermore, they often indicate issues in data recording, such as minor motion artifacts, disorder-related problems, or unexpected variations in regular periodic data behavior. Figure 1.3 illustrates an example of a motion artifact-related type II outlier, where the sequence of type II outliers deviates irregularly from the rest of the PQRST structure of the ECG data, highlighted in red.

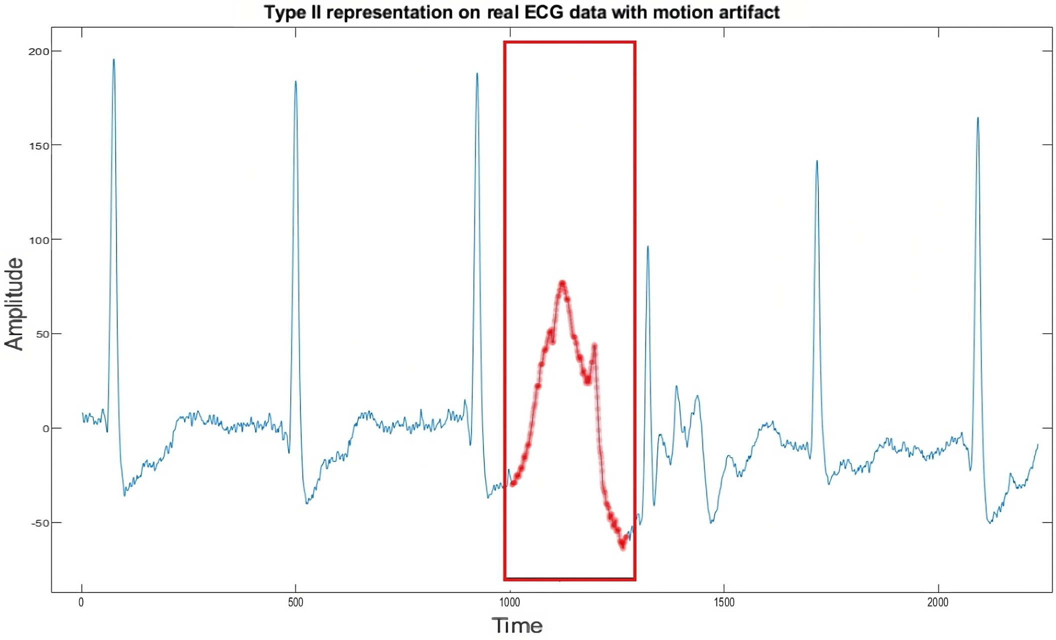


Figure 1.3: ECG data with a motion artifact in the form of a type II outlier sequence.

Detecting type II anomalies necessitates employing more sophisticated methods compared to standard parametric techniques. Initially, it is crucial to determine either the period or the expected behavior of the data, which can be achieved through frequency domain approaches, segmentation, or moving/sliding window analyses. After identifying the regular period of the data, additional analysis is carried out within each period. One potential approach involves detrending each period of the data segment and applying periodic outlier detection methods to pinpoint the outlying samples [63]. This involves removing any underlying trends or patterns from each period to focus solely on the irregular occurrences that characterize type II anomalies. By applying specialized techniques designed for detecting outliers within periodic data, it becomes possible to effectively identify the type II anomalies that deviate from the expected behavior within each period. Such adaptive approaches enable the detection of contextual outliers that may not exhibit extreme values but disrupt normal temporal patterns.

In the literature, numerous methods can be found that can be applied to detecting type II outliers, even though they may not have been specifically developed for this purpose. For example, modeling-based moving window approaches have proven to yield robust results in detecting point-type II outliers in periodic data. Auto-regression-based algorithms like ARIMA [64], median-based non-parametric outlier detection algorithms such as median-difference window subseries score (MDWS) [65], and algorithms based on short/long-term pattern recognition [66] are some examples of such methods. Another avenue is the use of learning-based approaches, such as employing a deep learning approach using an encoder-decoder system [67] for detecting type II outliers in ECG data. Modeling and frequency domain-based techniques are also utilized for identifying contextual outliers. One approach involves utilizing wavelet transform-based regression [68]. Furthermore, combining the frequency domain approach with the autocorrelation structure, known as Fourier auto-correlation structure (FLAC), has demonstrated high efficiency in detecting contextual outliers in multi-channel ECG data [69].

In the context of ECG datasets with type II outliers, clustering-based approaches are commonly employed. These approaches typically involve segmenting heartbeats and grouping together beats that exhibit similar shapes. A popular clustering algorithm for outlier detection is normalized cross-correlation clustering (NCCC), which adopts an iterative approach using cross-correlation values. The algorithm continues iterating until a type II outlier is detected in a heartbeat, where the cross-correlation of that particular heartbeat significantly differs from the rest of the group [70]. Another widely used clustering-based outlier detection algorithm is the density-based spatial clustering of applications with noise (DBSCAN), which relies on the sparsity of anomalous segments within ECG data [71].

1.2.1.3 Type III Anomalies (Collective Outliers) (Anomalies)

In addition to the statistical properties, the periodicity of outliers plays a crucial role in detecting anomalies, particularly in seasonal or periodic datasets where outlier behavior appears periodically and differs statistically from the overall data [33]. In quasi-periodic data, certain intervals may exhibit non-periodic behavior or periodicity with varying frequencies within a periodic dataset. Unlike the detection and removal of outliers in biomedical datasets, this periodic behavior often represents the distinguishing characteristics of the data modality. Much of the literature on ECG data anomalies focuses on the detection of type III anomalies, which are anomalies falling under the category of cardiac arrhythmias [72].

Arrhythmias encompass a wide range of sub-categories, and their causes can include signaling malfunction, hormonal instability, nutrition dependence, cardiovascular system degeneration, or disease-related malfunctions [73]. While each cause can be further explored, their primary effect on ECG data is the alteration of its regular periodic behavior. Moreover, such distortion of the regular periodic behavior can sometimes result from the temporal combination of independent components within the ECG data, leading to narrowed or prolonged sub-intervals in the PQRST structure and the emergence of a new outlying structure [74]. A clinical example of an outlying periodicity within the ECG data can be observed in Figure 1.4, where a Ventricular Tachyarrhythmia interval during an ECG recording generates a distinct structure that deviates from the rest of the periodic behavior due to an increased heart rate and severe desynchronization of the PQRST structure [75].

Detecting type III anomalies, which correspond to different types of arrhythmia intervals in ECG data, is generally more straightforward compared to the detection of type II anomalies. Several commonly used methods are available for detecting type III anomalies in ECG data, each with its own advantages for identifying specific types of arrhythmias. Frequency domain approaches, such as the wavelet transform, Fourier transform variants, and spectral analyses like moving window and autoregressive modeling, are frequently employed for detecting type III anomalies. These

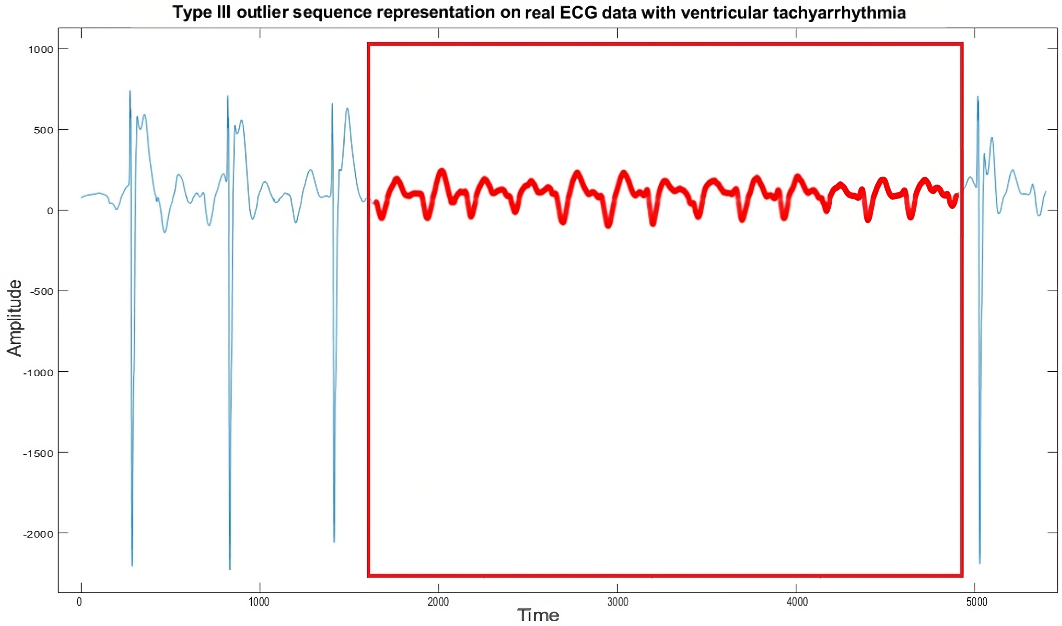


Figure 1.4: ECG data with ventricular tachyarrhythmia as a type III outlier sequence (red).

methods enable the analysis of frequency components and variations in the ECG data, aiding in the identification of specific arrhythmia patterns [40, 55, 76]. Learning and modeling approaches, including deep learning, neural networks, and support vector machines, offer the capability to learn complex patterns and relationships within the ECG data. These techniques can effectively detect various arrhythmia types by leveraging their ability to capture intricate dependencies and features in the data. Clustering and density-related approaches, such as k-means and k-medoids algorithms, can be utilized to group similar ECG patterns together. This clustering-based analysis helps identify abnormal arrhythmia intervals that deviate from the typical patterns observed in the data. Distribution-related approaches, like the Kolmogorov-Smirnov test, examine the statistical distribution of the ECG data to identify deviations from expected distributions. Such methods can effectively identify anomalies that exhibit significant differences in their distribution properties compared to normal ECG patterns. Mathematical approaches, such as derivative analysis, focus on analyzing the rate of change in the ECG signal. By examining the derivatives or gradients of the data, these methods can detect abrupt variations or abnormal trends associated with specific arrhythmias. Each of these approaches offers specific advantages in detecting different types of arrhythmias within ECG datasets.

1.2.2 Anomalies in Biomedical Data

Biomedical time series data encompasses various modalities such as electrocardiography (ECG), electromyography (EMG), skin conductance rate (SCR), and electroencephalography (EEG) [77]. These modalities capture different aspects of an organism's biological conditions, and data from biological sources are typically collected using specialized equipment with unique specifications [22]. As a result, biomedical signals exhibit diverse forms and quasi-periodic behavior, with signal characteristics varying across modalities [23].

The devices used for measuring quasi-periodic time series data in biomedical applications can occasionally experience malfunctions, produce noise, or miss intervals, making it essential to maintain proper electrode placement and connection to minimize the noise in the recorded data [78]. Anomalous conditions can disrupt the periodic behavior of biomedical signals, resulting in noisy intervals that can obscure or alter the underlying physiological signals [79]. Detecting these anomalous data intervals is crucial for various purposes, such as fine-tuning data models, diagnosing disorders, recognizing patterns, and improving the accuracy of future sample predictions [24, 80].

Anomalies in biomedical data can stem from diverse sources, presenting challenges to accurate analysis and interpretation such as measurement errors, including instrument calibration issues, data acquisition malfunctions, artifacts in medical images and recordings, such as motion artifacts or noise or human errors during data collection [81, 82]. Moreover, environmental factors like ambient noise, electromagnetic interference, or variations in experimental conditions can introduce unexpected variations and distort the data [83]. Systematic errors caused by faulty sensors or malfunctioning medical devices can even introduce consistent, or collective anomalies [84]. Outliers, on the other hand, extreme values deviating significantly from the expected range in biomedical time series data, originate mostly from the device or recording malfunctions and can distort statistical analysis and compromise result reliability [85].

Inconsistencies and anomalies in biomedical data may also arise when integrating multimodal data from different sources that have varying quality and protocols [86]. Anomalies can also be generated through biased or incomplete data, resulting from sample selection, collection processes, or inadequate data preprocessing techniques, such as handling missing values or imputing erroneous data [6]. Anomalies can also arise from complex variable interactions or hidden patterns in the data. Additionally, anomalies can be introduced unintentionally through insufficient quality control measures during data acquisition, as well as errors during data transmission, storage, entry, or labeling [87]. Statistical anomalies and biased results can arise from small sample sizes or imbalanced datasets too [40].

Detailed analysis of data characteristics enables the prediction of past and future behavior of biological systems, but this predictability is typically limited to stable internal and external environments. Unexpected conditions like disorders or environmental changes, irrespective of the organism, disrupt stability and cause irregular changes in signal behavior, resulting in anomalies [6, 36]. Unfortunately, anomalies with a higher intensity often result in more significant changes in signal characteristics, leading to inconsistent analysis results and failure in disorder diagnosis, particularly when the anomaly stems from an external cause [88, 89].

Understanding the diverse sources of anomalies in biomedical data is crucial for developing robust anomaly detection methods and ensuring the reliability and integrity of research findings and clinical decision-making [87]. Successful classification of biological datasets relies on detecting and extracting unique features. Furthermore, understanding the behavior of quasi-periodic anomalies enables the forecasting of future occurrences of similar anomalies [90]. The emergence of new technologies and novel data sources brings about new possibilities for the detection of a variety of anomalies in biomedical data, necessitating continuous monitoring and adaptive analysis techniques.

1.3 Anomaly Detection

Anomaly detection is a well-researched field, with a wide range of statistical outlier detection methods available in the literature [91, 62]. These methods can be categorized as parametric, non-parametric, or semi-parametric approaches [92]. Parametric methods involve data modeling and statistical moments and can be further divided into depth, distribution, and graph-based methods [90]. Parametric methods are particularly effective in detecting outliers with significant deviations from the mean, making them suitable for detecting outliers with higher amplitudes [93, 94, 34]. For example, the z-score method, among parametric approaches, performs well with small sample sizes and large outlier amplitudes [90]. On the other hand, non-parametric methods do not rely on data modeling and can be classified as distance, clustering, or density-based methods [95]. Non-parametric approaches tend to be more effective when dealing with larger sample sizes and a smaller number of outliers [96]. The box plot is a commonly used non-parametric method that works well with discrete data and limited samples, allowing for visualization of the median features of outlying samples in comparison to the overall median [97]. In addition to purely parametric or non-parametric methods, there are hybrid or more complex outlier detection techniques available, such as the sketching method [98], suffix tree-based noise resilient (STNR) approach [99, 100], iterative Grubbs approaches [101], autonomous anomaly detection [102], and the Tietjen-Moore test [103]. Frequency domain-based outlier detection methods also exist, utilizing algorithms that incorporate the frequency domain [104, 105, 106, 107, 108, 109]. For the detection of periodicity in time series data, the warping for periodicity (WARP) method has been developed [110]. However, it is important to note that these approaches either use the frequency domain for modeling purposes or as support for other outlier detection algorithms. It is crucial to recognize that there is no universally superior method for detecting all types of outliers across different data distributions. Each method has its own strengths and weaknesses depending on the characteristics of the outliers and the data itself [111, 34, 40]. Detailed information about the commonly used methods and the ones compared in the proposed algorithm can be found in Chapter 2 in the technical background section.

In preprocessing time series data, particularly in biomedical data, anomaly detection is crucial for the accurate prediction of future samples and for preparing the data for main analyses. Identifying distinctions between control and test groups can be critical, as data from subjects with disorders often exhibit specific outlier patterns [112, 18]. For instance, in regular ECG data, the periodic pattern created by the PQRST structures is characterized by common units. However, heart disorders or malfunctions in the heart's signaling pathway can disrupt the ECG data pattern by suppressing/enhancing or altering the timing of the PQRST components, ultimately affecting the regular beat frequency [113]. Therefore, the classification of healthy individuals versus those with a heart disorder may rely on the identification of anomalous patterns.

1.3.1 Real-time Anomaly Detection

In real-time operating systems, data is typically received from sensors at fixed sampling rates and transmitted in batches for efficient processing [114]. The amount of data forwarded to the anomaly detection algorithm within a specific time frame has a significant impact on its functionality and performance. Smaller data packets can improve computational speed, reduce lag, and enable swift processing [115]. However, it is important to consider that reducing the data quantity may compromise the accuracy of the anomaly detection algorithms [116]. Finding the right balance between computational efficiency and accurate anomaly detection is crucial. While fewer data points in a batch can enhance computational speed, the selected data quantity must still be sufficient for precise anomaly detection [117]. Excessively reducing the data amount can impair the algorithm's ability to accurately identify anomalies [118]. Therefore, determining the optimal data quantity is essential for efficient processing and accurate anomaly identification in real-time systems as there are two primary reasons for the potential decline in accuracy [119]. First, the algorithm may fail to capture meaningful features in the received data batch. Second, if the batch is shorter than a prolonged anomalous pattern, the algorithm may not detect the anomaly unless it compares feature values with those from the previous data batch [40].

Real-time anomaly detection is widely used across various industries and domains to enhance security, improve operational efficiency, prevent financial fraud, ensure patient safety, optimize logistics, and maintain critical infrastructure integrity [74]. In the cyber security field, it plays a crucial role in promptly identifying and mitigating security breaches by monitoring network traffic, user behavior, and system logs in real-time, enabling organizations to quickly detect and respond to malicious activities [120]. The financial sector utilizes real-time anomaly detection to detect fraudulent transactions, identify abnormal trading patterns, and prevent money laundering activities [121]. In the manufacturing industry, it is applied by monitoring sensor data from equipment and machinery to identify and predict equipment failures or production deviations, minimizing downtime and optimizing operational efficiency [122]. Transportation and logistics companies leverage real-time anomaly detection to monitor fleet operations, track vehicle routes, control autonomous vehicle patterns, and identify deviations or irregularities in delivery schedules, ensuring efficient logistics management [123]. Furthermore, real-time anomaly detection finds applications in the energy and utility sectors for monitoring power grids and detecting abnormalities that may indicate power outages or potential equipment failures [124].

Real-time anomaly detection is highly applicable in biomedical engineering, providing valuable insights and advancements in healthcare. It has significant use cases, including patient monitoring, where real-time anomaly detection algorithms can analyze physiological signals like electrocardiogram (ECG), blood pressure, and oxygen saturation levels to identify abnormal patterns that may indicate potential health issues [125, 126]. For instance, these algorithms can detect heart attacks in patients and identify real-time driver drowsiness, enabling healthcare professionals to intervene promptly and provide appropriate medical attention [127]. Another use case is the analysis of medical imaging data, such as X-rays, CT scans, and MRIs, where real-time anomaly detection algorithms can identify abnormalities or lesions in these images, facilitating early disease diagnosis, such as cancer, and enabling timely treatment [128, 74].

1.3.2 Subject-specific Anomaly Detection

Subject-specific anomaly detection refers to the detection of anomalies or abnormal patterns specific to a particular subject or individual within a given context [129]. It involves the analysis of subject-specific data and the identification of deviations from expected behavior or patterns. Parameters in subject-specific anomaly detection typically include the characteristics and attribute specific to the subject under study. By tailoring the anomaly detection algorithms to the subject's unique characteristics, subject-specific anomaly detection can provide more accurate and personalized results, improving the detection and interpretation of anomalies within a specific subject or population [21].

Subject-specific anomaly detection plays a critical role in the field of biomedical engineering, particularly in healthcare, where personalized approaches are paramount [129]. By incorporating individual factors such as genetic information, medical history, and physiological signals, subject-specific anomaly detection contributes to early disease detection, personalized treatment planning, and improved patient outcomes [130]. It empowers healthcare professionals to detect subtle deviations from expected patterns, aiding in disease diagnosis and treatment monitoring. Leveraging subject-specific anomaly detection techniques, biomedical engineers can enhance the accuracy, efficiency, and personalization of healthcare interventions, thereby advancing the field and benefiting patients [40].

Detecting anomalies in individual biomedical data is challenging due to the unique signal patterns exhibited by each person within a population [131]. This challenge is particularly amplified in human physiology, making anomaly detection even more difficult. As a result, subject-specific anomaly detection becomes necessary for individuals whose regular signal patterns deviate significantly. The literature presents various algorithms for subject-specific anomaly detection in generalized time series data [132, 111]. While subject-specific anomaly detection algorithms generally offer improved accuracy, they often come with higher computational complexity compared to generalized algorithms [87, 133, 134]. Many subject-specific anomaly detection algorithms operate offline due to their longer computational times, which limits their real-time applicability [87]. Despite the computational costs associated with subject-specific anomaly detection, it is often necessary to achieve higher true positive rates [135].

1.3.3 Features for Anomaly Detection

Features play a vital role in time series anomaly detection, as they are extracted to capture various characteristics of the data that may indicate abnormal behavior. Several types of features can be used for this purpose, including statistical features, frequency domain features, dissimilarity and similarity measures, transformational features, and other types of features [136]. Each type offers unique insights into the time series data, making them valuable for detecting anomalies, particularly in biomedical applications. With recent developments in machine learning algorithms, the need for unique feature extraction has become a trending topic in achieving accurate data classification [137]. As the number of unique features increases, since their overall representation of the data improves, the classification success rate generally improves while keeping sufficient sample sizes to prevent overfitting, and avoiding the curse of dimensionality [138, 139].

Statistical features provide a comprehensive view of the distributional properties of the time series. The most commonly used complicated statistical features are generally derived from the basic statistical moments or their variations such as mean, median, standard deviation, variance, skewness, and kurtosis [140]. Here, mean and median provide information about the central tendency of the data, standard deviation, and variance capture their dispersion, skewness indicates the asymmetry of the distribution, and kurtosis measures the tails' thickness [141]. By comparing these statistical features with expected values or normal ranges, anomalies can be detected based on significant deviations from the expected patterns.

Frequency domain features analyze the signal in the frequency domain and are derived from algorithms like the Fourier Transform or other spectrum estimation methods including power spectral density, band frequency, fundamental frequency, and other frequency components [142]. Detecting anomalies involves identifying changes in spectral composition or the presence of abnormal frequencies, where the spectral features are useful for capturing quasi-periodic patterns or oscillatory behavior in time series data [143]. To detect periodic components, frequency domain algorithms like the Fourier Transform are used to reveal the relationship between the frequency domain and quasi-periodic time domain elements [144]. This relationship allows for effective classification by extracting unique features from the data [145]. In cases with oscillatory quasi-periodic behavior with high sampling rates, frequency domain approaches generally outperform time domain approaches [44].

Dissimilarity metrics are often used to measure the distance between two or more time series, where these metrics calculate the distances between individual samples and provide an overall score for the entire dataset [146]. Commonly used dissimilarity metrics include Euclidean, Manhattan, and Chebychev distances, which require the time series data to have equal lengths and are suitable for numerical comparisons [147]. On the other hand, similarity metrics such as cross-correlation assess positional similarity by examining the displacement of one data part into another [148], and cross-covariance explores the relationship between different data parts and is sometimes employed in multivariate classification problems [149]. These dissimilarity and relationship metrics offer distinct representations of the differences between datasets and can be used as features for anomaly detection in time series data [150]. Dissimilarity metrics, such as Euclidean, Manhattan, and Chebychev distances, compare the numerical differences between time series data by matching samples one-to-one, whereas, temporal similarity metrics like cross-correlation evaluate the similarity between different parts of the data by analyzing their displacement or positional similarity [151]. Higher dissimilarity or lower similarity values can indicate the presence of anomalies.

When analyzing biomedical time series data, the selection of suitable features relies on the distinctive characteristics of the data and the specific anomalies under consideration. It is customary to employ a combination of different types of features to capture diverse aspects of the data [152]. Various feature types may perform exceptionally well in different scenarios or for particular types of anomalies. Hence, meticulous feature selection and evaluation are crucial for achieving accurate anomaly detection in biomedical time series data [153]. Employing domain-specific techniques in each field of study, including biomedical analysis, can provide unique advantages in addressing specific tasks, even though there may be some overlap with approaches from other fields [154]. This is because each field has its own set of algorithms and methods tailored to its unique challenges and requirements.

1.3.4 Detection of Quasi-periodic Outliers

Quasi-periodic outliers in time series data exhibit a quasi-periodic pattern, which can be observed as peaks in the frequency domain. For instance, by applying the discrete-time Fourier transformation (DFT), the time domain signal with periodic outliers, as shown in Figure 1.5, can be transformed into the frequency domain, resulting in Figure 1.6.

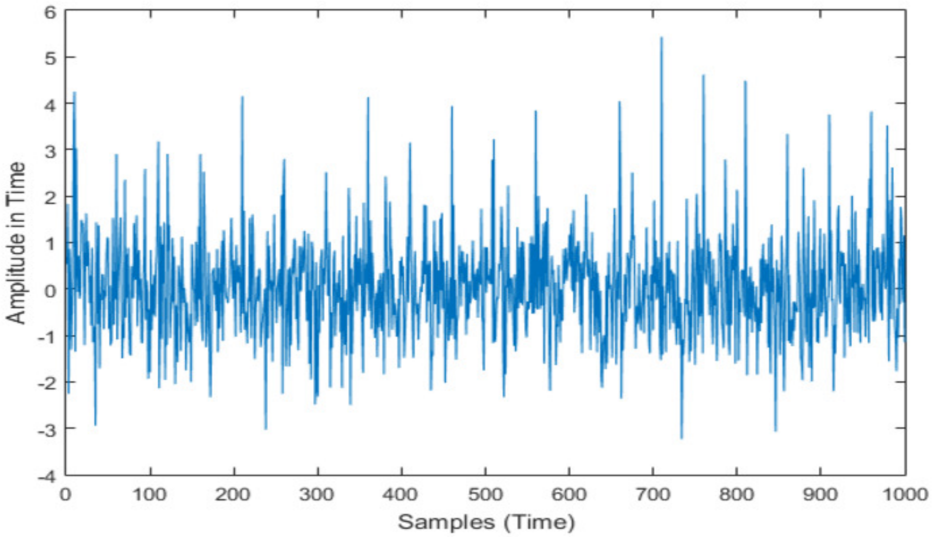


Figure 1.5: Randomly generated time-domain data with periodic outliers placed per 50 samples with random amplitudes.

As observed in Figure 1.6, the peaks in the frequency domain are more distinguishable compared to the fluctuations in the time domain shown in Figure 1.5. The first peak after the first sample in the frequency domain corresponds to the main oscillation frequency in the time domain data, representing the periodicity of the outliers. It is important to note that automated computation in a computer environment can be challenging in identifying this first peak, particularly when low-frequency periodicity is inherent in the data. Therefore, developing an algorithm that can detect all peaks and perform computations based on the detected peaks would be a better alternative

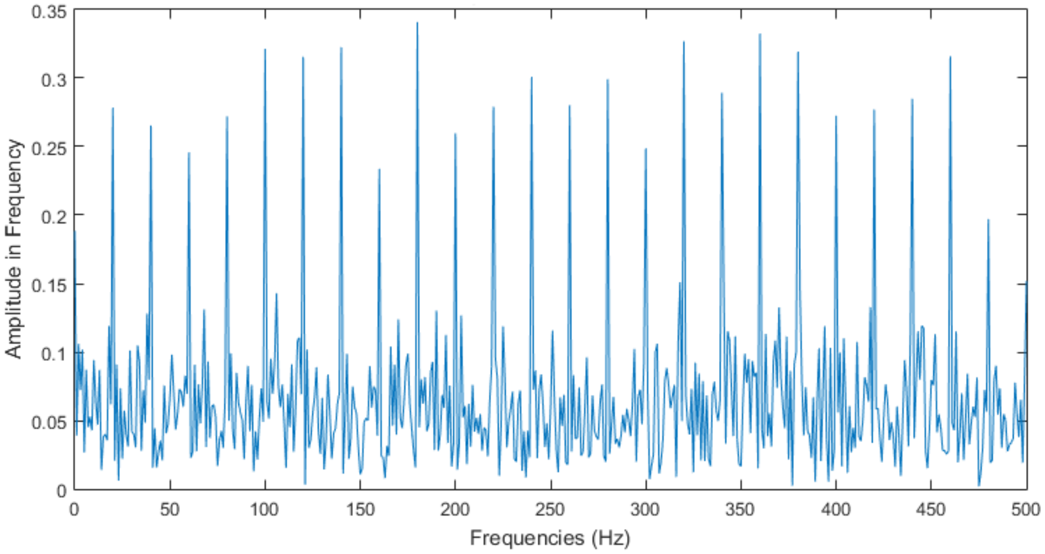


Figure 1.6: The frequency domain estimation of the same data in Figure 1.5, using DFT transformation.

for identifying the main periodicity of the quasi-periodic outliers. By analyzing all the peaks and their harmonics, which provide information about the periodicity of the data, a more comprehensive understanding of the time series can be achieved [155].

1.4 Motivation, and Possible Contributions to the Literature

Collective and contextual anomalies in quasi-periodic time series data can display intricate patterns, as demonstrated in Section 1.2.1.3 and Section 1.2.1.2, respectively. However, since the baseline data also exhibit quasi-periodic behavior and often encompass a broader range than the anomalies themselves, detecting these anomalies becomes challenging when relying solely on features from a single domain. Therefore, it is hypothesized that leveraging both time domain and frequency domain properties would be beneficial in detecting collective and contextual anomalies in quasi-periodic time series data.

Following this motivation, this thesis introduces a new anomaly detection approach, designed to detect collective anomalies in quasi-periodic time series data. Inspired by the FOD algorithm [44], and the WFOD algorithm [140], the proposed approach employs pitch frequency estimation in the spectral domain, a technique commonly utilized in audio signal processing [156, 157]. The pitch frequency, also known as the fundamental frequency, captures the primary oscillatory frequency and the highest energy component of the data [158]. The proposed approach employs a moving window strategy for real-time analysis. Within each window, it estimates the spectral domain and identifies the pitch frequency, which is then transformed into the time domain. By leveraging dissimilarity measures, the proposed approach effectively discerns normal behavior from potential anomalies. This hybrid algorithm capitalizes on the strengths of both time and frequency domains, enabling the detection of anomalies over short and long-term periods.

Consequently, the proposed approach provides a comprehensive perspective on anomalous patterns, making it particularly valuable in real-time scenarios. Moreover, this study introduces a distinctive feature extraction approach incorporating a comparative analysis of engineering and statistical methodologies. The objective of the proposed approach is to extract unique features from time series data in the moving windows approach and establish an anomaly detection pipeline that supports real-time analysis. Furthermore, it utilizes a reinforcement learning approach to update the decision-making behavior as it progresses through the data, which makes it adaptable to any subject by learning the subject-specific patterns. Therefore, the proposed approach algorithm can be considered as the subject-specific anomaly detection approach too.

In summary, this thesis study makes valuable contributions to the literature on signal processing, time series analysis, biomedical data processing, and machine learning. Hereby, the following novelties and remarks are presented in this thesis:

- The proposed algorithm is a novel method that employs both time and frequency domain properties to identify collective anomalous intervals in quasi-periodic time series data.
- An evaluation is conducted to compare the performance of the proposed approach with other anomaly detection methods on electrocardiography (ECG) data.
- The proposed algorithm is highly customizable and adaptable, as it seamlessly

integrates new features and adjusts to evolving data characteristics in a multivariate anomaly detection pipeline.

- The proposed approach holds the potential for real-time monitoring and detection of anomalies in biomedical or similar quasi-periodic time series data measurement devices or environments.
- The proposed approach adjusts the anomaly decision criteria by dynamically learning from the data, making it a subject-specific anomaly detection approach.

The remaining sections of the paper are organized as follows. Firstly, Chapter 2 provides a concise overview of the technical background related to the algorithms employed in the proposed method or used for testing purposes in the experimental setups. Subsequently, Chapter 3 presents the previous research and studies conducted to develop the final version of the proposed method. Chapter 4 offers a comprehensive explanation of the proposed approach, including a detailed description of the computational steps, their utilization, and the calculation of specific parameter values. The applications of the proposed algorithm, experimental setups, comparisons with other benchmark methods, experimental results, and their evaluation can be found in Chapter 5. Finally, Chapter 6 serves as the conclusion of the thesis, summarizing its content, highlighting the contributions of the proposed algorithm to the existing literature, and discussing potential future works and research directions.

CHAPTER 2

TECHNICAL BACKGROUND

This chapter presents the technical background for the used or referenced algorithms. Hereby, the chapter is divided into several sections, including the general definition and properties of the time series, which are linked and continued by the frequency domain and some algorithms and definitions related to it. Then, a section contains the most commonly used outlier and anomaly detection methods in the literature. Moreover, the dissimilarity measures are presented as they are used in the proposed algorithm as feature extraction techniques. Following that, the machine learning section takes place, which includes the most commonly used machine learning classifiers as well as the performance evaluation steps and properties for a classification task. Therefore, instead of beginning with an interconnected methods-like chapter, this chapter presents quick reference information for the reader to maximize their comprehension of the proposed algorithm.

2.1 Some Properties of Time Series

Since time series are thoroughly mentioned in Chapter 1, this section is prepared to present some of the main properties of time series related to their spectral domain estimations, such as sampling rate, stationarity, and periodicity as they are heavily referenced in this thesis research.

2.1.1 Stationarity

Stationarity refers to the statistical properties and joint probability distribution of a time series remaining constant over time [159]. A stationary time series exhibits consistent characteristics on the probabilistic data distribution and statistical moments, such as constant mean, constant variance, and stable autocovariance structure through time [160]. Many statistical signal processing methods work properly on stationary time series data or its weaker subset, namely, wide sense stationary (WSS) which provides weaker stationarity by assuming a constant mean and stable autocovariance [161]. Let's consider a discrete-time time series x_t, $C(\tau)$ denoting the autocovariance function with t denoting the time index, and τ as the lag in time. Then, a time series can be considered WSS if it satisfies Equation 2.1, and Equation 2.2.

$$\mu = E[x_t] = \text{constant} \tag{2.1}$$

$$C(\tau) = E[(x(t) - \mu)(x(t + \tau) - \mu)] = \text{constant} \tag{2.2}$$

Understanding the stationarity characteristics of ECG signals helps in the accurate diagnosis and monitoring of cardiac conditions [162]. In the context of ECG signals, stationarity implies that the average heart rate, heart rate variability, and amplitude characteristics of the ECG signal remain consistent throughout the recording [163]. This also indicates that statistical measures such as mean heart rate, heart rate variability, or power spectral density can be calculated accurately, enabling the detection of abnormalities and arrhythmias [164]. On the other hand, non-stationary ECG signals may exhibit time-varying characteristics, such as changing mean heart rate, varying heart rate variability, or amplitude fluctuations, and may indicate changes in cardiac activity due to physiological or pathological conditions and disorders [165]. Such abnormal conditions are often referred to as anomalies in the data.

2.1.2 Periodicity

Periodicity is a fundamental characteristic of time series data, indicating the presence of recurring patterns or cycles at regular intervals [166]. For a time series $x(t)$ to be considered periodic, it must exhibit a repeated pattern or cycle over a period T, as expressed in Equation 2.3.

$$x(t) = x(t + kT), \tag{2.3}$$

Here, k is an integer denoting the number of complete cycles.

In the context of biomedical time series data, achieving perfect periodicity over long-term measurements is highly challenging due to the inherent instability and slight changes in the periodic behavior of biological systems over time [167]. Consequently, the term "quasi-periodicity" is more suitable in the field of biomedical engineering, as it acknowledges the deviation from strict periodicity.

2.1.2.1 Quasi-periodicity

Quasi-periodicity pertains to a pattern or behavior observed in a time series that demonstrates approximate periodicity while also exhibiting variations or irregularities [168]. These deviations can arise from external factors, underlying physiological processes, or intrinsic variations within the system being observed [169]. In contrast to purely periodic signals that precisely repeat at regular intervals, quasi-periodic signals display slight variations in phase, denoted as ϕ_k, for each periodic component k, or exhibit modulations in their periodic behavior, as exemplified in Equation 2.4.

$$x(t) = \sum_{k=1}^{K} A_k \cdot \cos(2\pi f_k t + \phi_k), \tag{2.4}$$

In this equation, A_k represents the amplitude, while f_k denotes the instantaneous frequency.

ECG signals exhibit a quasi-periodic nature due to the repetitive patterns present in the cardiac cycle, such as the distinct PQRST structures [170]. The identification and analysis of different components and intervals in ECG signal heavily rely on their periodic characteristics. The duration between successive R peaks, referred to as the R-R interval, represents the periodicity of the cardiac cycle and they may not always exhibit perfectly periodic behavior but instead demonstrates a quasi-periodic pattern [171]. Changes in the variation of this periodicity can indicate irregular heart rhythms associated with certain disorders, and abnormal heart activities, including types of tachycardia and bradycardia [172].

2.1.3 Sampling Rate

In time series analysis, the sampling rate determines how frequently data points are collected over time, directly influencing the temporal resolution of the data [173]. Although the choice of sampling rate is often dictated by the data collection hardware, it holds significant importance as it impacts the types of analyses and the amount of information contained in each data batch, particularly in biomedical data modalities [174]. Data collection devices convert continuous systems into discrete time series data with a specific sampling rate, enabling signal processing in a digital environment [175]. The sampling rate is measured in Hertz (Hz), where 1 Hz is equal to s^{-1} (seconds). The sampling rate of a time domain signal is inversely related to the corresponding time interval, denoted as T_s, and the relationship between the sampling interval and the sampling rate, or sampling frequency, f_s, can be expressed by Equation 2.5.

$$T_s = \frac{1}{f_s} \tag{2.5}$$

Considering the frequency and periodicity information provided by Equation 2.5, the relationship between the continuous-time signal, $x(t)$, and its discrete counterpart, $x[n]$, can be mathematically represented by the sampling theorem, as shown in Equation 2.6 [176].

$$x[n] = x(nT_s) = x(t)\Big|_{t=nT_s} \tag{2.6}$$

Here, n represents the sample index, while t represents the time index.

A lower sampling rate leads to a coarser representation of the continuous-time signal, which can result in information loss, particularly when dealing with rapidly changing or irregular events [177]. Moreover, fine details and high-frequency components may be missed with low sampling rate values, which can reduce the accuracy of subsequent analyses [178]. Conversely, a higher sampling rate involves collecting more data points, leading to larger datasets with generally larger data sizes, posing challenges in terms of data storage, computational requirements, and processing time [179].

In the case of ECG signals, a higher sampling rate allows for a more detailed representation of the cardiac electrical activity, enabling accurate detection of specific features such as the PQRST structure complex. This level of detail is crucial for identifying arrhythmias and other cardiac abnormalities [180]. If the ECG signal is sampled at a rate lower than the periodicity of the underlying signal behavior, the resulting time series will provide a coarser representation of the cardiac electrical activity, which may lead to the loss of fine details and high-frequency components, compromising the accuracy of subsequent analyses and diagnostic interpretations [177]. Therefore, selecting an appropriate sampling rate for biomedical time series, such as ECG signals, is essential to ensure an accurate representation and meaningful analysis of the underlying physiological processes.

2.2 Frequency Domain and Spectral Analysis

The frequency domain refers to the representation of a signal's spectral content in terms of its frequency components, including information about amplitude and phase [181]. This representation is achieved through a mathematical transformation known as the Fourier transform (FT), which converts a signal from the time domain to the frequency domain [182]. The FT projects the time signal onto sinusoidal functions with discrete frequencies, allowing any time series data with periodic components and a sampling rate to be expressed as a sum of sinusoids with specific frequencies [183].

To apply the discrete Fourier transform (DFT), Equation 2.7 can be utilized.

$$X(k) = \frac{1}{N} \sum_{n=1}^{N-1} x(n) e^{-(\frac{j2\pi k}{N})}, \tag{2.7}$$

In this equation, $N - 1$ represents the maximum unique frequency component for the given sampling, corresponding to π radians of the periodic data (as defined by the Nyquist theorem). The variable n represents the sample index, while $x(n)$ denotes the amplitude in the time domain at sample n. The term k/N represents the number of cycles k per N samples, j represents the imaginary unit, and e^{-j} represents Euler's formula for the orthogonal sinusoidal components.

Frequency domain analysis provides valuable insights into the characteristics of a signal and finds applications in various fields such as signal processing in telecommunications, electromagnetic fields, control systems, and biomedical signal processing [184, 185]. In biomedical applications, frequency domain and spectral analysis techniques are commonly employed for diverse purposes, particularly in signal filtering, which is used to eliminate noise or undesired frequency components from biomedical signals [186]. Digital infinite impulse response (IIR) and finite impulse response (FIR) filters, as well as their well-known sub-types such as low-pass, high-pass, and band-pass filters, are commonly used in preprocessing steps for a wide range of biomedical signal analyses [187]. Additionally, the notch filter is widely used in biomedical signal processing to remove noise caused by electrical power lines, known as hum noise which often occurs around 50-60 Hz and manifests as noise in the time domain data [188]. Eliminating this noise from biomedical data may significantly improve the performance of models and analyses [189].

2.2.1 Quefrency

The quefrency domain, also known as the cepstral domain, provides information about the unique time-domain characteristics of a signal [190]. It offers insights into the periodicity and structure of the signal, making it particularly useful in peak detection for spectral analyses [191].

To obtain the quefrency representation of a signal, the following steps are typically followed:

1. Compute the Fourier transform $X(k)$ of the time domain signal $x(n)$.
2. Compute the logarithm of the magnitude of the Fourier transform, resulting in the logarithmic power spectrum.
3. Apply the inverse Fourier transform to the logarithmic power spectrum.

The resulting signal in the quefrency domain is referred to as the quefrency function or cepstrum. Mathematically, the quefrency representation C of a signal $x(n)$ can be derived using Equation 2.8, where $\mathcal{F}$ denotes the Fourier transform operator, and $\mathcal{F}^{-1}$ represents the inverse Fourier transform operator.

$$C = \mathcal{F}^{-1}\left\{\log\left(\left|\mathcal{F}\left\{x(n)\right\}\right|^2\right)\right\} \tag{2.8}$$

Detecting peaks in the time domain can be challenging, especially when dealing with noise or overlapping peaks [192]. Peaks in the quefrency domain, on the other hand, are often more distinct and separated, making them easier to detect while also providing valuable insights into the periodicity and structure of the signal, assisting in the identification of relevant peaks [166].

A common approach in peak detection using quefrency analysis involves identifying peaks in the quefrency function and then mapping them back to the time domain to obtain the corresponding peaks in the original signal [193]. This technique enhances the accuracy and robustness of peak detection algorithms, particularly in situations where traditional time-domain methods may struggle [194]. Additionally, quefrency analysis proves beneficial in analyzing the time-varying characteristics of the signal for non-stationary signals, where the frequency content of the signal varies over time [195].

2.2.2 Pitch (Fundamental) Frequency

The smallest harmonic frequency component of a periodic signal is known as the fundamental frequency, and it appears as the first peak among a sequence of periodic peaks in the frequency domain [160]. In the context of speech and audio processing, this frequency component, also known as the pitch frequency, represents the lowest harmonic frequency of sound vibrations [196]. As the pitch frequency increases, the sound becomes treble, higher-pitched, resulting in a shorter main signal period [197]. Detecting the pitch frequency is valuable as it provides insight into the properties of the sound [198]. Figure 2.1 provides an illustration of the pitch frequency, highlighting its appearance at 225.684Hz.

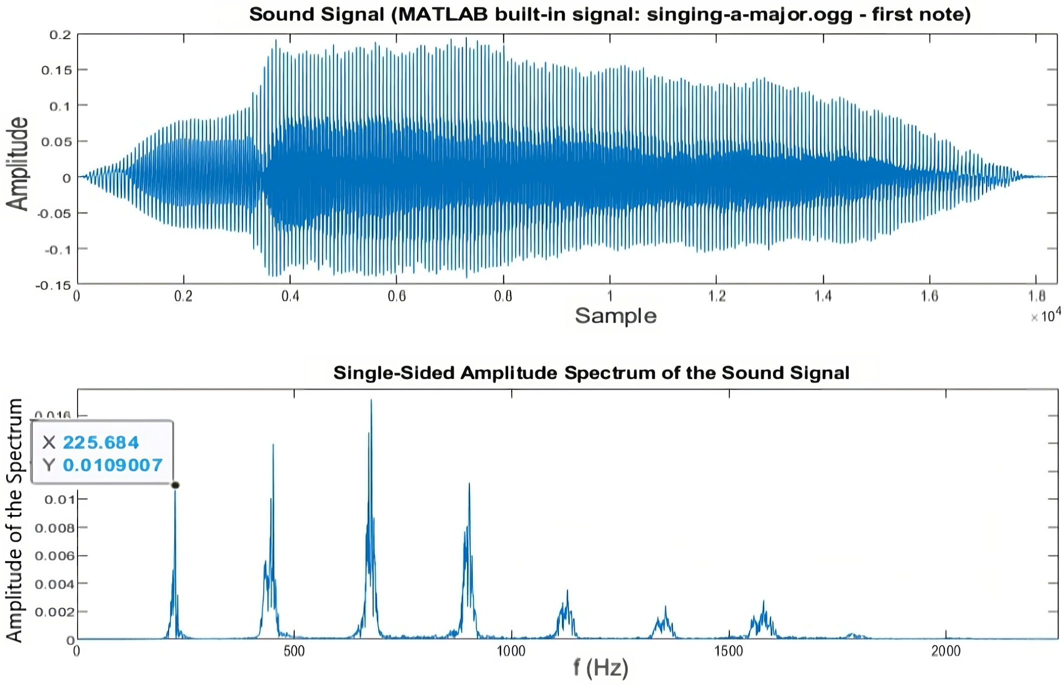


Figure 2.1: An illustration of the pitch frequency at 225.684Hz (bottom graph), obtained by the DFT of the upper sound data.

The fundamental frequency f_0 of a periodic time series signal $x(t)$ can be defined as the fundamental period's, t_0, reciprocal, which represents the duration between successive cycles of the periodic data [199]. Such an expression can be found in

Equation 2.9.

$$f_0 = \frac{1}{t_0} \tag{2.9}$$

Pitch frequency is commonly utilized as a prominent feature in speech recognition algorithms, as it can capture distinctive characteristics of the sound source [200]. Conversely, the use of pitch frequency in anomaly detection for time series, beyond audio signals, is infrequent, and the outcomes are typically task-specific [201]. The pitch frequency can also be applied to analyze periodic characteristics in other biomedical data modalities, such as ECG signals. In the case of ECG, the concept of pitch frequency, along with its reciprocal, the fundamental period t_0, can provide insights into the regularity of heart rate and the cardiac cycle [18]. Therefore, analyzing the pitch frequency of ECG signals can be particularly valuable for diagnosing arrhythmias [202].

2.2.3 Pitch Frequency Estimation Methods

This subsection exclusively focuses on the selected pitch frequency algorithms used in the proposed algorithm and its previous versions, namely the Normalized Correlation Function (NCF), Pitch Estimation Filter (PEF), Cepstrum Pitch Determination (CEP), Log-Harmonic Summation (LHS), and Summation of Residual Harmonics (SRH). These algorithms are chosen due to their direct availability within the MATLAB environment, specifically in the "Signal Processing" toolbox [203]. Hence, they are included in this subsection.

2.2.3.1 Normalized Correlation Function (NCF)

The Normalized Correlation Function (NCF) algorithm is utilized for pitch frequency estimation and is based on detecting peaks in the cross-correlation function in the frequency domain [204]. The algorithm follows these steps:

1. Compute the means and standard deviations of the reference pitch frequencies:

$$\begin{aligned} \mu_y &= \frac{1}{N}\sum_{i=1}^{N} y_i, \\ \sigma_y &= \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \mu_y)^2}, \end{aligned} \tag{2.10}$$

   where N represents the total number of reference pitch frequencies, and y_i denotes the ith reference pitch frequency.

2. Normalize the reference pitch frequencies:

$$\hat{y_i} = \frac{y_i - \mu_y}{\sigma_y}, \tag{2.11}$$

   where $\hat{y_i}$ represents the normalized reference pitch frequency.

3. For each frame of the input signal, calculate the normalized correlation coefficients:

$$\sigma_x = \sqrt{\frac{1}{M}\sum_{j=1}^{M}(x_j - \mu_x)^2}, \tag{2.12}$$

   where M is the length of each frame, x_j denotes the jth sample of the input signal, and μ_x is the normalized mean.

4. Normalize the frame:

$$\hat{x_j} = \frac{x_j - \mu_x}{\sigma_x}, \tag{2.13}$$

   where $\hat{x_j}$ represents the normalized sample of the frame.

5. Compute the cross-correlation between the normalized frame and each normalized reference pitch frequency:

$$R_k = \sum_{j=1}^{M} \hat{x_j} \cdot \hat{y}_{kj}, \tag{2.14}$$

where R_k denotes the correlation coefficient with the kth reference pitch frequency.

6. Identify the pitch frequency f_0 with the highest correlation coefficient:

$$f_0 = \arg\max_k R_k. \tag{2.15}$$

The NCF algorithm demonstrates robustness to noise, enabling accurate estimation even in challenging environments, and making it suitable for real-time pitch frequency estimation applications [205]. However, NCF assumes a dominant pitch frequency, which limits its accuracy in signals with multiple overlapping frequencies or harmonics where the estimation accuracy is influenced by the range of reference pitch frequencies used [206]. Additionally, the computational complexity is impacted by the number of reference pitch frequencies, which poses limitations in certain applications with a large number of possible pitch frequencies [21].

2.2.3.2 Pitch Estimation Filter (PEF)

The Pitch Estimation Filter (PEF) algorithm employs a filter bank to enhance the pitch-related components of a signal while suppressing unwanted noise and interference [207]. The steps involved in the PEF algorithm can be described as the following:

1. The input signal is decomposed into sub-bands using a filter bank:

$$x_{\text{sub}}(n) = \sum_{k=1}^{K} h_k(n) * x(n), \tag{2.16}$$

   where $x(n)$ represents the input signal, $x_{\text{sub}}(n)$ is the sub-band signal, $h_k(n)$ denotes the impulse response of the kth filter in the filter bank, and $*$ represents the convolution operation.

2. A nonlinear operation is applied to each sub-band signal to enhance the pitch-

related components:

$$y_{\text{sub}}(n) = F[x_{\text{sub}}(n)], \tag{2.17}$$

where $F[\cdot]$ represents the nonlinear operation applied to each sub-band signal.

3. The enhanced sub-band signals are combined to obtain the pitch estimation f_0:

$$f_0 = \arg\max_{n} \left| \sum_{k=1}^{K} y_{\text{sub}}(n) \right|. \tag{2.18}$$

The PEF algorithm effectively enhances the pitch-related components of the signal, leading to improved accuracy in pitch frequency estimation [208]. By incorporating a filtering approach, PEF enables noise reduction and enhances robustness against interference [207]. Furthermore, PEF allows for adjustable filter characteristics, making it adaptable to different signal and noise conditions [208]. Furthermore, the performance and computational complexity of PEF are heavily dependent on the filter design and the selection of nonlinear operations, necessitating careful consideration and optimization [209].

2.2.3.3 Cepstrum Pitch Determination (CEP)

The Cepstrum Pitch Determination algorithm utilizes the cepstral representation of a signal to determine its fundamental frequency [210]. The algorithm involves the following steps:

1. Obtain the magnitude spectrum of the input signal:

$$X(k) = \left| \mathcal{F}(x(n)) \right|, \tag{2.19}$$

where $X(k)$ represents the magnitude spectrum of the input signal $x(n)$, and $\mathcal{F}$ denotes the Fourier transform.

2. Take the logarithm of the magnitude spectrum to obtain the log-spectrum:

$$Y(k) = \log(X(k)). \tag{2.20}$$

3. Apply the inverse Fourier transform to obtain the cepstrum:

$$y(n) = \mathcal{F}^{-1}(Y(k)). \tag{2.21}$$

4. Identify the peak in the cepstrum corresponding to the pitch period, t_0:

$$t_0 = \arg\max_n y(n). \tag{2.22}$$

5. Calculate the pitch frequency estimate using the pitch period t_0:

$$f_0 = \frac{f_s}{t_0}, \tag{2.23}$$

   where f_s represents the sampling frequency.

Cepstrum analysis effectively separates the pitch-related components from background noise by utilizing the quefrency domain, resulting in robust pitch frequency estimation [211]. It can handle signals with multiple harmonics or complex spectral structures, making it suitable for various sound sources while also the implementation of CEP allows for real-time pitch estimation applications [212]. Nevertheless, cepstrum-based analyses may encounter periodicity ambiguity in the presence of overlapping harmonics which makes the pitch estimation performance highly dependent on the presence and prominence of the harmonic spectral peaks [213].

2.2.3.4 Log-Harmonic Summation (LHS)

The Log-Harmonic Summation algorithm estimates the fundamental frequency by summing the logarithmic amplitudes of harmonic peaks in the spectrum [214]. The algorithmic steps of the LHS algorithm can be found below:

1. Obtain the magnitude spectrum of the input signal:

$$X(k) = \left|\mathcal{F}(x(n))\right|, \tag{2.24}$$

   where $X(k)$ represents the magnitude spectrum of the input signal $x(n)$, and $\mathcal{F}$ denotes the Fourier transform.

2. Take the logarithm of the magnitude spectrum:

$$Y(k) = \log(X(k)). \tag{2.25}$$

3. Identify the peaks, $H(k)$ in the logarithmic spectrum corresponding to harmonics:

$$H(k) = \{k : Y(k) > Y(k-1) \text{ and } Y(k) > Y(k+1)\}. \tag{2.26}$$

4. Sum the logarithmic amplitudes of the harmonic peaks:

$$L(k) = \sum_{k \in H(k)} Y(k). \tag{2.27}$$

5. Calculate the pitch frequency estimate f_0 using the Log-Harmonic Sum $L(k)$:

$$f_0 = \frac{f_s}{\text{argmax}(L(k))}, \tag{2.28}$$

where f_s represents the sampling frequency.

LHS is effective in extracting the harmonic structure of the signal, making it strong for noise interference [215]. It can accurately estimate the pitch frequency even in the presence of overlapping harmonics or non-periodic components, and in real-time pitch estimation applications [216]. Yet, the accuracy of pitch estimation depends on the prominence and accuracy of spectral peak detection, which can be affected by noise or variations in the signal [217].

2.2.3.5 Summation of Residual Harmonics (SRH)

The Summation of Residual Harmonics (SRH) algorithm estimates the fundamental frequency by summing the residual harmonics obtained from the difference between the original signal and a synthesized harmonic model [218]. The algorithm consists of the following steps:

1. Generate a harmonic model of the input signal by synthesizing harmonics:

   $$\hat{x}(n) = \sum_{k=1}^{K} A_k \sin(2\pi f_0 k n T_s + \phi_k), \tag{2.29}$$

   where $\hat{x}(n)$ represents the synthesized harmonic model, K is the number of harmonics, A_k and ϕ_k denote the amplitude and phase of the kth harmonic, f_0 is the estimated fundamental frequency, n represents the sample index, and T_s is the sampling period.

2. Calculate the residual signal by subtracting the harmonic model from the original signal:

   $$r(n) = x(n) - \hat{x}(n), \tag{2.30}$$

   where $x(n)$ is the original input signal.

3. Identify the peaks in the magnitude spectrum of the residual signal corresponding to residual harmonics, $H(k)$:

   $$H(k) = \left\{k : \left|\mathcal{F}(r(n))\right|(k) > \text{threshold}\right\}, \tag{2.31}$$

   where $\mathcal{F}$ denotes the Fourier transform, $\left|\mathcal{F}(r(n))\right|(k)$ represents the magnitude spectrum of the residual signal, and the threshold is a predetermined value.

4. Sum the identified residual harmonics:

   $$f_0 = \sum_{k \in H(k)} k \cdot f_0. \tag{2.32}$$

The SRH algorithm focuses on the residual components, making it more resilient to noise and interference in the original signal which enables it to accurately estimate the pitch frequency in signals with complex spectral structures or non-harmonic components [219]. Nonetheless, the accuracy of pitch estimation is highly dependent on the identification of spectral peaks and the optimization of the appropriate threshold [218].

2.3 Dissimilarity Metrics and Temporal Similarity Measures

Dissimilarity metrics and temporal similarity measures are essential for data analysis and pattern recognition tasks. Temporal similarity measures are designed to capture temporal similarities between time series data. The choice of metric depends on the specific characteristics of the data and analysis goals.

2.3.1 Linear Dissimilarity Metrics

Dissimilarity in time series is the difference between two different time series, and there are several distance-based dissimilarity metrics used in literature. These include Euclidean, square Euclidean, Manhattan, Chebychev, and their general form: Minkowski distances[147]. These dissimilarity metrics are frequently used to quantify the numerical difference between time series data by matching the samples one-to-one, which necessitates that the time series data be of equal length [146]. The dissimilarity metrics used in this study and their calculation formula, where x and y indicate the samples, can be found in Table 4.1.

Table 2.1: Common dissimilarity metrics and their formulation.

Dissimilarity Metric	Formula
Minkowski	$(\lvert x_1 - y_1\rvert^p + \lvert x_2 - y_2\rvert^p)^{\frac{1}{p}}$, p: order
Euclidean	$\sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$
Square Euclidean	$(x_1 - y_1)^2 + (x_2 - y_2)^2$
City Block	$\lvert x_1 - y_1\rvert + \lvert x_2 - y_2\rvert$
Chebyshev	$max(\lvert x_1 - y_1\rvert, \lvert x_2 - y_2\rvert)$

Given the formulae of the dissimilarity metrics, each metric has distinct properties, advantages, and disadvantages, making them suitable for a variety of data and analysis tasks.

The Minkowski distance is a generalized distance metric that includes, as special cases, the Euclidean and Manhattan distances [220]. It allows for distance calculation customization by varying the parameter p. However, selecting an appropriate value for p can be subjective and have a significant impact on the results.

The Euclidean distance is the most commonly used and well-known dissimilarity metric, calculating the straight-line distance between two points in an n-dimensional space [221]. It is easy to compute and interpret, and it is widely used in a wide range of applications. But, it is assumed that the dimensions are equal in importance and are not affected by scaling differences.

The squared Euclidean distance is similar to the Euclidean metric, but it avoids taking the square root. It reduces the computational cost of calculating square roots, making it more suitable for large-scale applications [222]. However, it magnifies the effect of large coordinate differences and can result in distorted dissimilarity measurements.

The city block distance, also known as the Manhattan distance, calculates the distance between two points by adding their absolute coordinate differences. When the dimensions are not equally weighted or have different scales, it provides an appropriate measure [223]. But, it is affected by the dimensions' ordering and scaling.

The Chebyshev distance, also known as the chessboard distance, calculates the greatest difference between two points' coordinates along any dimension. It focuses on the maximum difference and is appropriate when outliers are expected or extreme values are desired [224]. On the other hand, it disregards intermediate differences between coordinates and may not be appropriate in cases where subtle variations are significant.

2.3.2 Dynamic Time Warping (DTW)

Dynamic Time Warping (DTW) is a similarity measure specifically designed for comparing sequences with varying lengths or temporal distortions [225]. DTW is a dynamic programming optimization and similarity measurement algorithm used to measure phase shifts between two time series [226]. It finds the optimal alignment between two sequences by warping the time axis and minimizing the total distance between corresponding elements and is useful for comparing time series data that may exhibit phase shifts or temporal distortions [227]. However, it has a drawback of its computational complexity due to its quadratic function optimization [228].

The warping path $W = w1, w2, ..., wL$ is optimized for the two sequences $X = x1, x2, ..., xM$ and $Y = y1, y2, ..., yN$ as follows:

$$Dist(W) = \sum_{l=1}^{L} c(w_{Mi}, w_{Nj}), \tag{2.33}$$

, where L is the length of the warping path and $c(wM, wN)$, is the distance metric between the ith and jth elements of the sequences X and Y, respectively. The main DTW value is then computed using the $Dist(W)$ values for each corresponding sample of i and j for the compared time series, such as [226]:

$$D(i,j) = \begin{cases} ||x_1(i), x_2(j)|| + min\{D(i,j-1), D(i-1,j), D(i-1,j-1)\}, & \text{if } i,j > 1 \\ ||x_1(i), x_2(j)|| + D(i,j-1), & \text{if } i = 1, j > 1 \\ ||x_1(i), x_2(j)|| + D(i-1,j), & \text{if } j = 1, i > 1 \\ ||x_1(i), x_2(j)||, & \text{if } j = 1, i = 1 \end{cases} \tag{2.34}$$

This formula leads the algorithm into an alignment path optimization problem in a 2D map of the $DTW(i, j)$ values along the path from one edge to the other. To solve this optimization problem, some conditions must be met for the constrained optimization of the DTW algorithm, such as boundary condition, monotonicity condition, and step size constraint [229]. Taking into account those constraints, the optimal path is found, whose metric values as given in Equation 2.34 yield the warping distance.

2.3.3 Temporal Similarity Measures

Cross-correlation and cross-covariance are mathematical techniques used to measure the similarity and relationship between two time series signals, and they are essentially used in signal processing, time series analysis, image processing, and machine learning [230].

Cross-correlation is based on the displacement of one data part into another, inferring positional similarity, which measures the similarity between two signals by examining the degree of similarity between their respective time shifts [160]. Given two signals $x(t)$ and $y(t)$, the cross-correlation function between them is defined in Equation 2.35.

$$C_{xy}(t) = \int x(t) \cdot y(t+\tau)dt. \tag{2.35}$$

where τ stands for the temporal displacement between the signals. Cross-correlation is based on linear relationships and stationary signals and can detect time delays or phase shifts between two signals, which can help with synchronization, alignment, and event detection [231]. However, it is sensitive to noise and may produce inaccurate results when signal-to-noise ratios are low.

Cross-covariance measures the linear relationship and similarity between two signals, taking into account their mean values, which can be used in a multivariate fashion during a classification problem [232]. The cross-covariance function between two signals $x(t)$ and $y(t)$ is defined in Equation 2.36.

$$K_{xy}(t_1, t_2) = E[(x(t_1) - \mu_x(t_1)) \cdot (y(t_2) - \mu_y(t_2))], \tag{2.36}$$

where $E[]$ denotes the expected value, μ_x and μ_y represent the means of the signals $x(t)$ and $y(t)$, respectively. Cross covariance provides information about linear dependence, lead-lag relationships, and similarity between two signals, making it valuable in finance, econometrics, and multivariate analysis [233]. However, it can be sensitive to outliers and non-linear relationships and assumes that the signals are stationary and their statistical properties remain constant over time. It is useful for investigating cause-effect relationships, time series forecasting, and dynamic analysis [234].

2.4 Machine Learning Classifiers

Machine learning classifiers are algorithms that use features to categorize or predict the class labels of input data. The most appropriate classifier is determined by factors such as the specific problem at hand, the characteristics of the data, and the desired balance of interpretability, accuracy, and computational efficiency [235]. To make an informed decision, it is necessary to conduct experiments with various classifiers and carefully consider the problem's unique aspects.

2.4.1 K-Means Clustering

The k-means clustering algorithm is an unsupervised technique for pattern recognition and data grouping based on features [236]. The algorithm minimizes the sum of squared distances within each cluster by iteratively updating the centroids. The basic k-means algorithm is shown below:

1. Begin the algorithm by selecting k cluster centroids at random.
2. Use the Euclidean distance to assign each data point to the nearest centroid.
3. Take the mean of all data points assigned to each centroid to recalculate the centroids.
4. Steps 2 and 3 should be repeated until convergence occurs, where the centroids no longer change significantly, or until the maximum number of iterations is reached.

K-means is a fast and simple algorithm that works well with large datasets and high-dimensional data and it is sensitive to the initial centroid selection and may converge to suboptimal solutions [237]. It has applications in a wide range of fields, such as image segmentation, customer segmentation, anomaly detection, data structure exploration, identifying homogeneous groups, and reducing dataset dimensionality [238, 239, 240]. However, it may not be appropriate for datasets with complex non-linear relationships or clusters with irregular shapes [241].

2.4.2 K-Nearest Neighbour (k-NN) Classifier

The k-nearest neighbor (k-NN) algorithm is a supervised, non-parametric classification algorithm that classifies new data points by considering their proximity to known data points [242]. It is straightforward to understand and implement and can handle noisy data better than other classifier algorithms. K-NN follows the following algorithmic steps to classify the data:

1. Save the training data points and their associated class labels.
2. Calculate the distances to all training data points for a given test data point using a distance metric (e.g., Euclidean distance).
3. Choose the k closest neighbors based on the shortest distances.
4. By majority vote among the k neighbors, assign the class label to the test data point.
5. Repeat steps 2-3-4 for each test data point.

K-NN is an algorithm that does not make strong assumptions about the underlying data distribution but is computationally expensive [243]. It is suitable for pattern recognition, recommendation systems, and anomaly detection, and is beneficial when dealing with complex and non-linear decision boundaries or datasets with overlapping classes [244, 245]. However, k-NN-based algorithms may not be suitable for datasets with a large number of features or imbalanced class distributions, unless they are modified for the task [246].

2.4.3 Decision Trees

Decision trees are algorithms that are used to classify data by applying rules to divide it into smaller groups, with the following computation steps [247]:

1. Initiate the algorithm parameters such as maximum depth, and minimum number of samples per leaf.
2. Choose the most informative feature as the tree's root node.
3. Divide the data into child nodes based on the selected feature.
4. Repeat steps 2-3 for each child node.

5. Continue recursively splitting the data until a stopping criterion is met.
6. Assign the majority class label to each leaf node's samples.

Decision trees have the ability to partition data by selecting the most informative feature at each node, guided by a splitting criterion like information gain or the Gini index [248]. This process results in a hierarchical structure where predictions are made at each leaf node. Decision trees offer several advantages, such as their ease of interpretation and visualization, ability to handle both categorical and numerical features, and capability to capture non-linear relationships [249]. However, they can be sensitive to minor data variations and prone to overfitting, and may not be well-suited for complex and non-linear class boundaries [250].

2.4.4 Support Vector Machines (SVM)

Support vector machines (SVM) seek a hyperplane that best separates the training data by optimizing the margin between each class, in which testing samples are assigned to a class based on their proximity to the hyperplane's side while taking into account parameters such as kernel type and coefficients [251]. An SVM algorithm employs the following steps:

1. Map the input data to a high-dimensional feature space using a kernel function.
2. Find the optimal hyperplane in the feature space that maximizes the margin between different classes.
3. Classify the new data samples based on their position relative to the hyperplane.

SVMs are effective in handling high-dimensional data and data with complex, non-linear decision boundaries, and can be memory-efficient when utilizing kernel functions [252]. However, they may suffer from slow training time when dealing with large datasets and can be sensitive to the choice of hyperparameters and the presence of outliers [253].

2.4.5 Naive Bayes Classifier

Naive Bayes classifiers are based on Bayes' theorem and assume independence among features, and they calculate the posterior probability of each class and assign the samples to the class with the highest probability [254]. The implementation steps of a Bayesian classifier are as follows:

1. Estimate the class prior and class-conditional probabilities based on the training data.
2. Using Bayes' theorem, compute the posterior probabilities for each class for a given input instance.
3. Assign the input instance the class label with the highest posterior probability.

The Bayes theorem for calculating the posterior probability is shown in Equation 2.37.

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}, \tag{2.37}$$

where $P(c|x)$ refers to the posterior probability, $P(x|c)$ is the likelihood, $P(c)$ is the class prior probability, and the denominator term $P(x)$ represents the predictor prior probability [255].

Naive Bayes classifiers are suitable for both binary and multi-class classification problems, and large datasets due to their fast computational times [249]. They are useful when dealing with high-dimensional data and large feature spaces, but may not be suitable when the independence assumption among features is violated or when capturing complex feature interactions is crucial [256].

2.5 Validation and Performance Evaluation

Validation and performance evaluation are critical for determining algorithm efficacy and reliability, evaluating method performance, comparing results to ground truth, and computational success, and subjecting it to rigorous testing [257]. Hereby, the confusion matrix and its commonly used metrics, and k-fold cross-validation methods are explained in this section.

2.5.1 Confusion Matrix and Common Performance Metrics

The confusion matrix is a table that summarizes the performance of a classification model by displaying the counts of true positive (TP), true negative (TN), false positive (FP), and false negative (FN) predictions, and can be found in Table 2.2 [258]. It is widely used in machine learning to evaluate the performance of classification models.

Table 2.2: Confusion matrix for two classes

		Actual Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

A variety of widely utilized performance metrics can be calculated based on the values in the confusion table. These metrics include sensitivity, specificity, precision, and accuracy, which are commonly used, along with more intricate metrics such as the f1-score and Mathew's correlation coefficient [259]. These performance metrics provide valuable insights into the performance of classification models and aid in making informed decisions based on the specific requirements of the problem at hand [260].

2.5.1.1 Sensitivity (Recall or True Positive Rate)

Sensitivity, also known as recall or true positive rate, quantifies the ratio of correctly predicted positive instances (TP) to the total actual positive instances (TP and FN). It evaluates the model's ability to capture all positive instances and can be calculated using Equation 2.38.

$$\text{Sensitivity} = \frac{TP}{TP + FN} \tag{2.38}$$

Sensitivity is essential when the cost of false negatives is high and is frequently used in situations where the goal is to maximize the identification of positive instances, such as in fraud detection or medical diagnosis. When the cost of false positives is high or the negative class is more important, the sensitivity might not be the best metric.

2.5.1.2 Specificity (True Negative Rate)

Specificity quantifies the proportion of correctly predicted negative instances (TN) out of the total actual negative instances (TN and FP). It measures the model's ability to avoid false positives and can be calculated using Equation 2.39 [261].

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{2.39}$$

When the cost of false positives is high and accurately identifying negative instances is important, such as in the classification of disorders or the detection of product flaws, specificity is helpful [262]. When the cost of false negatives is high or the positive class is more important, it might not be the best metric to use.

2.5.1.3 Precision

Precision measures the proportion of correctly predicted positive instances (TP) out of the total predicted positive instances (TP and FP). It provides insights into the model's ability to avoid false positives and can be calculated using Equation 2.40 [263].

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2.40}$$

Precision is valuable when the cost of false positives is high and the goal is to minimize them. It is commonly employed in tasks where reducing false positives is crucial, such as in legal case predictions or credit risk assessment [264]. However, precision may not be the most appropriate metric when the cost of false negatives is high or when the negative class is more critical.

2.5.1.4 Accuracy

Accuracy evaluates the overall correctness of the model's predictions and is defined as the ratio of correct predictions (TP and TN) to the total number of predictions (TP, TN, FP, and FN). It can be calculated using Equation 2.41 [265].

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.41}$$

When the dataset's classes are evenly distributed and have comparable importance, accuracy is frequently used. When both positive and negative misclassifications have comparable effects, it is appropriate for evaluating the model's overall performance [261]. But when classes are unbalanced and the costs of misclassification vary greatly between classes, accuracy can be deceiving.

2.5.1.5 F1 Score

The F1 score combines precision and recall into a single metric that balances both measures. It is the harmonic mean of precision and recall and can be calculated using Equation 2.42 [265].

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.42}$$

The F1 score is commonly used when a balance between precision and recall is desired [266]. It is useful in tasks where achieving a balance between precision and recall is important, such as in information retrieval or sentiment analysis [267].

2.5.1.6 Matthew's Correlation Coefficient (MCC)

Matthew's correlation coefficient is a comprehensive measure of binary classification quality, considering all four elements of the confusion matrix. It ranges from -1 to 1, with 1 representing a perfect prediction, 0 representing a random prediction, and -1 representing complete disagreement between the prediction and the actual class [265]. MCC can be calculated using Equation 2.43.

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{2.43}$$

It is appropriate when a comprehensive evaluation metric that takes into account all elements of the confusion matrix is required, which makes MCC an appropriate metric to use for imbalanced classes [268]. However, when the classes are balanced or the cost of misclassification varies significantly between classes, MCC may not be the best metric to use.

2.5.2 K-Fold Cross Validation

K-fold cross-validation is a widely used technique for evaluating the performance and robustness of machine learning models [269]. The main objective of k-fold cross-validation is to assess the efficiency of pattern recognition models [270].

The k-fold cross-validation procedure involves dividing the dataset into k equal-sized subsets or folds. Each fold, denoted as D_i, contains approximately n/k instances, where n is the total number of instances in the dataset. The steps of k-fold cross-validation can be summarized as follows:

1. Divide the dataset into k equal-sized subsets (folds), denoted as D_i.
2. For each iteration, use one fold D_i as the test set, and the remaining $k - 1$ folds as the training set.
3. Train the model on the training set and evaluate its performance on the test set.

4. Calculate the performance metrics for each iteration based on the model's predictions and the true values from the test set.
5. Aggregate the calculated performance metrics over all k iterations to provide an overall assessment of the model's performance.

K-fold cross-validation is commonly employed in machine learning for tasks such as model selection and hyperparameter tuning and it is also useful when dealing with limited data samples and serves as a resampling procedure [271]. It allows for a fair comparison among multiple models or parameter settings by considering the combinations of the different partitions of the dataset [272]. By using k-fold cross-validation, a more accurate estimate of the model's performance and generalization ability can be obtained, reducing the risk of overfitting and providing a more realistic assessment of the model's behavior on unseen data [273].

CHAPTER 3

RELATED ANOMALY DETECTION STUDIES AND RECENT DEVELOPMENTS

This chapter provides a general view of the anomaly detection methods that are commonly used for a variety of purposes and approaches. Moreover, the recent and directly related methods in the literature with their problems in computation are also provided in this chapter. The proposed outlier and anomaly detection approach focuses on three major classes of methods: statistical, frequency domain-based, and dissimilarity-based approaches. These classes are chosen based on their relevance to the development and testing stages of the proposed approach, as well as their comparison to the current version of the proposed approach. While there are other classes of outlier and anomaly detection methods available, they are not specifically included in the proposed approach due to their limited relevance or comparability. Therefore the following anomaly detection families and methods are selected based on their relationship with the proposed approach in their methodologies, and their common usages.

3.1 Common Outlier and Anomaly Detection Methods in Literature

3.1.1 Statistical Approaches

From a variety of outlier detection methods [274], three widely used statistical anomaly detection methods, namely z-score, box-plot, and the Grubbs method, are chosen for this thesis and the development process of the proposed algorithm. These methods are employed for comparing different versions of the proposed approach. Statistical outlier and anomaly detection algorithms are commonly utilized in the analysis of time series data to detect unusual patterns or observations that deviate significantly from the expected behavior. However, these methods assume stationarity in the data and may not be appropriate for non-stationary time series data.

3.1.1.1 z-score Test

The z-score test is a commonly utilized method for detecting outliers in univariate datasets [275]. It calculates a standardized z-value for each data point using the Equation 3.1 [93].

$$z = \frac{Y - \overline{Y}}{\frac{\sigma}{\sqrt{n}}}, \tag{3.1}$$

where z represents the standardized value, Y is the value of a data point, $\overline{Y}$ is the mean of the data, σ is the standard deviation, and n is the sample size. The z-score measures the number of standard deviations that a data point deviates from the mean.

To identify outliers, a threshold value for z is selected. Typically, values with $|z| > 3$ are considered outliers. This choice is based on the observation that values with $|z| < 3$ account for approximately 99% of the data and are close to the mean value [93]. Values with $|z| > 3$ are regarded as extreme and flagged as outliers.

It is important to note that the z-score test is a parametric method that relies on the statistical moments of the data, which implies that its effectiveness can be influenced by the variability of outlier fold changes in comparison to the data without outliers [44, 276]. If individual outliers exhibit high variances in fold changes or relatively low fold changes compared to the rest of the data, the z-score method may not accurately detect these outliers. Nevertheless, despite this limitation, the z-score test remains widely employed for outlier detection in various types of datasets due to its computational simplicity [277, 278].

3.1.1.2 Box-plot Test

The box-plot is a visual representation of standardized data that offers valuable insights into the distribution of a dataset [97]. It presents key summary statistics, including the minimum and maximum values, median, quartiles, and lower and upper extreme value limits [279]. The box portion of the plot represents the interquartile range (IQR), formed by the first and third quartiles, encompassing the middle 50% of the data around the median value [280].

To detect outliers using the box-plot method, horizontal lines are drawn above and below the box to represent the limits within which 95% of the data typically falls, and observations that lie beyond these horizontal lines are considered potential outliers [281]. In Figure 3.1, the red crosses indicate such outliers.

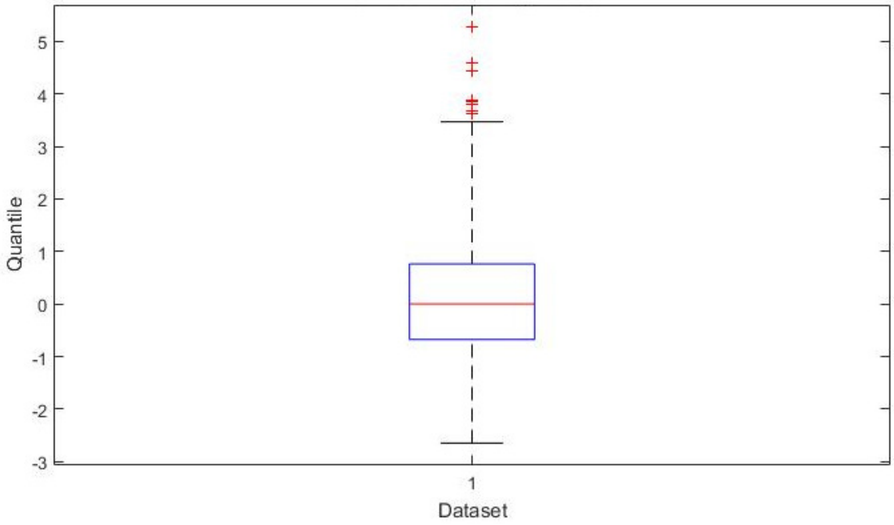


Figure 3.1: A representation of the box-plot graph. The red crosses denote the outliers.

However, similar to the z-score method, the box-plot approach's effectiveness can be affected by the variability of outlier fold changes and their relationship to the rest of the data [282]. Another drawback of the box-plot method is its uniform application across different types of datasets, regardless of their underlying nature, which may cause a lack of adaptability to specific data characteristics that can result in suboptimal outlier detection outcomes in certain scenarios [44].

3.1.1.3 Grubbs Method

The Grubbs' method, introduced by Grubbs in 1969 [93], is a statistical approach commonly used for outlier detection and anomaly detection. It employs the z-value as a measure to identify outliers. The test statistic G is defined in Equation 3.2.

$$G = \frac{|Y_i - \bar{Y}|}{s}, \tag{3.2}$$

where Y_i represents the ith sample that could potentially be an outlier, $\bar{Y}$ is the mean of the data, and s is the standard deviation.

One notable characteristic of the Grubbs' method is its suitability for handling outliers in time series data [40]. The Grubbs' test assumes a normal distribution for the data and can be applied iteratively until no outliers remain in the dataset [44]. Unlike the z-score method that operates on the entire dataset at once, the Grubbs' test examines individual samples one by one during each iteration. This feature makes the Grubbs' method more appropriate when there are only a few outliers or a single outlier present in the data [48].

It is important to consider that due to its iterative nature, the Grubbs' method can be more computationally demanding compared to the z-score method [44]. Consequently, the Grubbs' test is recommended when there are a small number of outliers or a single outlier in the dataset, as its practicality may diminish when dealing with large datasets containing numerous potential outliers [103].

3.1.2 Frequency Domain Related Approaches

The frequency domain-based approach is a class of outlier and anomaly detection methods that focuses on analyzing the frequency content of the data. These techniques utilize various frequency domain transformations such as Fourier analysis, wavelet transforms, or spectral analysis to examine the data in the frequency domain.

By analyzing the frequency components, these methods aim to identify unusual patterns or irregularities that may indicate the presence of outliers or anomalies in periodic time series data.

3.1.2.1 Spectral Residual (SR) Method

: The Spectral Residual (SR) method is a frequency domain-based approach for anomaly detection that utilizes the spectral properties of the data [283]. It aims to identify outliers by examining the difference between the original spectrum and a smoothed spectrum where the underlying assumption is that anomalies possess distinct spectral signatures that stand out in the residual spectrum [284].

Let $X(f)$ represent the magnitude spectrum of the data. The Spectral Residual $(R(f))$ is computed by taking the logarithm of the magnitude spectrum and subtracting the smoothed logarithmic spectrum:

$$R(f) = \log(|X(f)|) - S(\log(|X(f)|)), \tag{3.3}$$

where $\log$ denotes the natural logarithm and S refers to the smoothing function applied to the logarithmic spectrum.

The SR method is particularly effective in detecting anomalies that possess unique spectral characteristics by comparing the spectral residual values with predefined thresholds [283]. However, it relies on the assumption that anomalies exhibit discernible spectral signatures, which may not always hold true in practical scenarios [285].

3.1.2.2 Singular Spectrum Analysis (SSA)

Singular Spectrum Analysis (SSA) is a frequency domain-based technique that decomposes a time series into a set of components known as singular vectors [286]. These singular vectors are derived from the time series data and provide valuable information about its behavior. Anomaly detection using SSA involves examining the characteristics of these components to capture both local and global patterns in the data, enabling the detection of anomalies of various scales [287].

When applying SSA, the time series data $X(t)$ is decomposed into singular vectors or components denoted as $C_i(t)$. Unlike some other methods, SSA does not rely on specific assumptions about the data distribution, making it more versatile in handling different types of time series [288]. However, one challenge in using SSA for anomaly detection is determining the appropriate number of singular vectors or components consider, which depends on the specific characteristics of the data and the anomalies being targeted [289]. Additionally, SSA may not be as effective for datasets with irregular or noisy patterns, as these can complicate the decomposition process and potentially impact the accuracy of anomaly detection results [287].

3.1.2.3 Autoencoder-based Anomaly Detection

Autoencoder-based anomaly detection is a machine learning approach that leverages autoencoder neural networks to identify anomalies in the frequency domain [290]. By learning the normal behavior of the data, it can detect deviations from this learned representation, indicating the presence of anomalies [291]. The frequency domain representation of the data is used as input to an autoencoder neural network, which is trained to approximate the input itself, with the goal of reconstructing the original frequency domain representation as accurately as possible [292]. The reconstruction error is then calculated as the difference between the input X and the output of the autoencoder $f(X)$. Anomalies are detected based on the magnitude of the reconstruction error which measures the dissimilarity between the input and the output of the autoencoder and is commonly used as an indicator of anomaly likelihood [293].

Autoencoder-based methods offer several advantages. They can capture complex patterns and nonlinear relationships in the data, allowing them to effectively identify anomalies with diverse characteristics [294]. Furthermore, they can detect both global anomalies, which affect the entire dataset, and local anomalies, which are specific to certain regions or patterns within the data [295]. Nevertheless, the training process of autoencoders can be computationally expensive, particularly for large datasets, which may hinder their scalability to real-world applications [291]. Furthermore, the performance of autoencoder-based methods is heavily dependent on the quality and representativeness of the training data [296].

3.1.2.4 Power Spectral Density (PSD) Analysis

Power Spectral Density (PSD) analysis is a technique used to examine the power distribution across different frequencies in the frequency domain [297]. Anomalies can manifest as unusual spikes or drops in power within specific frequency bands, and by estimating the power at each frequency, enables the detection of anomalies by identifying deviations from the expected power distribution [298]. To perform PSD analysis, the magnitude spectrum of the data, denoted as $X(f)$, is utilized as in Equation 3.4.

$$S(f) = |X(f)|^2. \tag{3.4}$$

Anomalies can be detected by comparing the PSD values with a predefined threshold or by analyzing the power distribution across different frequency bands and are particularly effective in identifying anomalies that exhibit distinct frequency characteristics [299]. It provides valuable insights into the power distribution of the data across various frequency bands, which can aid in understanding the underlying patterns and abnormalities present in the dataset [87]. On the other hand, determining an appropriate threshold for anomaly detection can be challenging and may require domain knowledge or prior information about the expected power distribution [300].

3.1.2.5 Hilbert Transform-based Methods

Hilbert Transform-based methods analyze the analytic signal derived from the original signal using the Hilbert Transform, which uses the phase information of the analytic signal to identify anomalies [301]. Let $x(t)$ be the time series data. The Hilbert Transform of $x(t)$ yields the analytic signal $z(t)$ as in Equation 3.5.

$$z(t) = x(t) + i \cdot H(x(t)), \tag{3.5}$$

where $H(\cdot)$ represents the Hilbert Transform.

Hilbert Transform-based methods can detect phase-based anomalies in analytic signals by analyzing changes in their instantaneous phase or frequency as they are useful for detecting sudden changes or transient events [302]. However, Hilbert Transform-based methods may be sensitive to noise and outliers in the data, which may obstruct the phase information [303].

3.1.2.6 Cepstrum Analysis

Cepstrum analysis is a technique that involves transforming the magnitude spectrum of the data into the cepstral domain [304]. To perform cepstrum analysis, the inverse Fourier Transform of $\log(|X(f)|)$ is taken to obtain the cepstral coefficients, which represent the envelope of the spectrum, and anomalies can be identified by analyzing the variations in these coefficients [305].

Cepstrum analysis is particularly effective in capturing anomalies in signals with complex frequency components, which can uncover irregularities and unexpected patterns that are not easily detected in the original magnitude spectrum [304]. However, selecting the appropriate cepstral coefficients or threshold values for anomaly detection can be a difficult optimization problem for multivariate datasets [306].

3.1.2.7 Wavelet-based Methods

Wavelet-based methods are effective in detecting anomalies based on deviations from expected wavelet coefficients and analyzing data in both the time and frequency domains using wavelet transforms [307]. To apply wavelet-based methods, the time series data is decomposed using wavelet transforms, which provide a representation of the data at different scales or resolutions [308]. The wavelet coefficients obtained from the decomposition are then analyzed to locate the deviations or significant changes in the wavelet coefficients as they indicate the presence of anomalies [309].

One of the key advantages of wavelet-based methods is their ability to capture both time and frequency information simultaneously which makes them well-suited for detecting transient or time-varying anomalies that may occur at specific time intervals or exhibit frequency variations [310]. By performing multi-resolution analysis, wavelet-based methods can detect anomalies at different scales, allowing for a comprehensive examination of the data [286]. But, different wavelet bases have varying properties and may be more suitable for specific types of data or anomalies, and the computational complexity of wavelet transforms may pose challenges, particularly when dealing with large datasets [44].

3.1.3 Dissimilarity-based Approaches

The dissimilarity-based approach to anomaly detection revolves around quantifying the dissimilarity or distance between data points in order to identify outliers and anomalies. This family of methods relies on various techniques, such as comparing two time series data, clustering, or non-parametric approaches like nearest neighbor analysis, to identify data points that significantly deviate from the rest [311].

In the dissimilarity-based approach, the focus is on measuring the dissimilarity or distance between data points rather than making assumptions about their underlying distribution [146]. By computing the dissimilarity between data points, it becomes possible to identify those that exhibit distinct characteristics or patterns compared to the majority of the data.

3.1.3.1 Higher Order Time Series Symbolic Aggregate Approximation (HOT-SAX)

HOTSAX (Hierarchical Ordered Time SEries Approximation) is a dissimilarity-based method that uses symbolic representation to detect anomalies in time series data [312]. The HOTSAX algorithm can be computed using the following steps:

1. The time series data $x(t)$ undergoes a discretization algorithm to obtain a symbolic sequence representation. Let $x(t) = [x_1, x_2, \ldots, x_n]$ denote the time series data. Through the discretization process, $x(t)$ is divided into subsequences

of equal length, and symbols are assigned to each subsequence.

2. A distance matrix D is computed to quantify the dissimilarity between pairs of symbolic sequences. The elements of the distance matrix, denoted as D_{ij}, correspond to the dissimilarity between the symbols s_i and s_j. The distance measure used can be a function $d(s_i, s_j)$, such as the Euclidean distance, or a specialized distance function designed for symbolic sequences [313]. The distance matrix can be expressed as shown in Equation 3.6.

$$D = \begin{bmatrix} d(s_1, s_1) & d(s_1, s_2) & \dots & d(s_1, s_m) \\ d(s_2, s_1) & d(s_2, s_2) & \dots & d(s_2, s_m) \\ \vdots & \vdots & \ddots & \vdots \\ d(s_m, s_1) & d(s_m, s_2) & \dots & d(s_m, s_m) \end{bmatrix} \tag{3.6}$$

3. Utilizing the distance matrix, hierarchical clustering is performed to group the symbolic sequences. Initially, each symbolic sequence is treated as an individual cluster. The clustering process involves iteratively merging clusters according to their pairwise dissimilarity.

4. Subsequently, the hierarchical structure is examined to identify anomalies. Anomalies are characterized by clusters or subtrees that exhibit significant deviations from the overall data. By analyzing these aspects, anomalies can be detected within the hierarchical structure.

HOTSAX can handle large-scale datasets and longer time series that are possessing a regular baseline behavior [55]. However, selecting the appropriate parameters is critical for the algorithm's performance.

3.1.3.2 LDOF (Local Distance-based Outlier Factor)

The LDOF (Local Density-Based Outlier Factor) algorithm is a dissimilarity-based method that measures the local outlierness of each data point based on its distance to its neighbors [314]. It calculates an outlier factor, known as the LOF, to quantify the degree of outlierness for each data point where the outliers are detected by comparing the LDOF of each point to a predefined threshold [315].

Let x_i represent the ith data point in a dataset, and $d(x_i, x_j)$ denote the distance between x_i and its neighbor x_j. The local outlier factor for x_i is calculated using Equation 3.7.

$$\text{LOF}(x_i) = \frac{1}{k} \sum_{j=1}^{k} \frac{d(x_i, x_j)}{\max(d(x_j), \epsilon)}. \tag{3.7}$$

Here, $d(x_i, x_j)$ represents the reachability distance between x_i and x_j, k is the number of nearest neighbors considered, and ϵ is a small value. Anomalies are identified if the LOF for a data point exceeds the predefined threshold.

The LDOF algorithm is effective in detecting local outliers and anomalies as it is capable of handling datasets with varying densities or clusters [316]. However, it may not perform well for high-dimensional datasets due to the curse of dimensionality.

3.1.3.3 Autonomous Anomaly Detection (AAD) Method

The autonomous anomaly detection (AAD) method [102] is a non-parametric and unsupervised approach for outlier detection [317]. The algorithm utilizes Voronoi tessellations [318] to partition the data into distinct data clouds.

To identify outliers within the data clouds, the AAD method follows a multi-step process. In the first step, candidate samples for global anomalies, denoted as $\{x\}_{PA,1}$, are identified based on multimodal typicality [319]. Similarly, in the second step, locally anomalous candidate samples, denoted as $\{x\}_{PA,2}$, are identified using locally weighted multimodal typicality [319]. These local and global anomaly candidate samples are combined as $\{x\}_{PA} \leftarrow \{x\}_{PA,1} + \{x\}_{PA,2}$, and the autonomous data partitioning algorithm (ADP) [102] is employed to generate data clouds denoted as $\{C\}_{PA}$ [319].

In the ADP algorithm, the average distances between local areas are used to define the average radius of the local area of influence, referred to as the granularity [317]. The density of the data, forming local cells within the Voronoi tessellation, is calculated within local hyperspheres [317]. In the final stage of the AAD algorithm, the actual outliers, denoted as $\{x\}_A$, are detected from the potential outliers $\{x\}_{PA}$ by considering if they fall below the average support of the data cloud they belong to [317]. This method is particularly designed for time series datasets [319]. However, since the detection of anomalies (i.e., outliers) is empirically determined for each dataset, its application is not automated.

3.2 Frequency Domain Based Outlier Detection (FOD) Algorithm

The Frequency Domain-based Outlier Detection (FOD) approach has been developed as a robust non-parametric algorithm for identifying periodic outliers in time series data [44]. By applying a frequency domain-based outlier detection algorithm, which utilizes the Fourier Transform, the periodic nature of outliers can be effectively captured. In the frequency domain, peaks correspond to periodic behavior, albeit with a periodicity that is inversely related to the periodicity of outliers in the time domain due to the scaling property of the Discrete Fourier Transform (DFT) [320]. Specifically, when there are fewer or more widely spaced periodic outliers in the time domain, resulting in lower frequencies, a peak emerges at the beginning of the frequency domain, indicating a low-frequency value. The periodic behavior of peaks in the frequency domain arises from the harmonics of the primary oscillation frequency, which are influenced by the repeated periodicity of outliers. Consequently, the location of the first peak in the frequency domain holds the most meaningful information. Consequently, two distinct scenarios can be considered for the frequency domain response based on the data's characteristics. First, when the data exhibits negligible trends and a random-like distribution, such as those following normal or log-normal distributions, their frequency domain representation will also exhibit periodic behavior with harmonics of similar amplitudes to the main oscillation frequency peak. Second, when the data contains trends or periodic patterns, as observed in various real-world data such as ECG, annual temperature, and circular/seasonal data, the outliers manifest as peak values within their respective patterns. In such cases, a high peak appears in the initial samples of the frequency domain, indicating low-frequency components, typically dependent on the data collection sampling rate and perception [320]. These high peaks correspond to the main oscillation frequency of the periodic pattern. Since periodic patterns encompass multiple data samples (unlike the "one sample-outlier" scenario in simulated data), they tend to exhibit harmonics with decreasing amplitudes. Detecting the main oscillatory frequency can be challenging for various data types, including those with trends, and high levels of randomness/noise with low frequencies. The difficulty in identifying the main oscillatory frequency in such data can be attributed to:

1. The highest peak in the frequency domain may correspond to a harmonic frequency rather than the main oscillatory frequency.
2. It is possible for there to be peaks preceding the main oscillatory frequency in the frequency domain.
3. The amplitude of the peak representing the main oscillation frequency may be lower than that of other peaks in the frequency domain.

The scaling properties of Fourier Transform, which are known as scaling in frequency and scaling in time, are widely recognized and are inversely proportional to each other [320]. The relationship between the intervals in time (t_{int}) corresponding to the intervals between peaks in the time domain and the intervals in the frequency domain (f_{int}) due to harmonics can be established by exploiting these properties. This formula highlights the inverse relationship between the intervals in time and frequency domains and provides a useful tool for the analysis of periodic outliers in the frequency domain based on their corresponding intervals in the time domain. Therefore, the prehistoric form of the FOD algorithm, named the "Single-step FOD Algorithm," is developed by applying the new peak detection approach and utilizing the implication of the relation between frequency scaling and time scaling in Equation 3.8 in the frequency domain, as described in Section 3.2.1.

$$t_{int} = \frac{L}{f_{int}} \tag{3.8}$$

3.2.1 Single-step FOD Algorithm and Applications

The single-step FOD algorithm aims to detect periodic outliers in the time series data, using the frequency domain information, and the relationships between the time and frequency domains as preliminary defined in Equation 3.8. Although the multi-step versions of the FOD algorithms are planned and their fundamental research and development have been completed in several different studies [145, 321, 322], the multi-step approaches of the FOD and related algorithms do not take part in this thesis study. The single-step FOD algorithm is developed for the time series data in which the outliers are periodically scattered in the data. The algorithm is given below:

1. Compute the Discrete Fourier Transform (DFT) of the time domain data, representing the frequency components in Hertz. Remove the symmetrical half of the frequency domain, centered around 0Hz, as per the Nyquist theorem [323]. This results in a signal length of $\frac{L}{2}+1$, where L is the length of the time domain data.

2. Apply the recent peak detection approach in the frequency domain, described in Section 3.2.3.1, to identify the M largest peaks. The value of M is determined using the formula given in Equation 3.9.

3. Once the M largest peaks are detected, calculate the frequency interval f_{int} by determining the distances between each pair of peak locations and finding the mode of these distance values. Empirical results indicate that with proper selection of M, accurate estimation of f_{int} can be achieved. Utilize Equation 3.8 and the computed f_{int} values to determine the time intervals between the actual outliers.

4. Assume that the absolute global extreme of the time domain data represents an outlier and is part of a sequence of outliers occurring periodically. Locate this outlier sample and shift its position by t_{int}, which allows for the identification of all equidistant outliers exhibiting periodic behavior.

The FOD algorithm can be represented in simple pseudo codes as in 1, but it has its extensions in multiple Fourier Transform-based approaches, namely, the multi-step FOD algorithm (NFOD) as in [145, 321, 322]. But they are not explained in this thesis.

Algorithm 1 Single-Step FOD

```
Compute the Discrete Fourier Transform (DFT) of $x(t)$ using the Fast Fourier Transform (FFT) algorithm
Discard the first symmetric part of the DFT estimation
Find the value of $M$ using Equation 3.9
Find the highest $M$ peak samples with dynamically changing parameters for peak detection (improvements can be made for generalization)
Find $f_{int}$ as the mode of the distances between the locations of all $M$ peaks
if $N$ is odd then
    Calculate $t_{int}$ using Equation 3.8
end if
Find $\max x(n)$, the maximum value of the time domain data
Shift the location of $\max x(n)$ forward and backward by $t_{int}$ to identify the periodic outliers
```

With the application of the FOD algorithm, several analyses were conducted to evaluate its computational performances with some of the benchmark methods. The experimental setup including the simulations and the challenging condition to evaluate the performances of the FOD algorithm can be found in Section 3.2.2.

3.2.2 Simulations to measure FOD algorithm performances

In order to assess the performance of the FOD algorithm, a simulation experiment was conducted, where several other outlier detection methods from the literature were evaluated under the same conditions for comparability. A brief description of these other methods can be found in Section 3.1. The simulations were carried out using a Monte Carlo simulation study with 100 runs. In each iteration, random time series data was generated with periodic outliers placed within it. The independent variables considered in the simulations included the data length, outlier fold change, percentage of outliers relative to the data length, and the data distribution used to generate the baseline signal. Additionally, the computational times of the algorithms were recorded. The performance evaluations were conducted using the F1-score [324] to enable comparisons across different sizes of the evaluation metrics. The F1-score ranges from 0 to 1, with a value of 1 indicating the complete detection of all outliers in the given data. Since there were multiple simulation conditions due to the varying independent variables, the F1 scores were averaged and presented in the tables below. Each table focuses on the impact of a specific independent variable. The F1-scores were calculated by considering the correct identification of outliers, total identification attempts, and false identifications in the F1-score formula.

Firstly, the impact of data length on the performance of outlier detection algorithms was evaluated. Table 3.2 presents the results of the simulation experiment, focusing on the effects of baseline distributions and outlier fold changes on the performance of the outlier detection algorithms. The fold change is determined by generating outliers with varying mean values and a fixed variance parameter of 1, which introduces deviation in the values. The F1 scores were used as the performance metric for comparison.

Table 3.1: The average F1-scores for the effects of data length on the performance of the outlier detection algorithms

Data Length	Box-plot	z-score	Grubbs	AAD	FOD
200	0,129	0,096	0,097	0,015	0,237
1000	0,166	0,170	0,116	0,071	0,337
10000	0,365	0,410	0,306	0,152	0,796

Based on the results presented in Table 3.1, it can be observed that the FOD algorithm achieves the highest F1 scores compared to the other algorithms. Furthermore, there is a significant improvement in performance as the data length increases. This suggests that the FOD algorithm is effective in detecting outliers, and longer data sequences provide more reliable results. By analyzing the table, it can also be observed that the FOD algorithm consistently outperforms the other methods across different baseline distributions and fold change values. The F1 scores indicate that FOD achieves better detection of outliers compared to the alternative algorithms. The performance of all algorithms tends to improve as the fold change increases, indicating that larger deviations in the outlier values facilitate their detection. These findings highlight the effectiveness of the FOD algorithm in detecting outliers, regardless of the baseline distribution and the magnitude of the fold change.

Table 3.2: The average F1-score values for the effects of baseline data distribution and outlier fold change.

Baseline Dist.	Outlier Dist.	Box-plot	z-score	Grubbs	AAD	FOD
Gauss.(0,1)	Gauss. (1,1)	0,381	0,350	0,164	0,000	0,487
	Gauss. (3,1)	0,361	0,414	0,301	0,082	0,846
	Gauss. (5,1)	0,360	0,431	0,295	0,274	0,849
Log-n.(0,1)	Log-n. (1,1)	0,147	0,152	0,110	0,049	0,354
	Log-n. (3,1)	0,147	0,125	0,186	0,1112	0,344
	Log-n. (5,1)	0,146	0,084	0,200	0,094	0,189
t (v = 3)	v = 9	0,146	0,158	0,098	0,033	0,350

According to the results presented in Table 3.2, the FOD algorithm consistently achieves higher average F1 scores compared to the other outlier detection methods, except for the Grubbs method in the case of the largest fold change of outliers in the log-normal distributed baseline signal.

In general, FOD demonstrates strong performance across different baseline distributions and outlier distributions. It particularly excels in detecting outliers in Gaussian distributed baseline and outlier signals, consistently outperforming the other methods. However, there is a slight decrease in the performance of FOD for the largest fold change of outliers in the log-normal distributed baseline signal. This decrease in performance can be attributed to the high inner variance of the log-normal distribution, which makes it challenging for the FOD algorithm to accurately detect the peaks in the frequency domain. The results highlight the effectiveness of the FOD algorithm in various scenarios, especially for Gaussian distributed baseline and outlier signals. It provides reliable outlier detection performance, outperforming the alternative methods in most cases.

The performance of the FOD algorithm was evaluated in terms of the effect of the percentage of outliers, which represents the ratio of the number of outliers to the data length. The results of this investigation provide insights into the algorithm's performance under varying outlier conditions. The results for this simulation are presented in Table 3.3.

Table 3.3: The average F1-scores for the effects of outlier percentage in the data.

Outlier Percentage (%)	Box-plot	z-score	Grubbs	AAD	FOD
1	0,106	0,130	0,116	0,065	0,426
2	0,157	0,205	0,186	0,118	0,402
3	0,315	0,332	0,307	0,079	0,422
4	0,302	0,235	0,083	0,056	0,577

According to Table 3.3, the FOD algorithm exhibits the best performance among the methods in terms of average F1 scores for the effect of outlier percentage in the data. Furthermore, it is observed that the FOD performance improves as the percentage of outliers in the data increases, especially after the 2% outlier percentage condition.

Finally, Table 3.4 provides an overview of the computational times required for each iteration in the Monte-Carlo simulations for the outlier detection methods. The results show that the FOD algorithm demonstrates faster computation compared to the complex algorithms, namely Grubbs and AAD. However, it is relatively slow when compared to the basic methods, such as Box-plot and z-score.

Table 3.4: The average computational time (milliseconds) for each iteration for data length effect.

Data Length	Box-plot	z-score	Grubbs	AAD	FOD
200	2,5	1,6	5,6	6,9	3,6
1000	3,4	2,6	10,2	40,6	4,4
10000	12	3,3	56	2974	18,2

According to the table, the FOD algorithm exhibits faster computational times compared to complex algorithms, such as Grubbs and AAD. However, it is slightly slower than the basic methods, such as Box-plot and z-score. The computational time increases as the data length increases for all the methods, and FOD shows a reasonable increase in computational time compared to the other methods.

3.2.3 Conclusion and Problems of the FOD Algorithm

This paper introduces the frequency domain-based outlier detection method (FOD) for detecting periodic outliers and quasi-periodic patterns in time series data. The FOD algorithm is applied to various distributions, sample sizes, and percentages of outliers, and compared to other outlier detection methods. The Monte Carlo studies show that FOD consistently outperforms box-plot, z-score, Grubbs, and AAD methods in terms of accuracy. It is particularly effective in detecting fully periodic patterns and quasi-periodic patterns in normal distributions. Moreover, FOD demonstrates favorable computational efficiency compared to non-parametric methods and similar computational time to the parametric z-score method. The performance evaluation on real datasets further supports the effectiveness of FOD. Overall, FOD presents a preferable method for detecting periodic outliers and quasi-periodic patterns in time series data.

On the other hand, the FOD algorithm is developed as an offline and proper for the detection of periodic outliers in time series data. Moreover, since it does not utilize a batch-data processing approach such as moving windows, it can not be directly adapted to the real-time environment. Furthermore, a thresholding problem appears that is critically affecting the performance of detecting the periodic outliers in the data. Hereby Section 3.2.3.1 refers to the quick solution to the thresholding problem. However, considering the adaptability and more generalized use, a moving windows-based version of the proposed algorithm is required. Therefore, a new and moving-windows-based approach is developed later, which can be found in Section 3.3.

3.2.3.1 Problem with the FOD Algorithm: Thresholding

In the thresholding process of the FOD algorithm during the frequency domain calculations, the selection of an appropriate threshold value in the frequency domain is crucial. This task involves considering various statistical properties of the data, such as mean, maximum, minimum, and median. Several approaches have been explored to determine the threshold value, including combinations of statistical moments and dynamic adaptation based on data behavior. Each of these methods has shown promise for specific types of data but may not be suitable for others.

Alternatively, instead of using a fixed threshold value, the FOD algorithm adopts a different strategy by detecting the M largest peaks in the frequency domain. The value of M is calculated using Equation 3.9, which takes into account the length of the time series data L and the range between the minimum and maximum values in the frequency domain.

$$M = \frac{\max - \min}{\max} \times L \tag{3.9}$$

In Equation 3.9, M is determined as the product of the normalized range (max-min) divided by the maximum value, and the data length L. This calculation ensures that M is directly proportional to the data length and adjusts the range relative to the maximum value. This dynamic approach in detecting frequency peaks has yielded improved results for the FOD algorithm.

3.3 Windowed Frequency Domain Based Outlier Detection (WFOD)

The WFOD method is derived from the frequency domain-based outlier detection method (FOD) [44] and adapts the moving windows approach that has iterative steps in both the temporal and frequency domain (FOD). In this study, a novel hybrid feature called the WFOD feature is proposed [140], which combines properties from both the time and frequency domains.

The WFOD algorithm, a modified and simplified version of the FOD algorithm, is utilized for feature extraction and modeling purposes. The algorithm is outlined below:

Algorithm 2 WFOD: Windowed FOD

```
1: Compute Periodogram.
2: Limit the searching window in periodogram: Between the frequency values, corresponding to 0.25 seconds and 2 seconds in time domain, which are representing the 240bpm and 30bpm respectively for the extreme cases, by using the relation 3.10 in and 3.11 respectively.
       $f_{int}max = \frac{DataLength}{0.25 * SamplingRate}$    (3.10)
       $f_{int}min = \frac{DataLength}{2 * SamplingRate}$    (3.11)
3: Detect the peaks between $f_{int}min$ and $f_{int}max$.
4: Get the frequency value of the highest peak: $f_{int}$.
5: Find the WFOD periodicity, $t_{int}$, by using Equation 3.12.
       $t_{int} = \frac{2 * WindowSize}{f_{int}}$    (3.12)
```

Following the WFOD algorithm in Algorithm 2, firstly, the periodogram of the data is estimated using a defined window size. The window size is initially chosen to cover at least five heartbeats, allowing for clear patterns to emerge in the frequency domain. The second step involves identifying the principal frequency in the periodogram plot. To ensure accurate detection and exclude extreme or erroneous cases, such as heartbeats above 240 bpm or below 30 bpm, a search window is defined within the periodogram plot. The upper limit of the search window, $f_{int}max$, is computed using Equation 3.10, where the constant 0.25 represents the time interval between beats for a heart rate of 240 bpm. Similarly, Equation 3.11 is used to calculate the lower limit of the search window, $f_{int}min$, with the constant 2 corresponding to the time interval between beats for a heart rate of 30 bpm.

Finding the principal frequency can be challenging for datasets exhibiting multiple periodic behaviors or noise. To address this, the algorithm proceeds by detecting the peaks within the defined frequency interval, f_{intmin} and f_{intmax}, providing a selective approach to identify the correct peak. The peak with the highest amplitude and the first peak encountered is selected. If they correspond to the same frequency, that frequency is considered the principal frequency, f_{int}. Otherwise, the frequency value associated with the peak of the highest amplitude is chosen. It is important to note that further improvement is required in the detection of the principal frequency as a potential area for future work.

Using the modified FOD algorithm, WFOD (Algorithm 2), the common period of the data within the window, t_{int}, can be determined. The key idea behind this algorithm, which enables its use as a feature, lies in the location of the second peak in the periodogram. If the data within the window is regular and devoid of secondary oscillations or artifacts, the peak can be easily detected. However, in the presence of anomalies or artifacts, the location of the peak may shift or become undetectable. As a result, the values of t_{int} can serve as indicators of various types of anomalies in the data.

Furthermore, the value of t_{int} can be utilized in the transformation of the data into a more basic and stationary form known as the WFOD transform. The transformed data is represented by a zero vector of length equal to the window size ($WindowSize$) and single-point impulse peaks. These impulse peaks have an amplitude equal to the global extreme value (either minimum or maximum) that has the highest absolute amplitude value within the data window. The impulses are positioned equidistantly from each other with an interval of t_{int}. Consequently, the data window can be succinctly described by the common periodicity and the impulses. Hence, when the data in the window are regular and devoid of secondary oscillations or artifacts, the peak can be easily detected. Conversely, in the presence of anomalies or artifacts, the location of the peak may change or become undetectable. Such cases indicate the presence of anomalies in the current time window being iterated by the algorithm. Therefore, the values of t_{int} can serve as a preliminary indicator of various types of anomalies. The pseudo-code for the WFOD transformation is presented in Algorithm 3.

Algorithm 3 WFOD Transformation

```
Obtain the data window with length $WindowSize$.
Retrieve $t_{int}$ from Algorithm 2 and Equation 3.12.
Create a zero vector of length $WindowSize$.
Determine the global extremum value $GE_{amplitude}$ and its location $GE_{location}$ within the data window.
Place impulses of amplitude $GE_{amplitude}$ at the location $GE_{location}$ by shifting the location backwards and forwards by $t_{int}$ on the zero vector. This forms the WFOD transformed data window.
```

Figure 3.2 depicts the application of the WFOD transformation on real data [325]. The original data is represented by the blue line, while the red line represents its corresponding WFOD transformation. Notably, the figure displays equidistant impulses with a distance of t_{int} and an amplitude of $GE_{amplitude}$.

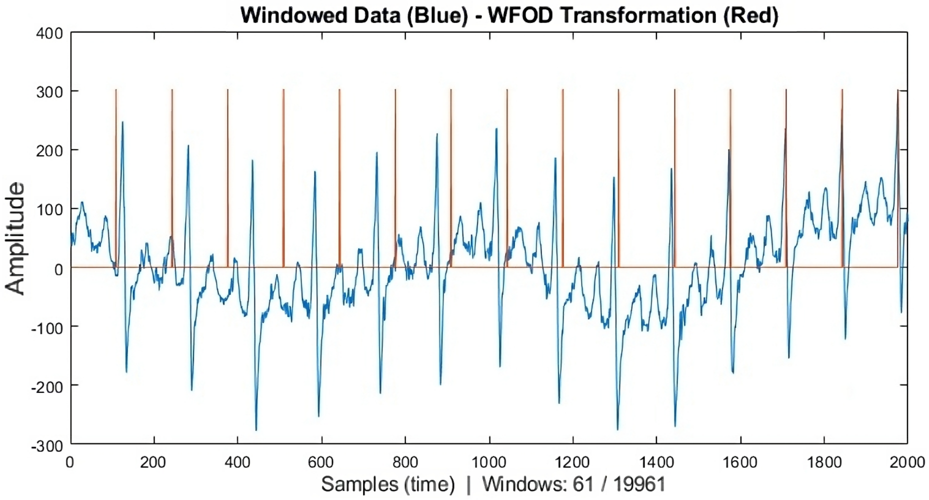


Figure 3.2: A representation of the windowed data (Blue) and their WFOD Transform (Red) on real ECG data from BIDMC Congestive Heart Failure database

The transformed data can be leveraged to extract additional features that capture properties in both the frequency and time domains. By implementing the WFOD transformation using the information of t_{int} and Algorithm 3, a new feature family can be derived based on the average fundamental period which is determined by the value of t_{int}, computed using Algorithm 2 and Equation 3.12. Moreover, the WFOD real-time single-window data transformation features are obtained following the WFOD transformation of the windowed data using Algorithm 3, the dynamic time warping (DTW) algorithm (as described in Section 2.3.2) is applied between the windowed data and its WFOD-transformed counterpart. The purpose is to calculate the DTW value, which serves as an indicator of the distortion between the original data and its WFOD transformation for the respective window. To illustrate, the DTW value is computed between the data represented by the blue line and its WFOD transform shown in the red line in Figure 3.2.

Therefore, by applying the WFOD transformation, a new feature represented by the DTW value can be obtained, encapsulating both the time and frequency domain characteristics of the data.

Figure 3.3 showcases a real-time analysis of the WFOD transform and its associated features, namely ’WFOD periodicity’ and ’DTW between the data and their WFOD transformation’. The figure demonstrates the windowing process applied to real ECG data from the BIDMC Congestive Heart Failure Database [325].

In the figure, the ECG data is divided into consecutive windows, as indicated by the vertical dashed lines. For each window, the WFOD transformation is performed, resulting in the red line representing the transformed data. The equidistant impulses within the transformed data correspond to the common periodicity captured by the WFOD periodicity feature.

Additionally, the DTW algorithm is employed to calculate the DTW value between the original data (represented by the blue line) and its WFOD transformation (represented by the red line). The DTW value serves as a feature that quantifies the distortion or dissimilarity between the two data representations. This real-time analysis demonstrates how the WFOD transform and its associated features can be applied to capture and analyze the characteristics of the ECG data.

3.3.1 Evaluation of the WFOD Feature Performances and Results

The performance of this new feature is evaluated in a classification study using an open-access ECG motion artifact dataset. The feature is constructed by pairing sta-

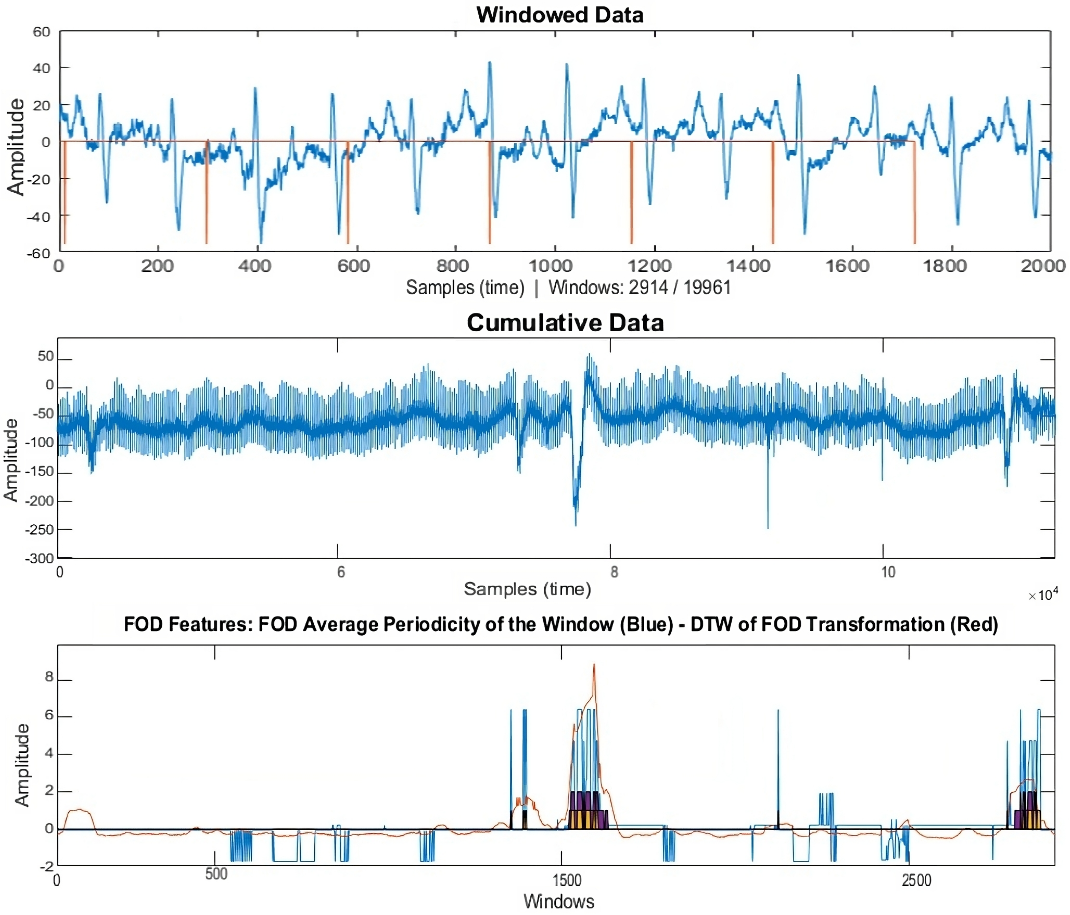


Figure 3.3: A representation of the windowing process and the WFOD features on real ECG data from BIDMC Congestive Heart Failure database.

tistical moments (mean, variance, skewness, and kurtosis) with each other, both with and without the WFOD feature. Four classifiers, namely the tree classifier, linear discrimination analysis (LDA), K-nearest neighbor (K-NN), and Naive Bayes classifier, are employed to independently classify the features. The results are compared, and the discrimination performance of the WFOD feature for different cases of ECG data is investigated.

The DTW dissimilarity metric exhibited subpar performance in discriminating synthetic ECG data, necessitating the testing of the WFOD algorithm, which incorporates DTW within its algorithms. In this phase of the study, the effectiveness of the new feature is evaluated using an open-access ECG motion artifact dataset in a classification study. Statistical moments (mean, variance, skewness, and kurtosis) are combined both with and without the WFOD feature to assess their impact. Four classifiers (tree classifier, linear discrimination analysis (LDA), K-nearest neighbor (K-NN), and Naive Bayes classifier) independently classify the features. The WFOD feature is included as an additional feature in half of the experimental conditions to observe its effects on classification performance. Due to the curse of dimensionality [326], the number of features is limited to a maximum of three, leading to the use of pairs of statistical moments in the experiments. The results are compared, and the discriminatory performance of the WFOD feature for different cases of ECG data is investigated.

To evaluate the classification performance of the proposed WFOD feature, a publicly available motion artifact-contaminated ECG dataset with three conditions [327] is utilized. This dataset, obtained from the Physionet website [328], includes ECG data collected from three subjects while they were in the standing, walking, and jumping conditions, allowing the observation of motion artifact contamination. The standing condition serves as the control, the walking condition represents the low anomaly condition, and the jumping condition reflects the high anomaly condition. The data has a sampling rate of 500Hz, with each case consisting of 8 seconds of measurements. Three offsets of electrode patches (0, 45, and 90 degrees) are considered as repetitive measures of the same subject in this study. Experimental conditions for classification are generated in pairs, such as standing-walking, standing-jumping, and walking-jumping cases, and classifications are independently performed on these pairs. The dataset prepared for the classification part comprises 18 instances of measurement, with 9 measurements per group in binary classification. Figure 3.4 provides a visualization of the standing-jumping condition pairs in a concatenated dataset from the motion artifact-contaminated ECG dataset [327], where the red markers indicate the jumping intervals. The stimuli are applied at the 4000th, 12000th, and 20000th samples.

The provided tables below present the classification accuracy results for different feature pairs and classifiers in three pairs of experiments. Here, Table 3.5 shows the results for the standing-walking experiment, Table 3.6 presents the results for the standing-jumping experiment, and Table 3.7 displays the results for the walking-jumping experiment.

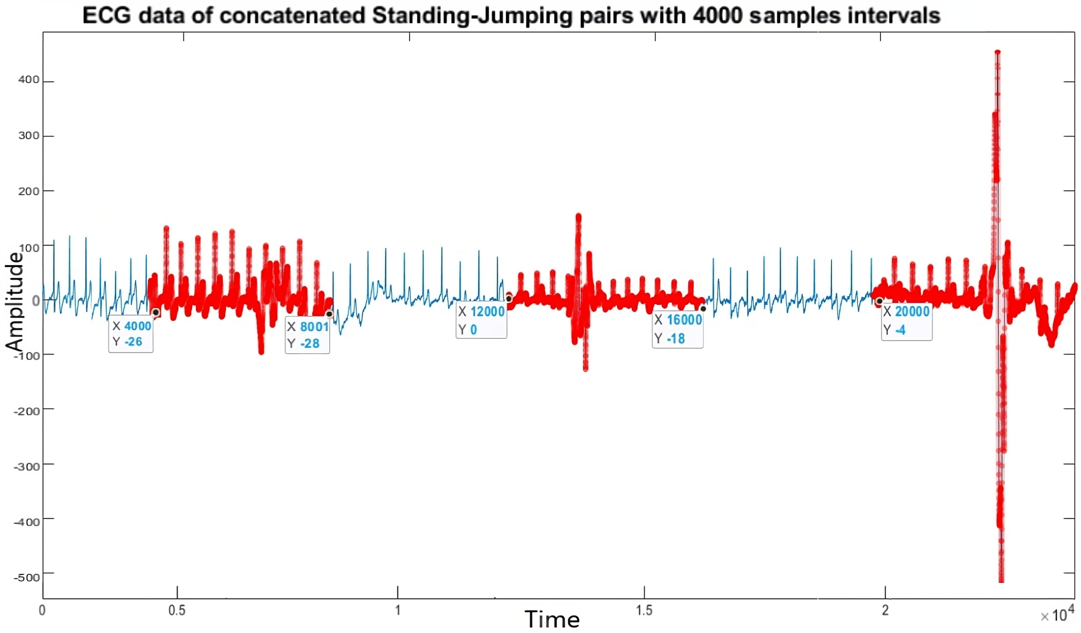


Figure 3.4: Visualization of standing-jumping condition pairs in a concatenated dataset from the motion artifact-contaminated ECG dataset.

In all analyses, the well-known classification approaches, namely, the tree algorithm, linear discriminant analysis (LDA), K-nearest neighbor approach (K-NN), and naive Bayes methods, are implemented while comparing the performance of different feature pairs.

Table 3.5 indicates that the inclusion of the WFOD feature does not have a significant impact on the classification accuracies of the classifiers. The WFOD feature either slightly decreases or increases the accuracies by 1% to 6%, without a consistent pattern. This suggests that the WFOD feature does not significantly contribute to discriminating the control and low anomaly groups. Additionally, the tree classifier generally outperforms the other classifiers in terms of classification accuracy.

Moving to Table 3.6, which focuses on the comparison between the control and high anomaly groups, it can be observed that the WFOD feature mostly leads to an improvement in classification accuracy ranging from 0% up to 14%. However, there are a few cases where the WFOD feature slightly decreases the accuracy by up to 4%. Overall, the tree classifier demonstrates better performance compared to the other classifiers for this pair of groups.

Table 3.5: Classification accuracy of statistical feature pairs with and without WFOD feature for standing - walking experiment under distinct classification methods.

$Feature\ Pairs$	$Tree$	LDA	$K-NN$	$Naive\ Bayes$
Mean + Variance	0.58	0.60	0.52	0.48
Mean + Variance + WFOD	0.57	0.61	0.51	0.49
Mean + Skewness	0.67	0.48	0.61	0.51
Mean + Skewness + WFOD	0.66	0.48	0.63	0.51
Mean + Kurtosis	0.73	0.53	0.56	0.49
Mean + Kurtosis + WFOD	0.67	0.52	0.62	0.50
Variance + Skewness	0.67	0.52	0.61	0.50
Variance + Skewness + WFOD	0.64	0.51	0.60	0.51
Variance + Kurtosis	0.73	0.58	0.56	0.53
Variance + Kurtosis + WFOD	0.72	0.56	0.63	0.47
Skewness + Kurtosis	0.70	0.52	0.67	0.60
Skewness + Kurtosis + WFOD	0.67	0.51	0.64	0.53
WFOD	0.66	0.51	0.46	0.50

Finally, Table 3.7 presents the results for discriminating the low and high anomaly groups. In this case, the WFOD feature shows superior performance, with accuracy improvements of up to 25% in differentiating the groups. The WFOD feature performs exceptionally well in this pair of groups. Similar to previous experiments, the tree classifier achieves the highest accuracy compared to the other classifiers.

3.3.2 Conclusion and Problems of the WFOD Algorithm

In this study, a new hybrid feature called the WFOD feature is proposed, which combines the time and frequency domain characteristics of data. The performance of this feature is evaluated in comparison to statistical moments as features in a classification study. The results demonstrate that the WFOD feature generally enhances the classification accuracies of ECG data with different anomaly groups. Particularly,

Table 3.6: Classification accuracy of statistical feature pairs with and without WFOD feature for standing - jumping experiment under distinct classification methods.

$Feature\ Pairs$	$Tree$	LDA	$K-NN$	$Naive\ Bayes$
Mean + Variance	0.75	0.57	0.73	0.56
Mean + Variance + WFOD	0.84	0.72	0.68	0.62
Mean + Skewness	0.87	0.73	0.71	0.68
Mean + Skewness + WFOD	0.88	0.80	0.76	0.66
Mean + Kurtosis	0.87	0.66	0.61	0.58
Mean + Kurtosis + WFOD	0.88	0.80	0.70	0.62
Variance + Skewness	0.86	0.71	0.75	0.72
Variance + Skewness + WFOD	0.87	0.81	0.73	0.72
Variance + Kurtosis	0.87	0.66	0.68	0.62
Variance + Kurtosis + WFOD	0.83	0.78	0.73	0.67
Skewness + Kurtosis	0.86	0.74	0.76	0.76
Skewness + Kurtosis + WFOD	0.89	0.80	0.75	0.76
WFOD	0.81	0.66	0.67	0.66

the WFOD feature shows the highest impact on improving the classification accuracy for distinguishing low and high anomaly groups, achieving an improvement of up to 25%. Furthermore, when used in unimodal classification, the WFOD feature achieves a good classification rate of up to 87% for the low and high anomaly groups.

To gain a better understanding of the behavior of individual features, it is proposed to conduct a dedicated study step to investigate each feature separately. Additionally, employing dimension reduction techniques such as principal component analysis or correlation-based analyses before further analysis would be beneficial in reducing computational time, especially for large or multivariate datasets. Moreover, the comparison of classifiers reveals that the tree classifier consistently achieves the best accuracy values among the tested classifiers. Therefore, in future extensions of this study, particularly in the analysis of biomedical data, the tree classifier will be used as the primary classifier to identify the optimal combination of classifiers and fea-

Table 3.7: Classification accuracy of statistical feature pairs with and without WFOD feature for walking - jumping experiment under distinct classification methods.

$Feature\ Pairs$	$Tree$	LDA	$K-NN$	$Naive\ Bayes$
Mean + Variance	0.78	0.66	0.65	0.53
Mean + Variance + WFOD	0.90	0.71	0.67	0.63
Mean + Skewness	0.77	0.69	0.69	0.66
Mean + Skewness + WFOD	0.90	0.65	0.73	0.71
Mean + Kurtosis	0.84	0.70	0.69	0.65
Mean + Kurtosis + WFOD	0.89	0.66	0.73	0.67
Variance + Skewness	0.77	0.67	0.67	0.66
Variance + Skewness + WFOD	0.92	0.67	0.69	0.66
Variance + Kurtosis	0.81	0.69	0.59	0.66
Variance + Kurtosis + WFOD	0.93	0.68	0.69	0.63
Skewness + Kurtosis	0.68	0.71	0.65	0.71
Skewness + Kurtosis + WFOD	0.93	0.69	0.70	0.70
WFOD	0.87	0.62	0.76	0.60

ture extraction methods. Based on these promising results, it is believed that further modifications to the WFOD algorithm could potentially enhance the performance of the WFOD feature in classifying time series data. Additionally, in addition to the WFOD periodicity and DTW value, the plan is to incorporate additional features into the WFOD feature family to further enhance its classification capabilities.

However, a problem arises with the grand averaging of the WFOD algorithm, as it does not align the grand average patterns for the regular data behavior. Therefore, the anomalies which have different patterns than the baseline data become hard to detect unless the window size and slide size parameters are well-adjusted in a moving windows approach. The definition and the solution to this grand average alignment problem are defined in Section 3.3.3.

3.3.3 Problem with the WFOD Algorithm: Grand Average Alignment

The data intervals are shifted throughout the entire dataset, represented by the red dashed vertical lines in the upper graph of Figure 4.4. Each of these intervals forms a data structure, which tends to exhibit similarities in quasi-periodic datasets. Therefore, all of these data structures are grand averaged to obtain a common shape for the data, as illustrated in Figure 3.5 for the current example.

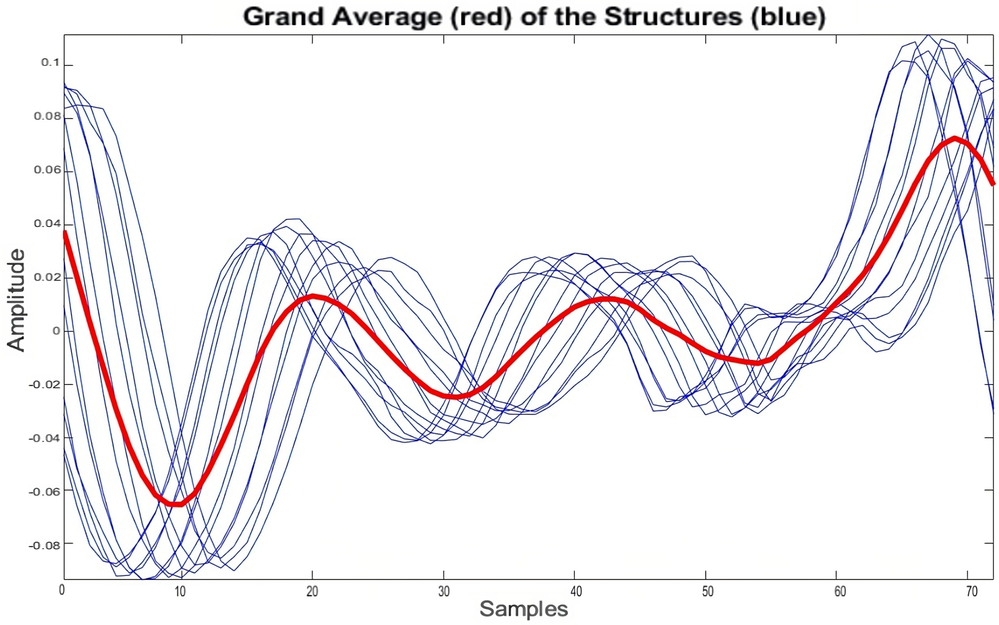


Figure 3.5: Computation of the grand average by cropping the structures as shown in Figure 4.4.

However, aligning the windows in this manner can distort the data behavior, especially when using a higher number of windows. Therefore, a better alignment approach is needed. To address this, a basic solution is employed, aligning the windows based on their highest amplitude sample. In this approach, the maximum value within each window is identified. A temporary window is then formed by centering the maximum value and shifting the window boundaries accordingly. This alignment method effectively resolves alignment issues for quasi-periodic datasets. Figure 3.6 and Figure 3.7 provides two examples of applying this approach to synthetic quasi-periodic data. Additionally, the maximum-shift aligned version of the data from Figure 3.5 is utilized in the latest version of the proposed algorithm, as depicted in Figure 4.5.

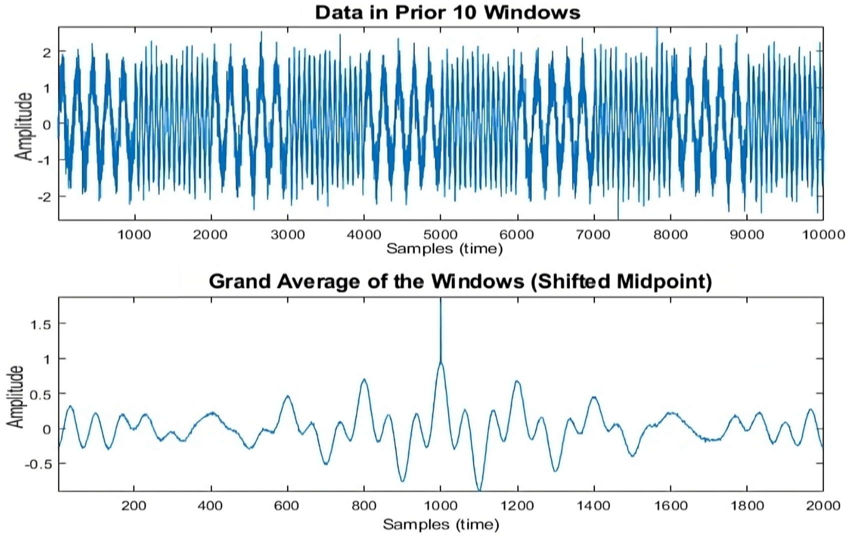


Figure 3.6: Grand average pattern for quasi-periodic data consists of two epochs with the same amplitudes and different periods.

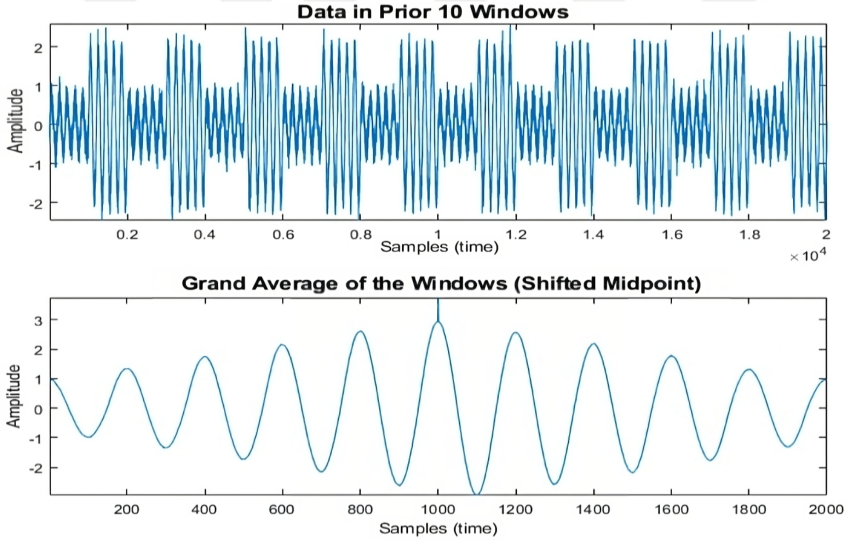


Figure 3.7: Grand average pattern for quasi-periodic data consists of two epochs with the same periods and different amplitudes.

An alternative approach to this partitioning step is to locate the global maximum value in the data, which appears at the 316th sample in the upper graph of Figure 4.5. This global extreme point is then centered within a data interval of length t_{int}, and the algorithm proceeds. Once the grand average of the structures is computed and each structure is identified, the algorithm advances to the extraction of dissimilarity features.

CHAPTER 4

THE PROPOSED APPROACH

A novel algorithm is proposed for the analysis of quasi-periodic time series data, and it is an extension of the FOD [44], the WFOD [140], and the PAD [21] algorithms. Moreover, there are also several extension studies regarding the usage and performances of the various components of the proposed approach such as the comparison of the dissimilarity metrics as features which can be found in Section A.1.1, and a comparison of the pitch frequency estimation algorithms that can be found in Section A.1.2.

The proposed algorithm incorporates various fundamental approaches that utilize pitch frequency in the calculations. The algorithm can be applied to the entire dataset or to sliding windows. However, the sliding windows approach is preferred due to its computational efficiency and potential for real-time implementation. Currently, the proposed algorithm is implemented in MATLAB, which limits its computational performance and the selection of sub-task algorithms to MATLAB references. A future plan for the approach involves converting the algorithm into the Python domain and creating a library to make it open source. This would enable wider accessibility and usage of the proposed algorithm. The flowchart of the proposed algorithm is presented in Figure 4.1, while a pseudo-algorithm is provided in Algorithm 1.

4.1 Initialization of the Sliding Window Approach

The proposed algorithm is designed to operate in sliding windows with a defined window size and step size. The window size is estimated to cover at least some (say 5+) periods of data oscillation to capture the pitch frequency effectively. Sliding

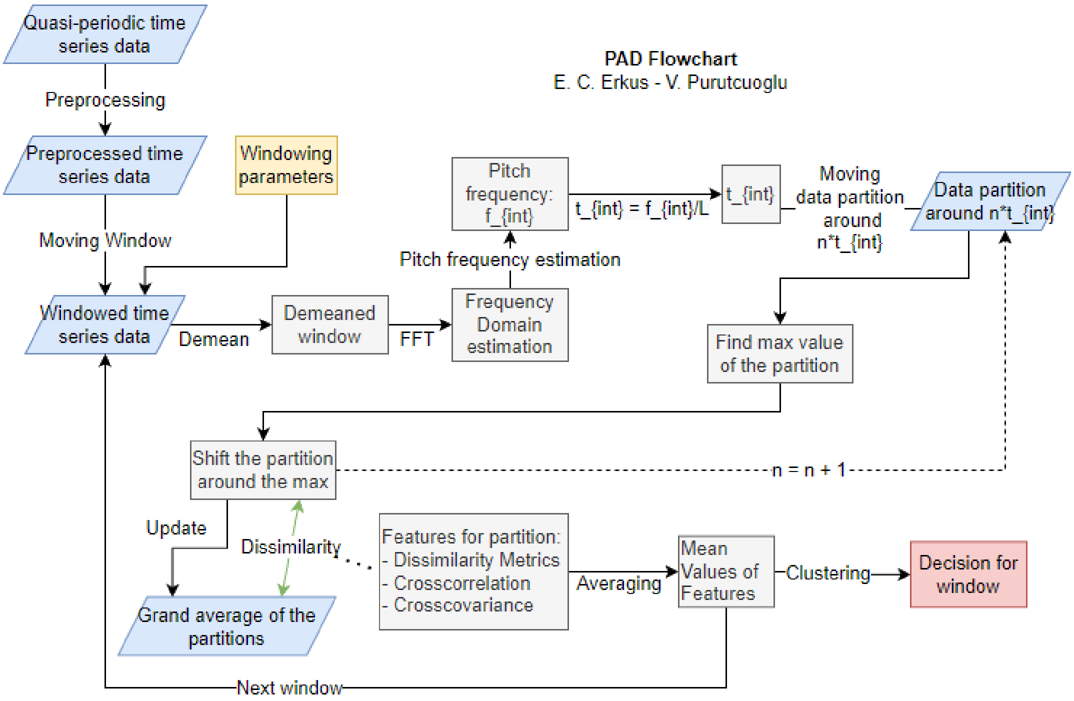

PAD Flowchart
E. C. Erkus - V. Purutcuoglu
Quasi-periodic time series data
Preprocessing
Preprocessed time series data
Windowing parameters
Moving Window
Windowed time series data
Demean
Demeaned window
FFT
Frequency Domain estimation
Pitch frequency estimation
Pitch frequency: f_{int}
t_{int} = f_{int}/L
t_{int}
Moving data partition around n*t_{int}
Data partition around n*t_{int}
Find max value of the partition
Shift the partition around the max
n = n + 1
Update
Dissimilarity
Grand average of the partitions
Features for partition:
- Dissimilarity Metrics
- Crosscorrelation
- Crosscovariance
Averaging
Mean Values of Features
Clustering
Decision for window
Next window

Figure 4.1: Flowchart illustrating the proposed approach.

windows are used to process the data, where the window length and slide distance are automatically determined by the algorithm. In real-time reading, the step size should be set to ensure timely processing and avoid any lag. However, if the data is inputted in intervals with sufficient length, the window length may match the data length, and the step size becomes irrelevant.

A challenge arises when the data length is not suitable to fit the window and shift size for complete scanning of the data. To address this, a solution has been implemented by gradually increasing the pre-auto-determined window size until the expression in Equation 4.1 becomes an integer.

$$\frac{DataLength - WindowSize}{ShiftSize} \tag{4.1}$$

By making a slight adjustment to the window length, the sliding windows can be aligned with the data. Within each window, feature extraction and evaluation steps take place, which are further explained in the subsequent subsections.

In real-time applications, ML models are typically unsupervised. To introduce supervision, a dynamic learning approach can be implemented, which is particularly useful for subject-specific diagnosis in biomedical applications where unique patterns may exist for each subject.

The sliding windows method is used to simulate the real-time processing of time series data. While the window and parameter settings should ideally match the real-time specifications of the device, this study omits such limitations. To mimic real-time operation, parameters such as the window size and slide size need to be defined. These parameters are illustrated in Figure 4.2.

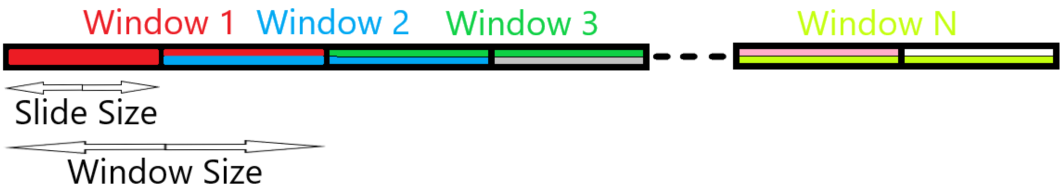


Figure 4.2: Illustration of window size and slide size.

The analysis of preprocessed data begins by defining the window and slide sizes for the sliding windows approach. Within each sliding window, the first step is the demeaning operation. This involves subtracting the mean value of the samples within a window from each individual sample. The purpose of this demeaning operation is to ensure comparability between future and past windows in the iterative sliding window process. By considering the trending behavior of the data over time, this step helps maintain consistency in the analysis. Standardization is not employed since it would further reduce the range of all samples, including potential outliers. The proposed algorithm relies on anomalies' statistical and spectral characteristics, and their variances may not be homogeneous. In fact, anomalies often lead to increased variances, providing valuable information for the proposed algorithm to identify anomalies by leveraging the spectral domain and the resulting changes in pitch frequency estimation performance.

4.2 The Pitch Frequency Estimation

The next step is to compute the frequency domain estimation of the windowed data using the Fast Fourier Transform algorithm [329]. The frequency-domain estimation of the quasi-periodic data behavior often results in a peak at the fundamental frequency. The finding of such a peak in the frequency domain in audio signal processing is often called "pitch frequency estimation". There are several other methods for estimating pitch frequency such as Normalized Correlation Function (NCF) [204], Log-Harmonic Summation (LHS) [214], Summation of Residual Harmonics (SRH) [218], and Pitch Estimation Filter (PEF) [207]. Their basic technical background and algorithmic steps can be found in Section 2.2.2. However, thanks to the faster computational speed of the Cepstrum Pitch Determination (CEP) algorithm [210], and the fact that it provides slightly better classification metrics compared to others, according to the extension studies regarding the pitch frequency estimation algorithms as some can be found in Section A.1.2, the proposed algorithm adopts the CEP algorithm for real-data analyses, and NCF or PEF for the synthetic data with specific purposes.

The selected pitch frequency estimation algorithm is expected to yield the pitch frequency value, f_0, within the operating window, by following the respective algorithmic steps as can be found in Section 2.2.2, and the next steps of the proposed algorithm proceed.

4.3 The Relationship Between Frequency and Time Domain

The time and frequency domains exhibit an inverse relationship with respect to sample and frequency indices. This relationship allows us to convert the estimated pitch frequency, denoted as f_0, to its corresponding time domain counterpart, referred to as t_{int}, using Equation 4.2:

$$t_{int} = \frac{f_s}{f_0}. \tag{4.2}$$

Here, t_{int} represents the average oscillation period, which corresponds to the average distance between consecutive quasi-periodic peaks in the time domain data. It is inversely related to the pitch frequency, f_0, and scaled by the sampling frequency, f_s. An illustrative example of the relationship between f_0 and t_{int} values can be observed in Figure 4.3. In this example, the signal has a sampling rate of 16000Hz, and the pitch frequency is marked as the first harmonic peak in the frequency domain at 225.684.

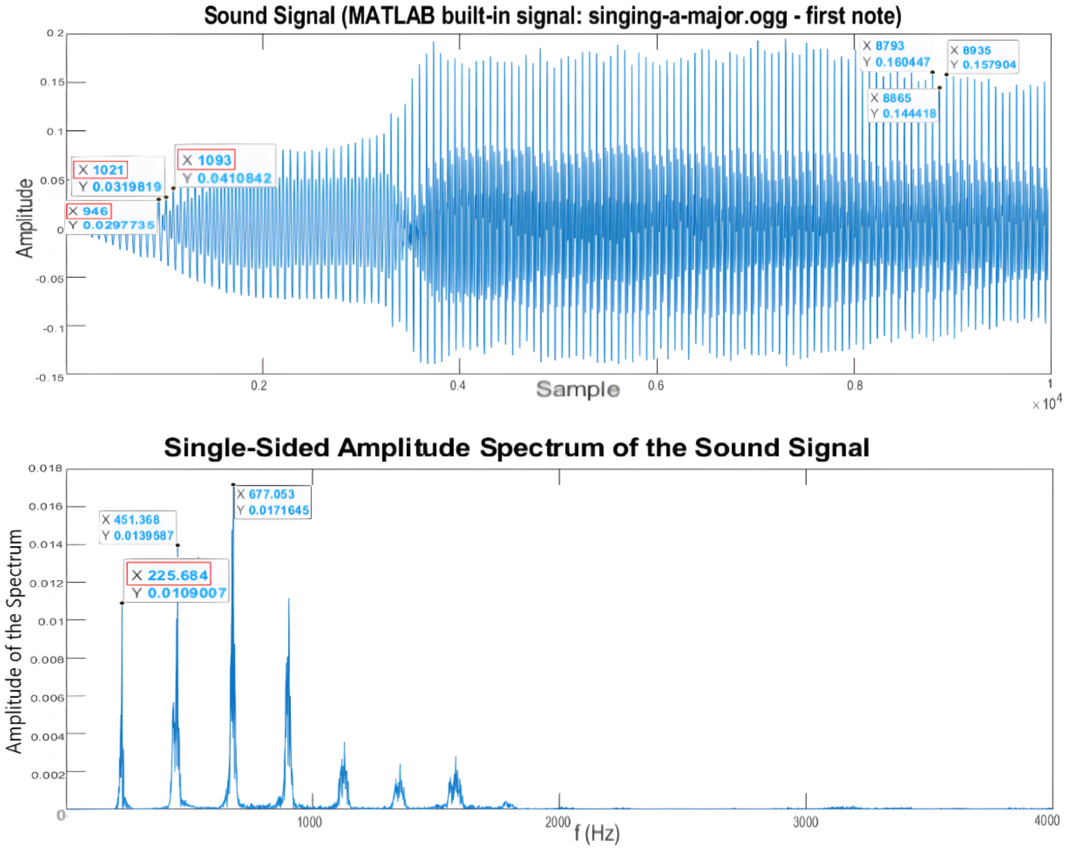


Figure 4.3: The relationship between f_0 and t_{int} values.

The upper graph in Figure 4.3 corresponds to a signal that can be accessed by loading the "singing-a-major.ogg" file in MATLAB (R2021b) and cropping the first cluster of data. In this case, the distance between subsequent peaks in the time domain fluctuates around the value of 71. This value approximately represents the average period of the quasi-periodic oscillation and can be computed by subtracting the consecutive peak location values, such as 1093, 1021, and 946, as indicated by the marked samples in the upper sub-figure of Figure 4.3. Furthermore, the value 71 corresponds to the rounded integer value of $16000/225.684$, where 16000 is the sampling rate in this example, and 225.684 denotes the location of the first peak in the frequency domain, as shown in the lower sub-figure of Figure 4.3.

It is hypothesized that a collective anomaly may alter this average oscillation period, either increasing or decreasing its value or even causing the oscillation to disappear from the recorded data for a certain interval. Such anomalies can arise due to statistical changes in the time domain that impact the frequency domain as well. For instance, during the measurement of quasi-periodic data, an electrode disconnection interval often introduces random noise, which obstructs the regular data pattern from emerging during that duration. As a result, the fundamental peaks in the frequency domain may flatten or entirely merge with other smaller peaks. Consequently, the main principle of the proposed algorithm lies in identifying abrupt changes in the main oscillation frequency over time. Various types of features are proposed and tested to measure such changes in the time domain, with the objective of observing differences in these features between resting and anomalous conditions of the data.

To provide an alternative perspective and proceed to the next step of the algorithm, a segment of the audio data from Figure 4.3 is extracted, spanning the 14000th to 15000th samples, and the proposed algorithm is applied. The t_{int} value is calculated using Equation 4.2.

4.4 Data Partition and Grand Averaging

The subsequent step involves dividing the windowed data into smaller segments of length t_{int}, represented by the red dashed vertical lines in Figure 4.4.

Once the windowed data has been partitioned as shown in Figure 4.4, each partition is independently shifted around its maximum value while maintaining a fixed length of t_{int} throughout the windowed data. This shifting operation may result in overlaps or unused samples, but these areas are most susceptible to the presence of anomalies, making them the focus of anomaly detection in the proposed algorithm. In essence, each of these shifted partitions represents a similar segment in the absence of anomalous behavior within the quasi-periodic data. Consequently, for the current example, computing the grand average of these partitions yields a common pattern across the

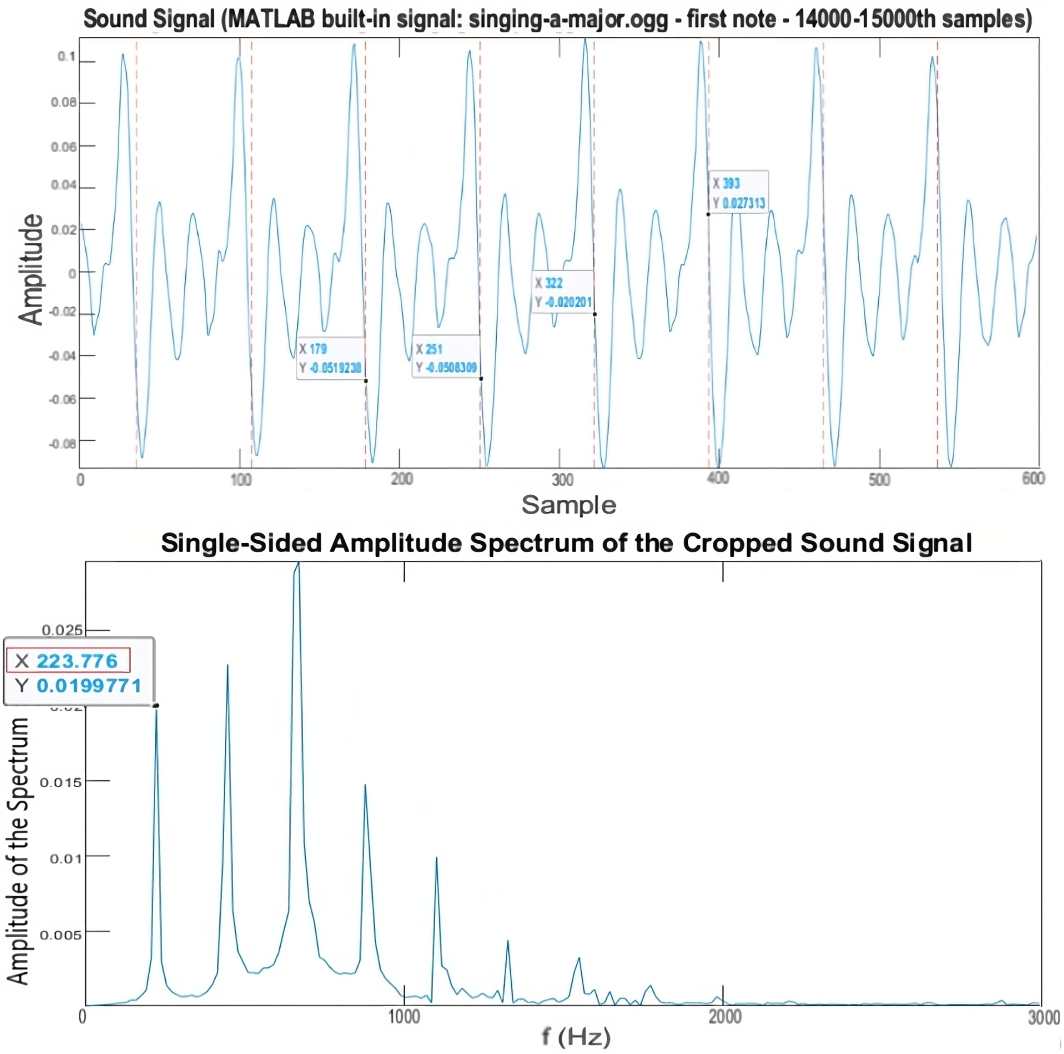


Figure 4.4: Formation of initial sequential partitions with a length of t_{int}.

data, as depicted in Figure 4.5.

During the progression of the algorithm, in each windowed data, each partition contributes to the updating of the grand average, allowing for a comprehensive analysis of the data from start to end. This updating process enables the algorithm to learn subject-specific partitioning, leading to a personalized approach to anomaly detection. The update is performed using weighted sample-wise averaging, where the weights assigned to the new partition are 1, and the weight assigned to the previous sample-wise grand average is $g-1$, where g represents the total number of partitions involved in the sample-wise grand average thus far. Mathematically, the update of the grand average $G[i]$ with its previous version $G[i-1]$ and the partition $P[i]$ is

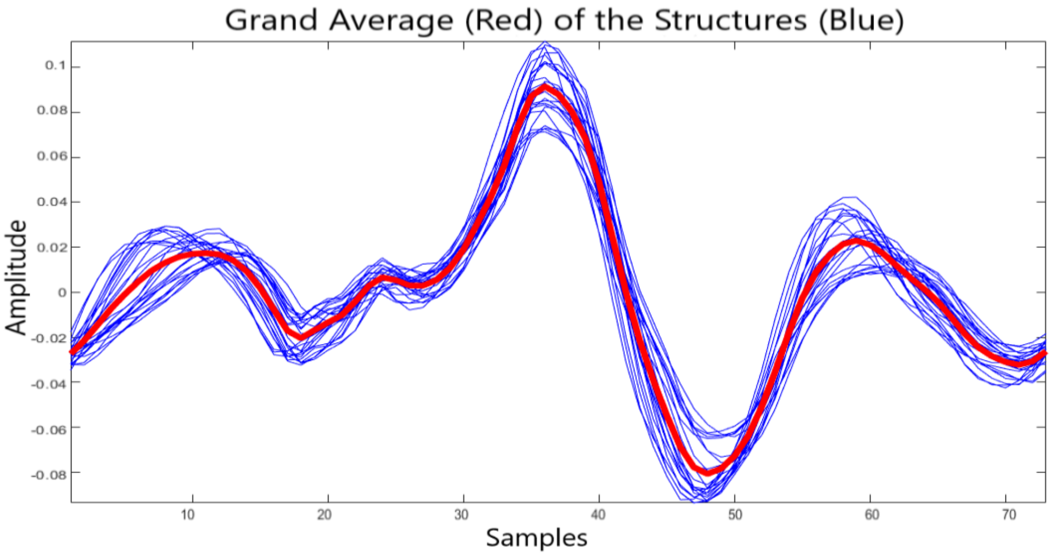


Figure 4.5: Computation of the grand average of partitions obtained from Figure 4.4.

expressed by Equation 4.3:

$$G[i] = \frac{(g-1) * G[i-1] + P[i]}{g}. \tag{4.3}$$

It should be noted that the grand average update operation, as described in Equation 4.3, is performed for each iterated window and partition $P[i]$ within a window. Here, i denotes the current cumulative number of partitions. In the proposed algorithm, prior to updating the grand average of the partitions ($G[i]$), the dissimilarity and distance metrics are employed to compare the current partition ($P[i]$) with the previous grand average ($G[i-1]$).

4.5 Dissimilarity Computation as Feature Extraction

Once the grand average of the partitions is computed and each partition is separated within an iterated data window, the algorithm proceeds to extract dissimilarity features. Dissimilarity metrics are employed to measure the distance between two different time series by evaluating the distances between individual samples and deriving an overall score for the entire dataset. Generally, dissimilarity metric values increase as the two time series differ, often in a statistical sense. Therefore, higher dissimilarity metric values are expected in anomalous data intervals. When comparing data with distinct behaviors, each dissimilarity metric has its own advantages and disadvantages. The current analyses utilize the most commonly used dissimilarity metrics listed in Table 4.1.

Table 4.1: The dissimilarity metrics applied in the proposed algorithm and their formulations.

Dissimilarity Metric	Formula
Euclidean	$\sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$
Square Euclidean	$(x_1 - y_1)^2 + (x_2 - y_2)^2$
City Block	$\mid x_1 - y_1 \mid + \mid x_2 - y_2 \mid$
Chebyshev	$max(\mid x_1 - y_1 \mid, \mid x_2 - y_2 \mid)$

Time-dependent similarity functions, such as cross-correlation and cross-covariance, are incorporated in addition to the dissimilarity metrics used in the multivariate classification process. Thus, the application of these functions is chosen under the assumption of wide-sense stationarity (WSS) of time series data, as shown in Table 4.2.

Table 4.2: The formulations of the time-dependent similarity metrics applied in the proposed algorithm under the WSS assumption.

Time-dependent Similarity Metric	Formula
Cross-correlation (WSS assumed)	$R_{XY}(t_1, t_2) = E[X_{t_1}\overline{Y_{t_2}}]$
Cross-covariance (WSS assumed)	$K_{XY}(t_1, t_2) = E[(X_{t_1} - \mu_X(t_1))\overline{(Y_{t_2} - \mu_Y(t_2))}]$

During the sliding window iteration of the proposed algorithm, all the dissimilarity measures from Table 4.1 and Table 4.2 are independently averaged across the partitions, as explained in Section 4.4, to enable the comparison of windows with different partition sizes. The underlying assumption is that anomalous windows have a distinct number of partitions compared to normal windows due to the computed t_{int} and a varying spread of averaged values. Once the partitions within a window are averaged, the resulting averaged dissimilarity measure features are utilized in the clustering process to make a decision about the current window.

The four most commonly used dissimilarity metrics for the Minkowski distance generalization are selected based on the preliminary work and empirical results obtained from toy data, as presented in Table 4.3. It has been observed that improved supervised classification performance is achieved by the multivariate usage of certain combinations of dissimilarity metrics, although this comes at the expense of a slight increase in computational cost [330]. The toy dataset is synthetically generated, with the anomalous data interval being covered by 5% of the subject data, generated from a Gaussian Normal distribution and sequentially located in the middle of the data. The baseline data is generated using a sinusoidal waveform with an amplitude of 1, a period of 1000 samples, and an oscillation around 0. This results in a total of 100,000 samples for each of the 100 synthetic subjects, with additional noise under a signal-to-noise ratio of 10dB. The preliminary proposed algorithm is applied to the generated toy dataset, and F1-scores are obtained for each of the dissimilarity metrics applied individually. The differences in F1-score values are expected to arise from the representation of various aspects of the data in a multidimensional clustering approach.

Table 4.3: Preliminary F1-score values of the proposed algorithm for different univariate dissimilarity metrics, obtained using the toy data.

Feature	Average F1-score
Euclidean	0.80
Square Euclidean	0.65
City Block	0.77
Chebyshev	0.82
Multivariate	0.84

Based on the preliminary F1-score results shown in Table 4.3 for the toy data, the decision is made to use the multivariate approach with dissimilarity metrics, despite the associated increase in computational time. Therefore, both the dissimilarity metrics and the time-dependent similarity functions specified in the proposed algorithm are employed in the multivariate clustering approach.

During each iteration of the sliding window process, when a new data partition is created, a set of feature values is extracted as a 1×6 vector, referred to as the *dis-*

similarity values vector. The number 6 represents the total number of features, which includes the dissimilarity and time-dependence features defined in Table 4.1 and Table 4.2 respectively. As the iteration progresses through the data partitions within a window, these vectors are stored for the corresponding window, forming a $p \times 6$ matrix called the *dissimilarity values* matrix. Here, p denotes the number of partitions in the window. If all data partitions in a window are considered, the *dissimilarity values* matrix is column-wise averaged, resulting in a vector of length 1×6 known as the *anomaly score* vector for that window. Since the sliding windows approach is utilized, the *anomaly score* vectors for each window are concatenated to form an *anomaly score* matrix of size $w \times 6$, where w represents the number of windows. Considering the slightly better performance observed using the multivariate feature classification approach, as indicated in the preliminary results in Table 4.3, each new *anomaly score* vector in the sliding windows approach is evaluated by the trained model using the *anomaly score* matrix containing the *anomaly score* vectors from all previously processed windows.

4.6 Clustering for the Decision of Anomaly

The iterative process involves creating a feature vector for each window, which is stored and referred to as an "anomaly score." These multivariate feature vectors are utilized for unsupervised binary classification of each window, enabling real-time anomaly detection using the proposed algorithm. The windows are clustered based on their predicted multivariate anomaly scores, determining whether they are anomalies or not.

To perform binary classification in each iteration of the sliding windows, the K-medoids clustering algorithm is adapted. Specifically, the Partitioning Around the Medoids (PAM) algorithm is employed, which is a greedy algorithm that evaluates all possible swaps between the medoids and the feature values to identify any potential decrease in distance values [331]. The distances between centroids and features are measured using the Euclidean distance.

After each iteration, the windows are re-clustered, and the smaller cluster is considered the anomaly group. This approach is based on the assumption that anomalies are sparsely distributed throughout the entire data. The proposed algorithm classifies the iterating windows as either anomalies or non-anomalies, assuming that there are fewer anomalous data windows than regular ones. The smaller cluster is chosen iteratively as the anomaly group by the proposed algorithm. As the number of data windows increases, this assumption becomes more valid due to the convergence of the data distribution to a Gaussian distribution, as predicted by the law of large numbers. In this case, the anomalous structures are expected to correspond to the tails of a Gaussian distribution.

4.7 The Proposed Algorithm Pseudocode

The overall process for finding anomalous windows and intervals in time series data is summarized in Algorithm 1. As previously mentioned, the resolution of samples detected by the proposed algorithm depends directly on the initialized window and slide sizes used in the calculation. However, a detailed analysis of the effects of window and slide sizes, as well as finding the optimal values for them, is left for future work.

It is worth noting that the specific choice of window size and slide size can affect the resolution and accuracy of the detected anomalies. Therefore, as a future work, an automatic determination of the optimized window and slide size is planned to be developed.

4.8 The Time Complexity

In order to observe and quantize the time complexity of the proposed approach, it's computational times are measured when it is applied to the data with different lengths. This part of the study involves both synthetical and real data analyses. The first part of the time complexity analysis is performed on the synthetically generated datasets, where the samples are randomly produced using the Gaussian Normal distribution with the mean value of 0, and unit standard deviation. The overall data are generated

Algorithm 4 The proposed algorithm

```
Input the window size and slide size, the total number of dissimilarity features as 6, $w = 1$, the anomaly score vector of size $w \times 6$, and initialize sliding windows:
 1. Initiate the $w^{th}$ windowed data.
 2. Utilize the selected pitch frequency estimation algorithm to find the pitch frequency, $f_0$.
 3. Convert $f_0$ into $t_{int}$ using Equation 4.2.
 4. Split the windowed data into $P$ sequential partitions, each with a length of $t_{int}$ and centered around the sample $p \times t_{int}$, where $p$ is the currently iterated partition number.
 5. Initialize $p = 1$ and the grand average vector with a length of $t_{int}$, and start data partitioning.
 6. Find the maximum value of the partition.
 7. Shift the partition, locating the maximum value as the middle sample.
 8. Compute the dissimilarity values vector with a size of $1 \times 6$ between the current data partition and the current grand average.
 9. Append the dissimilarity values vector to the dissimilarity values matrix with a size of $p \times 6$.
10. Update the grand average of the data partitions using Equation 4.3.
11. Increment $p$ by 1.
12. Proceed to Step 13 if $p = P$; otherwise, return to Step 6.
13. Compute the partition-wise mean of the dissimilarity values matrix to obtain the anomaly score vector of length $1 \times 6$ of the current window. .
14. Erase the dissimilarity values matrix.
15. Update the cumulative anomaly score matrix of size $w \times 6$.
16. Classify the current anomaly score vector as the test set and the current anomaly score matrix as the train set using the k-medoids clustering algorithm and record the label for the current window.
17. Increment $w$ by 1.
18. Stop the algorithm if $w = W$; otherwise, proceed to the next window, i.e., Step 1.
```

100 times under the fixed data lengths, i.e., 1000, 10000, 100000, and 1000000 simulating a Monte-Carlo experiment with 100 runs, but with different data in each run that are generated using the same probabilistic distribution. Here, the window size has been set to cover 5% of the overall data length for each of the conditions with different data lengths in each moving windows iteration with the same slide size that

is resulting in no overlapping windows. The results for the synthetic data experiment for the time complexity can be found in Table 4.4.

Table 4.4: The average computational times for randomly generated datasets consisting of 100 subjects in each with fixed length under Gaussian Normal Distribution.

Data Length	Computational Time (seconds)
1000	2.60
10000	18.01
100000	162.98
1000000	1467.04

Additional to the synthetic data experiment, the proposed approach is tested on the real, time series ECG data, namely MIT-BIH Malignant Ventricular Ectopy Dataset [332]. In this experiment, the window size is also set fixed to the overall data length to cover 10% of the data in each iteration. Hereby, the results of this real data experiment can be found in Table 4.5.

Table 4.5: The average computational times for real benchmark dataset MIT-BIH Malignant Ventricular Ectopy with cropped data lengths.

Data Length	Computational Time (seconds)
5250	9.91
10500	18.33
21000	35.32
52500	75.43
105000	179.64
210000	341.08
525000	839.35

Note that, both of these experiments are performed on the same computer with the processor specifications 11th Gen Intel(R) Core(TM) i7-11700KF @ 3.60GHz without using any accelerator or hardware boost. Before the simulations are run, all of the non-necessary user-based applications except the MATLAB 2021b, academic license are closed where the simulations are performed with. Considering the computational times as recorded by the built-in functions of the MATLAB software as given in Table 4.4 and Table 4.5, respectively for the synthetic and real data experiments, the change in the computational times for the tested data lengths is found that it is almost linearly changing by the data length. The slight decrease in the computational times is believed to originate from the data reading and process optimization by the MATLAB software. Hence, the proposed approach can be considered to possess a linear time complexity, i.e., o(n).

CHAPTER 5

APPLICATIONS, RESULTS AND DISCUSSION

This chapter presents a series of experiments conducted to assess the applicability of the proposed algorithm and compare its performance with other existing methods. The experiments are categorized into two sections: synthetic data analyses and real data analyses. The synthetic data analyses aim to simulate extreme data conditions to observe the behavior and performance of the proposed algorithm. Conversely, the real data analysis aims to demonstrate the effectiveness of the proposed algorithm in capturing regular data behavior and distinguishing faulty data intervals from the baseline data.

5.1 Synthetic Data Analyses

The synthetic data analysis part is divided into two analyses. The first one is conducted to observe the dissimilarity metric features that are used in the proposed algorithm and compare them with the other common features from several feature families, namely, statistical, spectral, and transformational. On the other hand, the second synthetic data analysis is conducted to observe the fundamental effects of the sliding window parameters, especially when the window size and the slide size match with the specific ratios with the baseline data periodicity. Hereby, in both of the analyses, the dataset generation, the experimental setup, and other methods to compare, and the results of the experiments are explained in different subsections.

5.1.1 Synthetic Data Analysis 1: Comparison of Feature Extraction Methods

This section is dedicated to observing the classification performances of the different basic features that are extracted from a variety of data conditions, and the results are used to further modify the proposed approach regarding its performance compared to those basic features.

5.1.1.1 Synthetic Dataset 1 Generation

The evaluation of the proposed algorithm's performance initially relies on testing with synthetically generated datasets. These datasets are crucial in assessing the algorithm's effectiveness under various data distributions and parameters for both anomaly and non-anomaly conditions. In this paper, eight different synthetically generated datasets are presented for performance evaluation.

Each dataset consists of 50 subject data, each containing 50 sequential anomaly intervals and 50 sequential non-anomaly intervals. The data intervals for each subject are generated using two independent probabilistic data distributions, each with its own set of parameters. Both anomaly and non-anomaly data intervals have equal sample sizes of 1000 samples and are sampled at a rate of 100Hz. Therefore, the total data length for each subject is 100,000 samples, comprising a combination of 50 data intervals with 1000 samples each. The selected distributions for the non-anomaly condition have relatively low variance and mean values, representing normal data behavior. Conversely, for the anomaly conditions, distributions with contrasting characteristics are chosen to simulate abnormal data behavior. Additionally, white noise with a signal-to-noise ratio (SNR) of 10dB is applied to each dataset to introduce further randomness.

Table 5.1 provides details of the eight synthetically generated datasets, including the data distributions used for each condition. These datasets are designed to observe the discriminability of the proposed algorithm in different scenarios, ranging from random data with high mean or variation to periodic data with high oscillation, amplitude, or both.

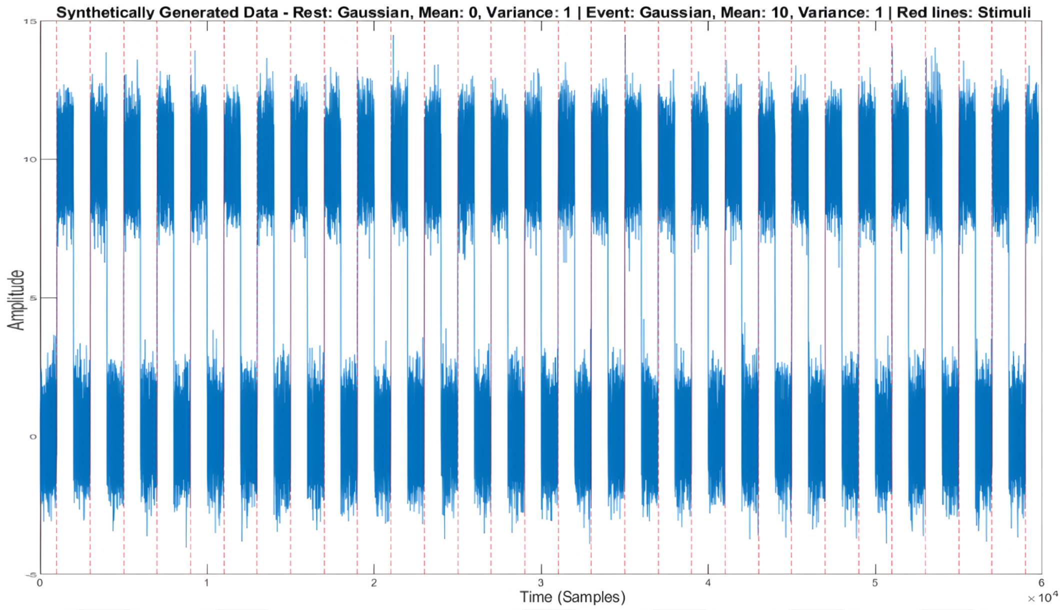
Table 5.1: The list of simulated datasets.

Rest Data Distribution	Event Data Distribution	Aim to Observe the Discriminability of
Gaussian, μ: 0, σ^2: 1.00	Gaussian, μ: 10, σ^2: 1.00	Random data with high mean
Gaussian, μ: 0, σ^2: 1.00	Gaussian, μ: 0, σ^2: 10	Random data with high variation
Gaussian, μ: 0, σ^2: 1.00	Exponential, Rate: 5	Random data with high variation
Gaussian, μ: 0, σ^2: 1.00	Student's t, ν: 5	Random data with high variation
Gaussian, μ: 0, σ^2: 1.00	Log-normal, μ: 0, σ^2: 5	Random data with high amp. outliers
Sinusoidal, Amp.: 1.00, F_s: 5	Sinusoidal, Amp.: 1.00, F_s: 10	Periodic data with high osc.
Sinusoidal, Amp.: 1.00, F_s: 5	Sinusoidal, Amp.: 2, F_s: 5	Periodic data with high amp.
Sinusoidal, Amp.: 1.00, F_s: 5	Sinusoidal, Amp.: 2, F_s: 10	Periodic data with high osc. and amp.

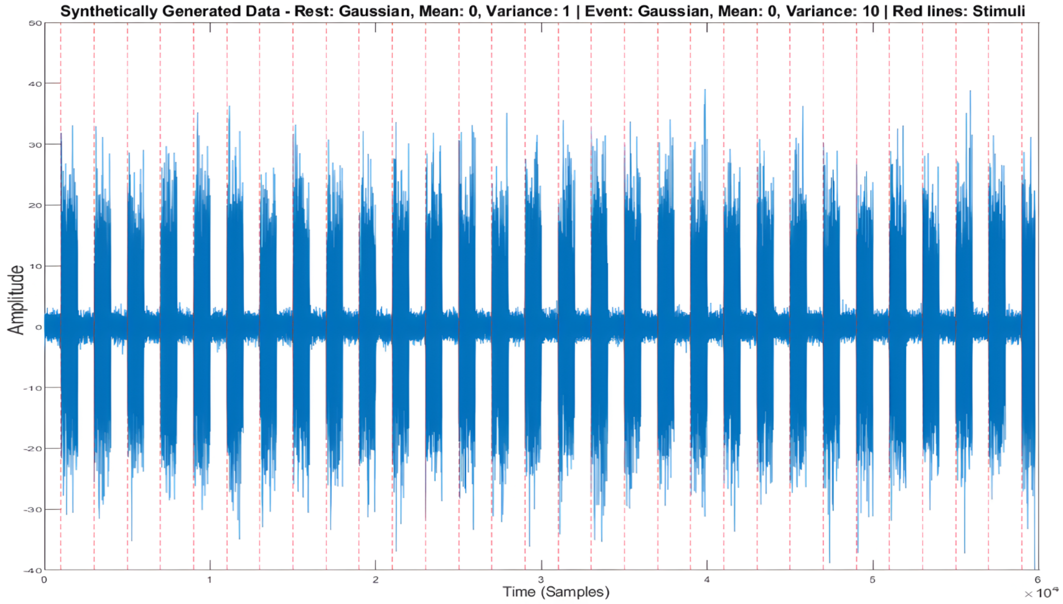
Here, Figure 5.1, Figure 5.2, Figure 5.3, and Figure 5.4 provide visual representations of each simulated dataset, corresponding to the distributions and parameters listed in Table 5.1. In these graphs, the red dashed vertical lines indicate the stimulus points, dividing the data into the rest condition (left side of the stimulus) with 1000 samples and the event data intervals (right side of the stimulus) with 1000 samples.

5.1.1.2 Other Benchmark Features to Compare

This section aims to select features for the study based on their common usage in the literature and their suitability in terms of computational times, accessibility, and availability in MATLAB software. The statistical feature group includes time domain metrics, such as moments. The features in this group are computed using notations where x represents the data point, μ denotes the mean, n indicates the number of samples, $x(t)$ represents the time domain signal and $k = 0, 1, 2, ..., n-1$. The statistical features consist of the mean, variance, skewness, kurtosis, median, range, and root-mean-square. The spectral feature group comprises three features that are computed from the frequency domain estimation of the data. These features consider the entire one-sided frequency domain for their computations. The spectral features include power, mean frequency in the frequency domain, and median frequency in the frequency domain. Lastly, the transformational feature group utilizes the discrete cosine transform (DCT). The DCT coefficients are considered individual features during the classification step. By selecting these features, a comprehensive representation of the data and an effective classification analysis are aimed. Table 5.2 presents the chosen



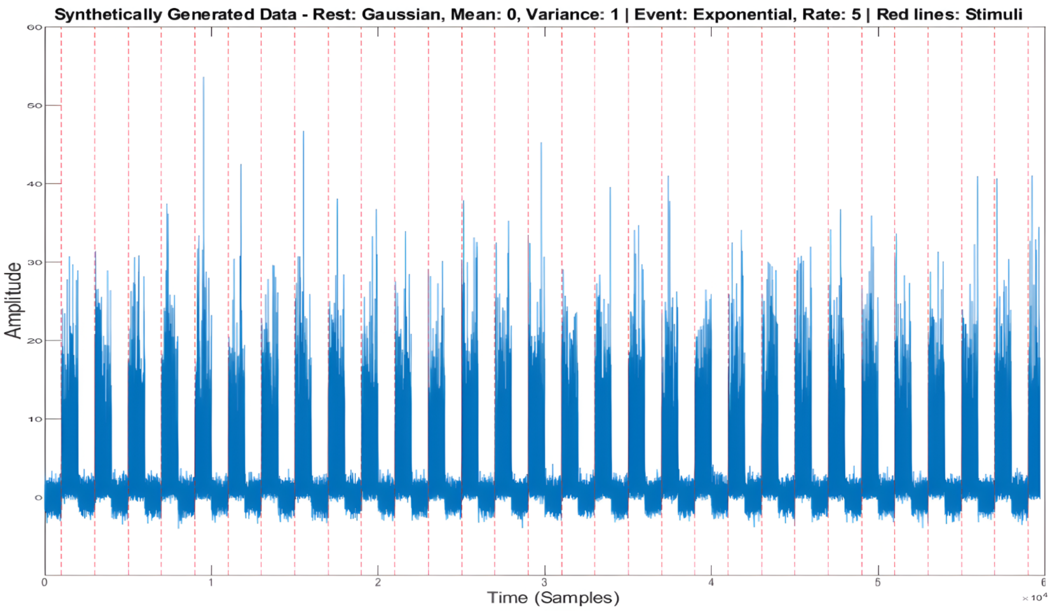
(a) Rest: Gaussian, μ: 0, σ^2: 1.00, Event: Gaussian, μ: 10, σ^2: 1.00



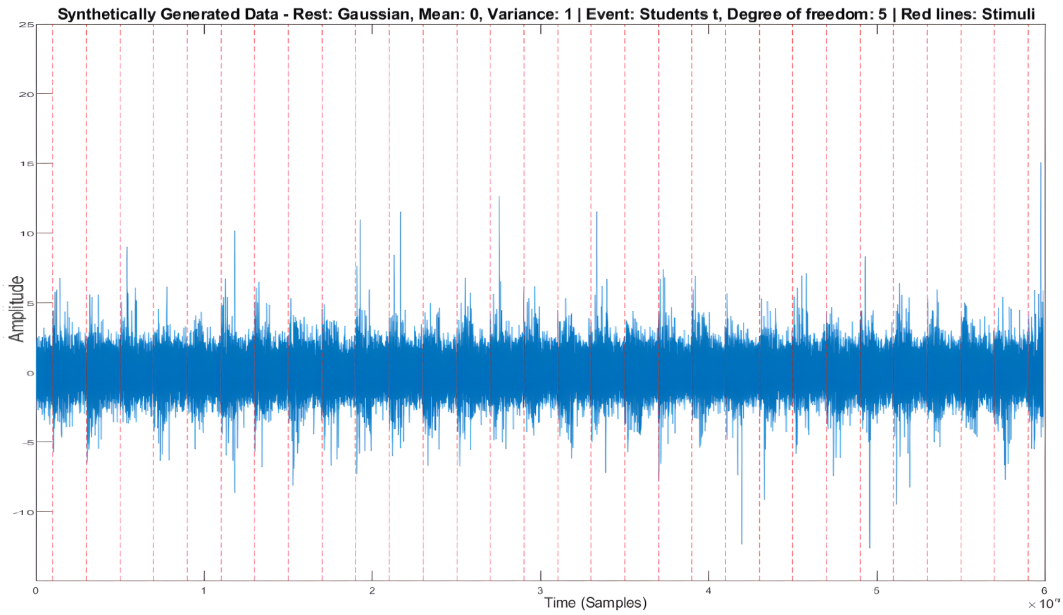
(b) Rest: Gaussian, μ: 0, σ^2: 1.00, Event: Gaussian, μ: 0, σ^2: 10

Figure 5.1: Simulated datasets, part 1.

benchmark features, categorized into three groups: statistical, spectral, and transformational, where x represents a sample, μ represents the mean, n corresponds to the number of samples, $x(t)$ denotes the time domain signal and k ranges from 0 to $n-1$.

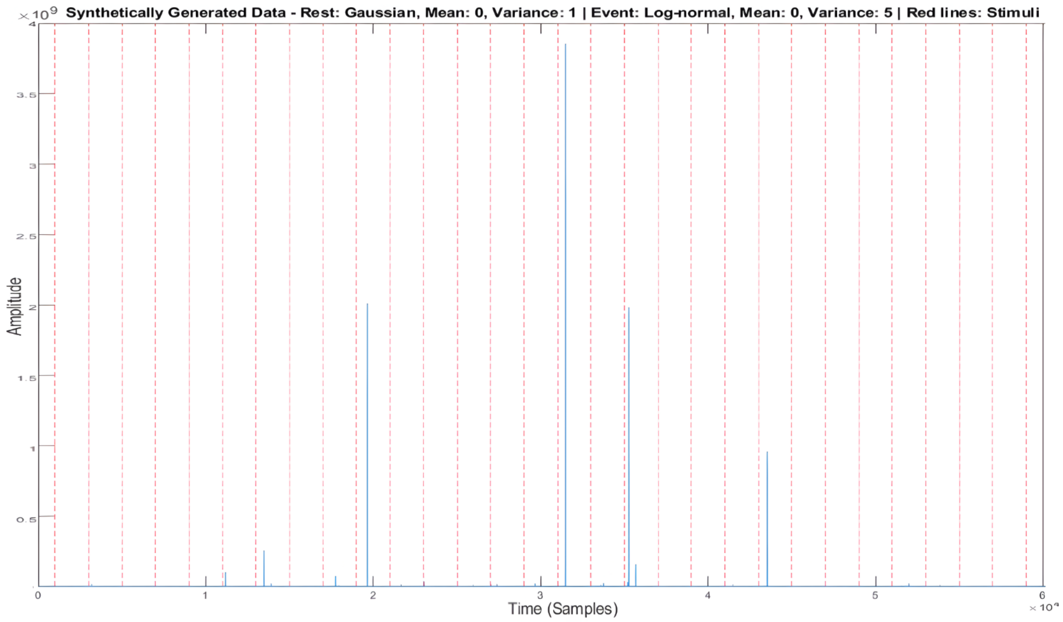


(a) Rest: Gaussian, μ: 0, σ^2: 1.00, Event: Exponential, Rate: 5

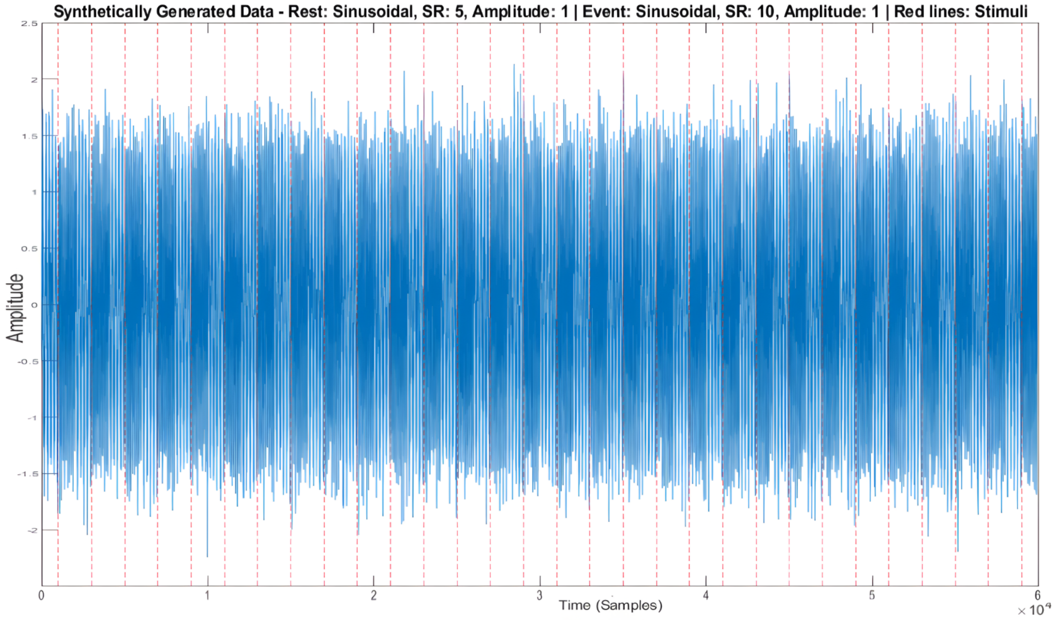


(b) Rest: Gaussian, μ: 0, σ^2: 1.00, Event: Student's t, ν: 5

Figure 5.2: Simulated datasets, part 2.



(a) Rest: Gaussian, μ: 0, σ^2: 1.00, Event: Log-normal, μ: 0, σ^2: 5

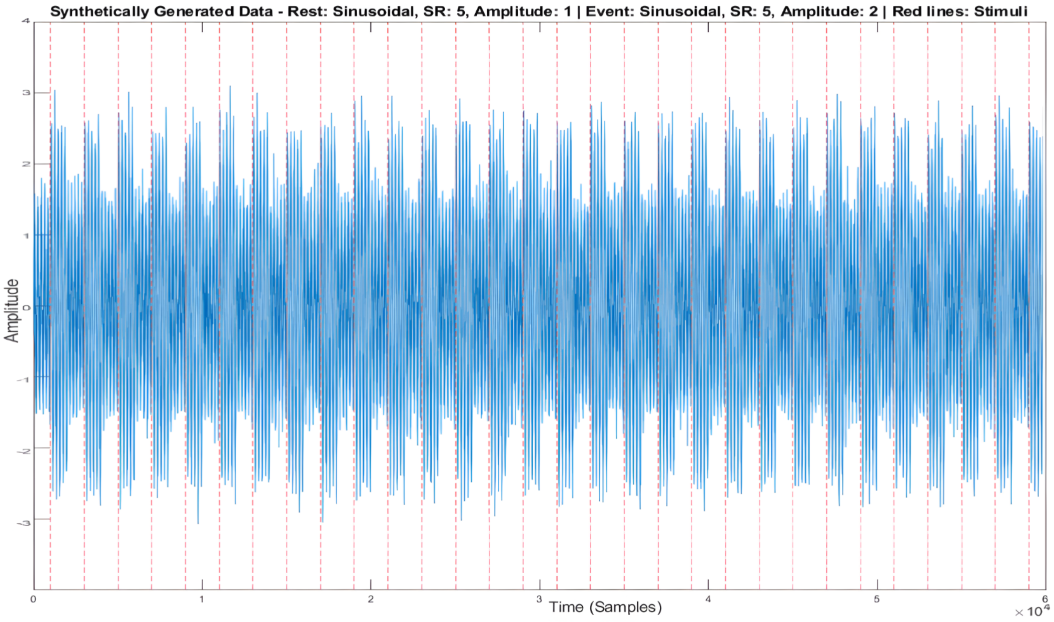


(b) Rest: Sinusoidal, Amp.: 1.00, F_s: 5, Event: Sinusoidal, Amp.: 1.00, F_s: 10

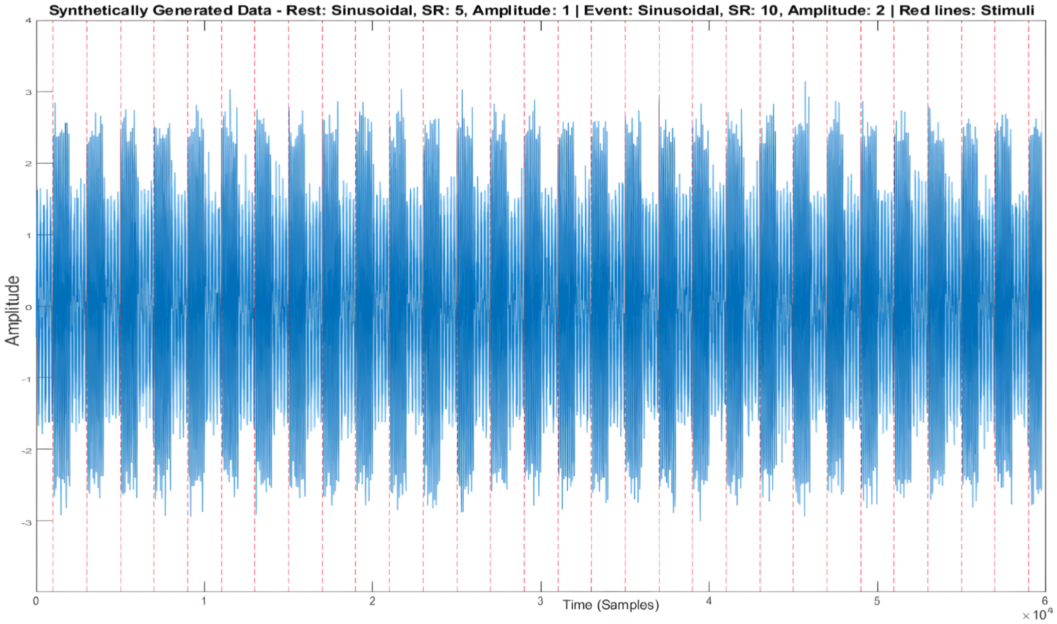
Figure 5.3: Simulated datasets, part 3.

5.1.1.3 Results and Discussion for Synthetic Data Analysis 1

The classification matrix is generated by applying the proposed algorithm to compute individual feature values, as well as extracting benchmark features. For classification,



(a) Rest: Sinusoidal, Amp.: 1.00, F_s: 5, Event: Sinusoidal, Amp.: 2, F_s: 5



(b) Rest: Sinusoidal, Amp.: 1.00, F_s: 5, Event: Sinusoidal, Amp.: 2, F_s: 10

Figure 5.4: Simulated datasets, part 4.

the K-NN algorithm with a value of K set to 10 is employed, along with the K-fold cross-validation method using K set to 10 for evaluating accuracy. To manage the extensive results, only accuracy values are reported, but additional evaluation

Table 5.2: The compilation of feature groups and their corresponding features utilized in this investigation.

Feature Group	Feature	Formula
Statistical	Mean	$\frac{\sum x}{n}$
	Variance	$\frac{\sum(x-\mu)^2}{n}$
	Skewness	$\frac{\sum(x-\mu)^3}{(n-1)\sigma^3}$
	Kurtosis	$n\frac{\sum(x-\mu)^4}{\sum(x-\mu^2)^2}$
	Median	$\frac{n+1}{2}$
	Range	$max - min$
	Root-mean-square	$\sqrt{\frac{\sum x^2}{n}}$
Spectral Features	Power	$\int_{-\infty}^{\infty} x^2 \lvert x(t)\rvert^2 dx$
	Mean frequency	$\frac{\sum x}{n}$, frequency domain
	Median frequency	$\frac{n+1}{2}$, frequency domain
Transformational	DCT coefficients	$(\frac{1}{n})\sum_{u=0}^{n-1} x(n)cos(k2\pi u/n)$

metrics such as sensitivity and specificity, derived from the confusion matrix, are also computed and available upon request.

Since the number of units in both the rest and event conditions is equal, the classification accuracies are unbiased. When both conditions are random or exhibit indifference with respect to the feature used, the accuracies converge to 0.5. Therefore, an accuracy value of 0.5 indicates poor discrimination performance, suggesting that the feature contributes minimally or not at all. Conversely, an accuracy value of 1 signifies perfect discrimination between the two data categories using the respective feature. Roughly speaking, accuracy values ranging from 0.8 to 0.9 can be considered good, while values above 0.9 can be regarded as very good for such experiments. The features are evaluated in both univariate and multivariate conditions to observe their discrimination performances. Accuracy values are reported as the primary evaluation metric, indicating the performance of the features in discriminating between rest and event conditions. The unbiased nature of the classification accuracies and the reference values provide insights into the discrimination capabilities of the features used.

Univariate Classification Accuracies: Table 5.3 presents the accuracy results for the proposed algorithm using NFC as the pitch detection method. The performance of the algorithm is evaluated on different datasets, and the results are discussed independently before comparing them.

Table 5.3: The accuracy results of the proposed algorithm with the NFC pitch frequency detection method. Abbreviations: G.: Gaussian, E.: Exponential, L.: Log-normal, S.: Sinusoidal, A.: Amplitude.

Feature	G., μ: 0, σ^2: 1 G., μ: 10, σ^2: 1	G., μ: 0, σ^2: 1 G., μ: 0, σ^2: 10	G., μ: 0, σ^2: 1 E., Rate: 5	G., μ: 0, σ^2: 1 St., ν: 5	G., μ: 0, σ^2: 1 L., μ: 0, σ^2: 5	S., A.: 1, F_s: 5 S., A.: 1, F_s: 10	S., A.: 1, F_s: 5 S., A.: 2, F_s: 5	S., A.: 1, F_s: 5 S., A.: 2, F_s: 10
Euclidean	0.97	0.81	0.91	0.68	0.80	0.78	1.00	1.00
Square Euc.	0.98	0.81	0.91	0.67	0.80	0.68	1.00	1.00
City Block	0.97	0.81	0.84	0.60	0.80	0.69	0.95	0.94
Minkowski	0.98	0.80	0.90	0.68	0.80	0.78	1.00	1.00
Chebyshev	0.92	0.80	0.92	0.79	0.80	0.66	1.00	1.00
DTW	0.98	0.81	0.84	0.59	0.81	0.77	1.00	1.00
t_{int}	0.98	0.51	0.63	0.51	0.52	0.61	0.64	0.77

According to Table 5.3, for datasets with varying mean or amplitude values, the proposed algorithm with NFC achieves good accuracy results (>0.8). However, for datasets with lower mean differences, the algorithm's performance ranges from bad to average (0.59 to 0.81). Notably, the proposed algorithm with NFC is not suitable for datasets with periodicity differences, as it yields relatively lower accuracies (0.66 to 0.78).

Comparing the dissimilarity metrics, there is no significant difference in their performances across different datasets. However, when dealing with datasets exhibiting high variance and outliers, choosing the Chebyshev distance metric is recommended. On the other hand, the t_{int} feature yields poor results compared to the dissimilarity metrics.

Overall, the proposed algorithm with NFC pitch detection shows good discriminability in datasets with different mean or amplitude values. However, its performance is poorer for datasets with lower mean differences or periodicity differences. Among the dissimilarity metrics, the choice of Chebyshev distance is preferable for datasets with high variance and outliers. Conversely, the t_{int} feature does not provide satisfactory discrimination results.

Table 5.4 presents the accuracy values for proposed algorithm features with the PEFAC pitch detection method for eight simulated datasets.

Table 5.4: The accuracy results of the proposed algorithm with the PEFAC pitch frequency detection method. Abbreviations: G.: Gaussian, E.: Exponential, L.: Log-normal, S.: Sinusoidal, A.: Amplitude.

Feature	G., μ: 0, σ^2: 1	G., μ: 0, σ^2: 1	G., μ: 0, σ^2: 1	G., μ: 0, σ^2: 1	G., μ: 0, σ^2: 1	S., A.: 1, F_s: 5	S., A.: 1, F_s: 5	S., A.: 1, F_s: 5
	G., μ: 10, σ^2: 1	G., μ: 0, σ^2: 10	E., Rate: 5	St., ν: 5	L., μ: 0, σ^2: 5	S., A.: 1, F_s: 10	S., A.: 2, F_s: 5	S., A.: 2, F_s: 10
Euclidean	0.53	1.00	1.00	0.80	1.00	0.59	0.99	0.97
Square Euc.	0.52	1.00	1.00	0.79	1.00	0.59	1.00	0.98
City Block	0.52	1.00	1.00	0.60	1.00	0.60	0.87	0.84
Minkowski	0.53	1.00	1.00	0.81	1.00	0.60	0.99	0.97
Chebyshev	0.51	1.00	1.00	1.00	1.00	0.56	1.00	0.99
DTW	0.51	1.00	1.00	0.62	1.00	0.63	0.95	0.92
t_{int}	0.50	0.51	0.51	0.51	0.81	0.51	0.50	0.52

The results in Table 5.4 show that, in contrast to the NFC method, the PEFAC method fails to discriminate the data conditions with mean value differences. However, the proposed algorithm achieves perfect accuracy values (1.00) in discriminating data conditions with higher variances, outliers, and oscillations.

Among the dissimilarity metrics, the Chebyshev distance yields better results overall compared to other metrics, particularly for datasets with more outliers. On the other hand, Dynamic Time Warping (DTW) is preferable (0.63 to 0.95) for discriminating data conditions with different periodicity. However, the t_{int} value is not recommended as a feature on its own. Instead, it is suggested to be converted into dissimilarity metrics for better discrimination performance.

The accuracy values for different statistical features are presented in Table 5.5.

Table 5.5: The accuracy results of the statistical features. Abbreviations: G.: Gaussian, E.: Exponential, L.: Log-normal, S.: Sinusoidal, A.: Amplitude.

Feature	G., μ: 0, σ^2: 1	G., μ: 0, σ^2: 1	G., μ: 0, σ^2: 1	G., μ: 0, σ^2: 1	G., μ: 0, σ^2: 1	S., A.: 1, F_s: 5	S., A.: 1, F_s: 5	S., A.: 1, F_s: 5
	G., μ: 10, σ^2: 1	G., μ: 0, σ^2: 10	E., Rate: 5	St., ν: 5	L., μ: 0, σ^2: 5	S., A.: 1, F_s: 10	S., A.: 2, F_s: 5	S., A.: 2, F_s: 10
Mean	1.00	0.90	1.00	0.53	1.00	0.51	0.50	0.50
Var.	0.50	1.00	1.00	1.00	1.00	0.50	1.00	1.00
Skew.	0.50	0.50	1.00	0.76	1.00	0.50	0.62	0.64
Kurt.	0.51	0.50	1.00	1.00	1.00	0.50	1.00	1.00
Med.	1.00	0.89	1.00	0.50	1.00	0.50	0.57	0.56
Ran.	0.50	1.00	1.00	1.00	1.00	0.50	1.00	1.00
RMS	1.00	1.00	1.00	1.00	1.00	0.50	1.00	1.00

Table 5.5 provides insights into the performance of these features when applied to random datasets generated under specific statistical distributions. Upon analyzing the results, certain patterns emerge. The mean and median features demonstrate remarkable discrimination capabilities, achieving perfect accuracy (1.00) in distinguishing datasets with high mean differences. Similarly, the variance feature performs exceptionally well, achieving perfect accuracy (1.00) in identifying datasets with high variance conditions, as expected.

However, the statistical features encounter challenges when it comes to identifying periodic signals with varying oscillation rates. In these cases, the features exhibit lower accuracy values, ranging from 0.50 to 0.51. This suggests that the statistical features struggle to effectively capture the characteristics of such periodic signals.

Among the statistical features, the root mean square (RMS) stands out as the most reliable and consistent discriminator. It consistently achieves perfect classification accuracy (1.00) across a wide range of datasets, indicating its robustness. However, it should be noted that the RMS feature fails to differentiate sinusoidal datasets with the same amplitude, resulting in an accuracy of 0.50 for this particular condition.

The accuracy values for spectral features are presented in Table 5.6.

Table 5.6: The accuracy results of the spectral features. Abbreviations: G.: Gaussian, E.: Exponential, L.: Log-normal, S.: Sinusoidal, A.: Amplitude.

Feature	G., μ: 0, σ^2: 1	G., μ: 0, σ^2: 1	G., μ: 0, σ^2: 1	G., μ: 0, σ^2: 1	G., μ: 0, σ^2: 1	S., A.: 1, F_s: 5	S., A.: 1, F_s: 5	S., A.: 1, F_s: 5
	G., μ: 10, σ^2: 1	G., μ: 0, σ^2: 10	E., Rate: 5	St., ν: 5	L., μ: 0, σ^2: 5	S., A.: 1, F_s: 10	S., A.: 2, F_s: 5	S., A.: 2, F_s: 10
Power	1.00	1.00	1.00	1.00	1.00	0.50	1.00	1.00
Mean Fr.	1.00	0.51	1.00	0.51	0.91	0.83	1.00	1.00
Median Fr.	1.00	0.52	1.00	0.50	0.83	1.00	1.00	1.00

Table 5.6 provides the accuracy results for the spectral features that are evaluated based on their ability to discriminate random datasets with different statistical distributions. Upon analyzing the results, it is evident that the performance of spectral features varies across different conditions, ranging from poor to perfect accuracy (0.50 to 1.00). Among the spectral features, the power feature stands out as the best overall performer. It achieves perfect accuracy (1.00) in most cases, except for the sinusoidal dataset with the same amplitude condition, where its accuracy drops.

The power feature demonstrates consistent and reliable discrimination capabilities across a wide range of datasets. It effectively captures the power characteristics of the signals and performs well in distinguishing datasets with varying statistical distributions. In addition to the power feature, other spectral features such as mean frequency and median frequency also exhibit reasonably good discrimination abilities.

Table 5.7 showcases the accuracy results of using discrete cosine transformation (DCT) coefficients as independent features.

Table 5.7: The accuracy results of the DCT coefficient features. Abbreviations: G.: Gaussian, E.: Exponential, L.: Log-normal, S.: Sinusoidal, A.: Amplitude.

Feature	G., μ: 0, σ^2: 1 G., μ: 10, σ^2: 1	G., μ: 0, σ^2: 1 G., μ: 0, σ^2: 10	G., μ: 0, σ^2: 1 E., Rate: 5	G., μ: 0, σ^2: 1 St., ν: 5	G., μ: 0, σ^2: 1 L., μ: 0, σ^2: 5	S., A.: 1, F_s: 5 S., A.: 1, F_s: 10	S., A.: 1, F_s: 5 S., A.: 2, F_s: 5	S., A.: 1, F_s: 5 S., A.: 2, F_s: 10
DCT – 1	1.00	1.00	1.00	0.86	1.00	0.69	1.00	1.00
DCT – 5	1.00	1.00	1.00	0.97	1.00	0.98	1.00	1.00
DCT – 10	1.00	1.00	1.00	0.99	1.00	1.00	1.00	1.00

Table 5.7 includes different statistical distributions and their corresponding accuracies. When examining the results, it becomes evident that as the number of DCT coefficients used increases, the classification accuracy generally improves, which aligns with expectations. The accuracy of DCT coefficients tends to be high across various statistical distributions. Specifically, the DCT-1 coefficient achieves high accuracy in most cases, except for the sinusoidal dataset with an amplitude of 1 and a sampling frequency of 5, where the accuracy drops to 0.69. The DCT-5 and DCT-10 coefficients consistently exhibit perfect accuracy (1.00) in almost all conditions. It should be noted that using a higher number of DCT coefficients increases the computational complexity. However, the trade-off for increased accuracy may justify the additional computational resources required.

By comparing the results in Table 5.3 and Table 5.4, some conclusions about the performance of the PEFAC and NFC pitch frequency detection algorithms within the proposed algorithm can be made. Firstly, the PEFAC algorithm is suggested for datasets with changing values, characterized by higher variances and outliers. It excels in such scenarios, as it can effectively detect pitch frequency variations. On the other hand, if the data exhibits more deterministic behavior with mean value changes between different conditions, the NFC algorithm is a suitable choice for pitch frequency detection.

When comparing the proposed algorithm results with other feature groups, it becomes evident that its performance varies depending on the datasets being analyzed. The dissimilarity metrics used in the proposed algorithm, particularly when combined with the PEFAC pitch frequency detection algorithm, can be advantageous for anomaly detection in datasets with high variance. In such cases, the proposed algorithm can achieve perfect classification accuracies (1.00).

In summary, the choice between the PEFAC and NFC pitch frequency detection algorithms within the proposed algorithm depends on the characteristics of the data. The PEFAC algorithm is suitable for datasets with changing values, while the NFC algorithm is more appropriate for data with deterministic behavior. Proper parameter tuning or smart parameter selection algorithms are necessary for optimizing the performance of the proposed algorithm. When compared to other feature groups, the proposed algorithm, particularly with the PEFAC pitch frequency detection algorithm, can be highly effective in detecting anomalies in datasets with high variance, offering the potential for perfect classification accuracies.

Multivariate Classification Accuracies: The feature groups are combined to create a classification matrix, allowing the data to be classified in multiple dimensions. The classification accuracies of these feature groups can be seen in Table 5.8. As expected, the multivariate classification accuracies are higher compared to the accuracies achieved using univariate features. However, it is important to note that the computational complexity increases significantly when utilizing multivariate features, resulting in longer computational times compared to the respective univariate classifications.

Table 5.8 provides an overview of the multivariate classification accuracies for different feature groups. According to these results, employing multivariate feature groups generally improves classification accuracies compared to using univariate features. However, it is important to consider that the use of multivariate features yields higher computational complexity. The choice of feature group depends on the dataset char-

Table 5.8: The multivariate classification accuracies of feature groups. Abbreviations: G.: Gaussian, E.: Exponential, L.: Log-normal, S.: Sinusoidal, A.: Amplitude.

Feature	G., μ: 0, σ^2: 1	G., μ: 0, σ^2: 1	G., μ: 0, σ^2: 1	G., μ: 0, σ^2: 1	G., μ: 0, σ^2: 1	S., A.: 1, F_s: 5	S., A.: 1, F_s: 5	S., A.: 1, F_s: 5
	G., μ: 10, σ^2: 1	G., μ: 0, σ^2: 10	E., Rate: 5	St., ν: 5	L., μ: 0, σ^2: 5	S., A.: 1, F_s: 10	S., A.: 2, F_s: 5	S., A.: 2, F_s: 10
Prop. alg. Dist. (NCF)	0.96	0.81	0.90	0.80	0.80	0.96	1.00	1.00
Prop. alg. All (NCF)	0.97	0.81	0.91	0.81	0.80	0.97	1.00	1.00
Prop. alg. Dist. (PEF)	0.52	1.00	1.00	1.00	1.00	0.75	1.00	1.00
Prop. alg. All (PEF)	0.50	1.00	1.00	1.00	1.00	0.78	1.00	1.00
Stat. Moments	1.00	1.00	1.00	1.00	1.00	0.50	1.00	1.00
Stat. All	1.00	1.00	1.00	1.00	1.00	0.50	1.00	1.00
Spect. All	1.00	1.00	1.00	1.00	0.92	1.00	1.00	1.00
All Features	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00

acteristics, but combining all available feature families tends to yield the best performance, achieving perfect classification accuracies.

5.1.2 Synthetic Data Analysis 2: Comparing the Moving Windows Parameters

The primary objective of this study is to present preliminary findings on the applicability of dissimilarity metrics for future research on anomaly detection in time series data. Additionally, the study aims to investigate the performance of dynamic time warping (DTW) in detecting anomalies in quasi-periodic data, serving as a basis for future studies. Another goal is to introduce a novel approach for calculating the grand average within sliding windows. To achieve these objectives, three synthetic datasets are initially generated, each representing a distinct data behavior. Furthermore, noisier versions of each dataset are created by introducing additional noise. Periodic anomalous intervals are also included in each dataset to simulate real-world scenarios. The datasets are then processed using sliding windows to simulate real-time data analysis.

5.1.2.1 Synthetic Dataset 2 Generation

Three synthesized datasets are utilized to evaluate dissimilarity-based anomaly detection approaches. Each dataset consists of a single time series with 1,000,000 samples. The data is divided into baseline and event conditions, with 500 instances of each. These conditions are concatenated to form the datasets. Additionally, two versions of each dataset are created: one with relatively low noise (signal-to-noise ratio of 10dB) and another with high noise (signal-to-noise ratio of 0dB). The noise is added to each sample using a Gaussian distribution to enhance the realism of the generated datasets. The event data is generated with different distributional parameters compared to the baseline data, inducing anomaly-like structures in the time series. Each dataset exhibits distinct characteristics of anomalies.

The first dataset represents anomalous structures with higher mean values compared to the baseline data, while the overall data behavior is not quasi-periodic. The baseline data in this dataset is produced by a Gaussian distribution with a mean value (μ) of 0 and a variance value (σ^2) of 1 (white noise). On the other hand, the event data is generated using a Gaussian distribution with a variance of 1 and a mean value of 3. Figure 5.5 provides a visual representation of this dataset, where groups of lower amplitude samples correspond to the baseline, and the others represent anomalous conditions.

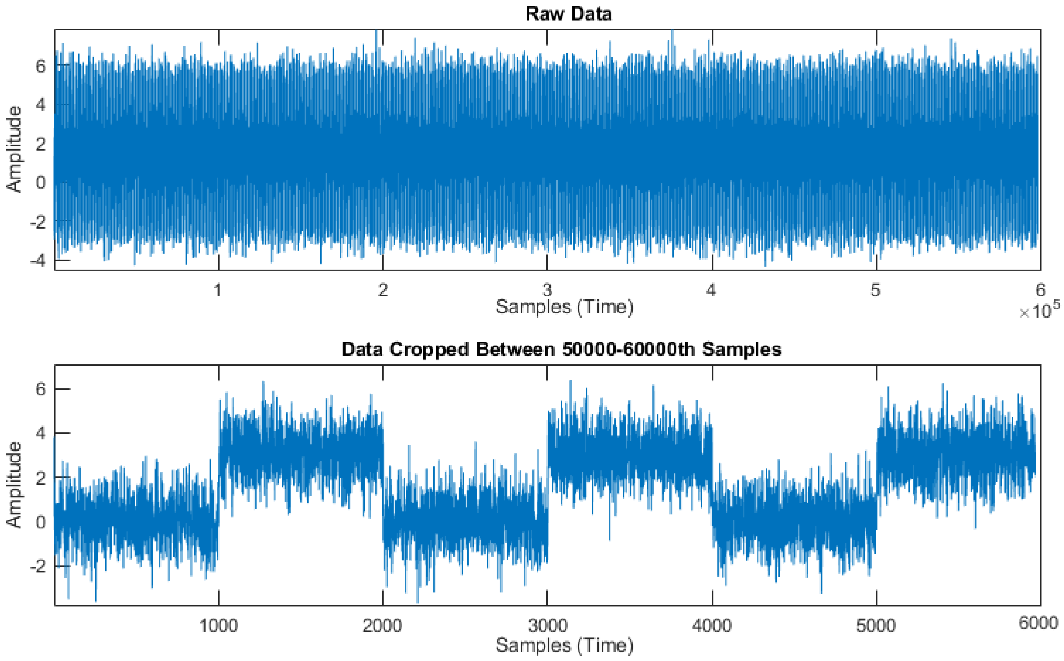


Figure 5.5: Generated dataset 1 (upper) and zoomed in (lower). Baseline: Gaussian with $\mu = 0$ and $\sigma^2 = 1$. Anomalous: Gaussian with $\mu = 3$ and $\sigma^2 = 1$.

The second dataset aims to test high-amplitude anomalous conditions in quasi-periodic data behavior. The samples in this dataset are generated from a sinusoidal function with a predefined amplitude and sampling rate (Fs). The baseline data is generated with a sampling rate of 5 Hz, and its oscillation amplitude, including the added noise, is set around 1. In contrast, the event data samples have the same sampling rate but an oscillation amplitude of 3. Figure 5.6 displays a snapshot of this dataset, where the baseline data segments can be distinguished by their lower amplitudes.

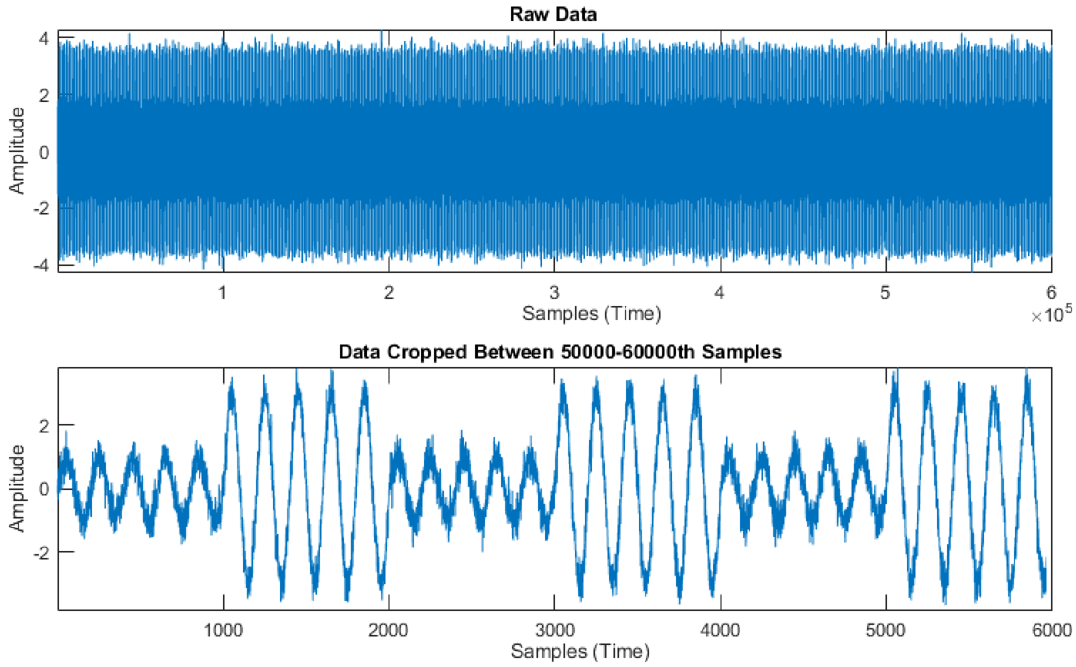


Figure 5.6: Generated dataset 2 (upper) and zoomed in (lower). Baseline: Sinusoidal with amplitude = 1 and sampling rate = 5 Hz. Anomalous: Sinusoidal with amplitude = 3 and sampling rate = 5 Hz.

The final dataset conveys the quasi-periodic characteristics of the previous dataset though with a different sampling rate for the anomalous condition compared to the baseline data. In this case, the oscillation amplitudes are set to 1, but the sampling rate for the anomalous condition is selected as 15 Hz. This dataset aims to test the effect of higher oscillatory periodic behavior in anomalous data. Figure 5.7 provides an example representation of this dataset, showing the distinguishably higher oscillation rate in the anomalous data, despite the similar amplitudes.

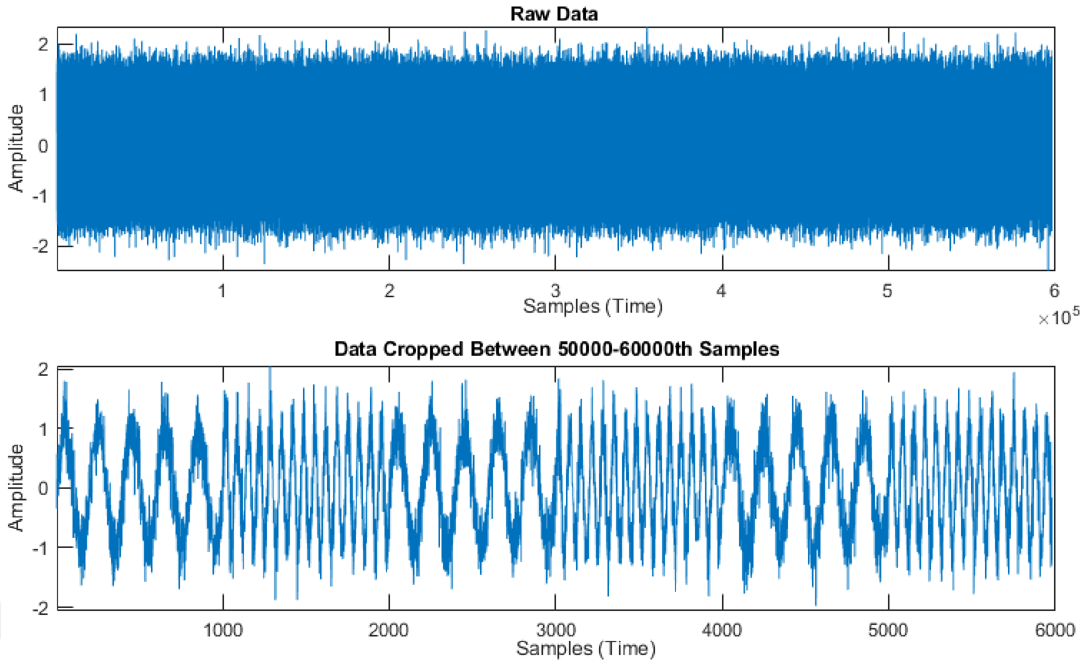


Figure 5.7: Generated dataset 3 (upper) and zoomed in (lower). Baseline: Sinusoidal with amplitude = 1 and sampling rate = 5 Hz. Anomalous: Sinusoidal with amplitude = 1 and sampling rate = 15 Hz.

5.1.2.2 Experimental Setup and Feature Extraction

In this study, to maintain simplicity, both the window size and slide size are kept constant. The window size is set to 1000, which matches the size of the conditions (baseline and event) in the datasets. Additionally, a window size of 2000 is used to cover both condition sizes in independent trials. The slide size, on the other hand, is set to 500, 1000, and 2000 independently for different analysis cases. The slide size determines the step size or overlap between consecutive windows. The processing algorithm operates on the cropped sliding windows, one at a time, and applies the necessary operations for analysis, such as feature extraction and decision-making. After processing a window, the algorithm proceeds to the next window based on the window size and slide size.

The determination of samples in the next window depends on the window size and the chosen slide size. For example, if the window size is 1000 and the slide size is 500, the next window will start 500 samples after the previous window's starting point.

This process continues until all sliding windows have been processed.

In the feature extraction step of this study, the dissimilarities between the i^{th} grand average (GA_i) and the i^{th} window (W_i) are computed sample-wise. These dissimilarities serve as the feature values for anomaly detection. Three dissimilarity metrics are selected for this purpose: Euclidean distance, Chebyshev distance, and dynamic time warping (DTW) distance.

Euclidean distance is a commonly used distance measurement that calculates the linear and one-to-one dissimilarity between corresponding sample pairs of two vectors with the same length. It can be seen as the L2 norm and is represented by the formula:

$$d_{euc}(GA_i, W_i) = \sqrt{\sum_{j=i}^{n}(GA_{i,j} - W_{i,j})^2} \tag{5.1}$$

Here, j represents the sample number, n is the window size, GA_i denotes the grand average, and W_i represents the window.

Chebyshev distance, also known as the $L\infty$ norm, measures the maximum sample-wise difference between GA_i and W_i. It relies on one-to-one matching between the samples and can be calculated using the formula:

$$d_{cheb}(GA_i, W_i) = \max_j |GA_{i,j} - W_{i,j}| \tag{5.2}$$

The final dissimilarity metric used in this study is dynamic time warping (DTW). DTW allows for a non-linear matching between the samples of GA_i and W_i through an optimization process involving various constraints. However, due to its computational complexity, DTW is typically slower compared to Euclidean and Chebyshev distances. A simplified formulation of DTW dissimilarity between GA_i and W_i is as follows:

$$d_{dtw}(GA_i, W_i) = |GA_{i,j} - W_{i,k}| + \min_j \begin{cases} D(j-1,k) \\ D(j-1,k-1) \\ D(k-1) \end{cases} \tag{5.3}$$

Here, j and k represent the sample numbers, and D is the optimization function used in DTW.

These dissimilarity metrics capture different aspects of similarity or dissimilarity between the grand average and each window, providing valuable information for detecting anomalies in the time series data. It should be noted that the computational complexity of DTW makes it slower compared to Euclidean and Chebyshev distances.

5.1.2.3 Clustering and Decision Making

The final step in the algorithm involves determining whether a window is anomalous or not based on the extracted dissimilarity metric values. The unsupervised clustering algorithm used for this classification task is k-medoids, also known as the partitioning around medoids (PAM) algorithm.

The k-medoids algorithm begins by setting the number of clusters parameter, denoted as k. In this study, since there are two classes to classify (anomalous and non-anomalous), the value of k is chosen as two. The algorithm selects k arbitrary feature values as initial seeds. The remaining feature values are then assigned to one of these initial seeds based on their distances from the seed values. The distance is computed for each remaining feature value and measures the difference between the feature values and the initial seeds. This process is repeated by selecting different arbitrary values for k until the difference is minimized. Finally, based on the assigned seeds, the feature values are labeled into respective classes. Therefore, according to the main hypothesis, windows marked as anomalous are assumed to have a different cluster of values compared to the rest of the windows.

The clustering approach in this study follows a subject-wise learning approach, where the classification model is dynamically trained and updated to reduce the classification error rate. Since no prior information is available about the data, the error rate is expected to be high. To mitigate this, a small portion of the data from the beginning is selected as the training set. After conducting empirical experiments with different percentages (0.5%, 1%, 2%, 3%, 5%, and 10%), it is found that a training set size of 2% of the total data length yields the most suitable results with the minimum training set size. In other words, using smaller training set percentages (0.5% and 1%) resulted in poor performance, while increasing the training set percentage did not significantly improve the performance.

The training portion of the data is marked from the start of the data where the algorithm operates, but the decisions made for the respective windows are not considered. The unsupervised classification is applied to the entire data, resulting in decisions for each window during the operation. As more data are processed, the accuracy of detecting anomalous windows increases. After the entire data is processed, the overall accuracy is computed by dividing the number of correctly identified anomalous windows by the total number of anomalous windows in the simulated datasets.

5.1.2.4 Results and Discussion for Synthetic Data Analysis 2

A MATLAB figure is created to provide real-time monitoring of the data processing. This figure allows the user to visualize various components, including the cropped window, cumulative data, recent data, cumulative feature values, grand average up to the current window, and the decisions made for each window so far. A snapshot of this MATLAB figure is shown in Fig.5.8.

In the first experiment, the impact of window size on accuracy is examined by comparing the accuracies achieved with window sizes of 1000 and 2000. The window size of 1000 is specifically chosen to match the exact length of the anomalous condition in the simulated datasets. Each window with a slide size of 1000 represents a unique window in the data, alternating between the baseline and anomalous intervals in each iteration.

On the other hand, the window size of 2000 is selected to create windows composed

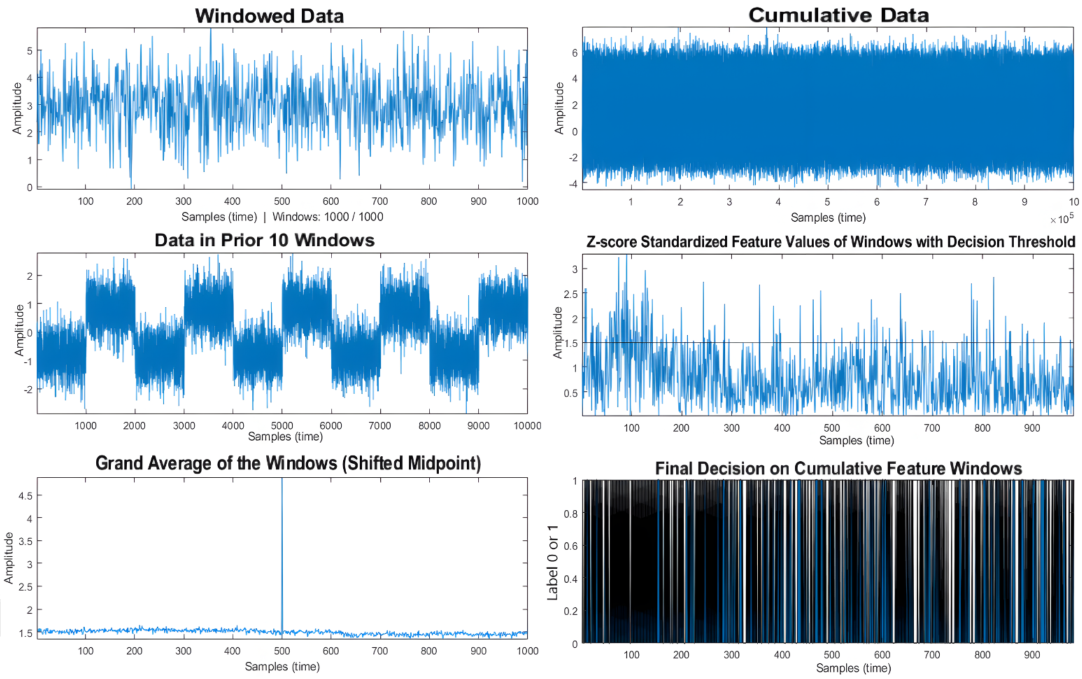


Figure 5.8: Snapshot of the real-time operation in a single MATLAB figure.

of both baseline and anomalous parts with equal length. With a slide size of 1000, the window alternates between baseline and anomalous for half of its length. This means that each window, except the first and the last, is processed twice. The accuracy results for different window sizes in each simulated dataset, using the Euclidean dissimilarity metric and a slide size of 1000, are presented in Table 5.9.

From the results in Table 5.9, it can be observed that the matching window size of 1000 achieves better accuracy in detecting anomalies in the sinusoidal dataset with different amplitude values compared to the window size of 2000. In fact, the matching window size of 1000 perfectly detects the anomalous windows in this sinusoidal dataset. However, for the other datasets, a window size of 2000 yields better results than the matching window size. This difference in performance might be attributed to the distinct patterns and harmonic formations in the grand average of the sinusoidal dataset with varying amplitude values, as depicted in Fig.5.9.

The second experiment focuses on comparing different slide sizes while keeping the window size fixed at 1000 samples. The matching window size of 1000 samples is selected as the standard. The slide sizes are set as 500, 1000, and 2000.

Table 5.9: Comparison of accuracies for window size using the Euclidean dissimilarity metric, with a slide size of 1000. ’B’ refers to baseline, ’Sinus’ abbreviates sinusoidal, ’A’ stands for anomalous windows, ’SNR’ refers to signal-to-noise ratio, ’Fs’ is the sampling rate.

Dataset generation conditions		Window size	
Distribution	SNR	1000	2000
B: Gaussian. $\mu = 0$, $\sigma 2 = 1$ and A: Gaussian. $\mu = 3$, $\sigma 2 = 1$	0	0.7661	0.7534
	10	0.6835	0.7787
B: Sinus. Amplitude = 1, Fs = 5 and A: Sinus. Amplitude = 3, Fs = 5	0	1.000	0.8991
	10	0.9980	0.7105
B: Sinus. Amplitude = 1, Fs = 5 and A: Sinus. Amplitude = 1, Fs = 15	0	0.7189	0.8226
	10	0.5305	0.5749

With a slide size of 500 samples, except for the first and last windows, the samples are investigated twice, alternating between perfectly matching and half-matching of the correct anomalous windows. A slide size of 1000 ensures perfect matching of the windows, while a slide size of 2000 is used as a control set, capturing only the baseline parts. The accuracy results for different slide sizes using the Euclidean dissimilarity metric and a window size of 1000 can be found in Table 5.10.

Table 5.10: Comparison of accuracies for different slide sizes using the Euclidean dissimilarity metric, with a window size of 1000. ’B’ refers to baseline, ’Sinus’ abbreviates sinusoidal, ’A’ stands for anomalous windows, ’SNR’ refers to signal-to-noise ratio, ’Fs’ is the sampling rate.

Dataset generation conditions		Slide size		
Distribution	SNR	500	1000	2000
B: Gaussian. $\mu = 0$, $\sigma 2 = 1$ and A: Gaussian. $\mu = 3$, $\sigma 2 = 1$	0	0.7918	0.7661	0.5020
	10	0.7037	0.6835	0.5183
B: Sinus. Amplitude = 1, Fs = 5 and A: Sinus. Amplitude = 3, Fs = 5	0	0.5665	1.000	0.5102
	10	0.5395	0.9980	0.5061
B: Sinus. Amplitude = 1, Fs = 5 and A: Sinus. Amplitude = 1, Fs = 15	0	0.7506	0.7189	0.5000
	10	0.5485	0.5305	0.5082

From the results in Table 5.10, it is evident that the selection of the correct slide size is crucial for accurate analysis. In the case of the sinusoidal dataset with different

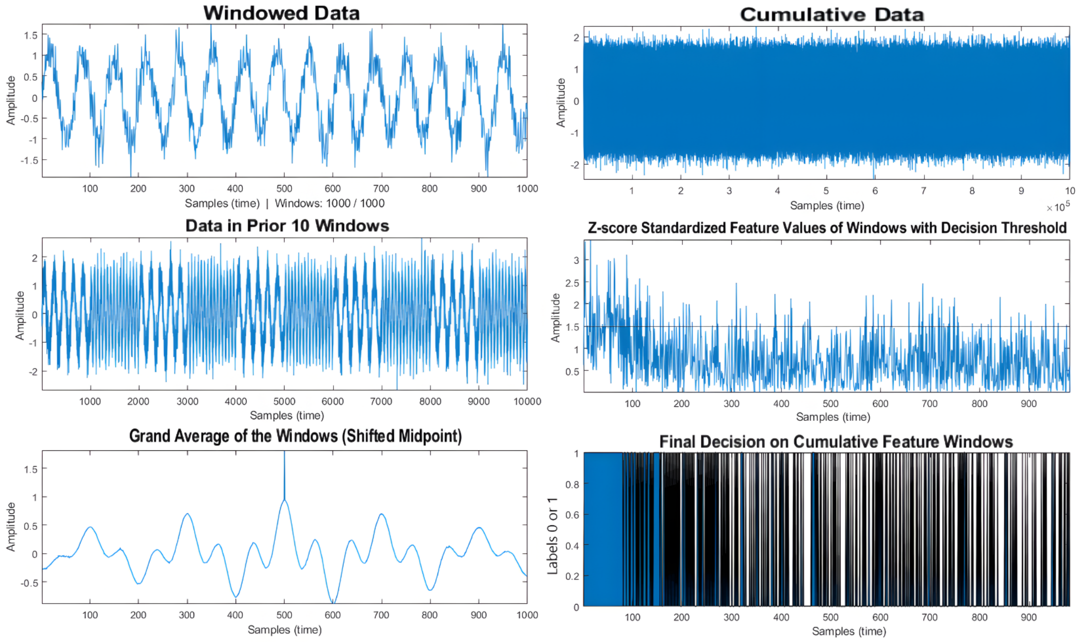


Figure 5.9: Snapshot of the real-time operation in a single MATLAB figure where a pattern observed in the grand average analysis.

amplitude values for data conditions, matching the patterns of the grand average with the window is essential. The classification accuracy drops to 56% for a slide size that is half-matching compared to perfect matching.

The third experiment compares the performances of different dissimilarity metrics for a fixed window size of 1000 samples and a slide size of 1000 samples. The classification accuracy results of the dissimilarity metrics in discriminating the anomalous windows are presented in Table 5.11.

From the accuracy values of the dissimilarity metrics in Table 5.11, it can be observed that the Euclidean distance performs better than the Chebyshev distance in all situations. However, for the sinusoidal datasets, the dynamic time warping (DTW) metric achieves the highest accuracy, with perfect classification in some cases. The superior performance of DTW in the sinusoidal datasets can be attributed to its ability to handle non-linear matching of the periodic peaks between the current window and the grand average.

Indeed, the results from Tables 5.9, 5.10, and 5.11 suggest that noise can have a detri-

Table 5.11: Comparison of accuracies for different dissimilarity metrics with a window size of 1000 and a slide size of 1000. 'B' refers to baseline, 'Sinus' abbreviates sinusoidal, 'A' stands for anomalous windows, 'SNR' refers to signal-to-noise ratio, 'Fs' is the sampling rate.

Dataset generation conditions		Dissimilarity metric		
Distribution	SNR	Euclidean	Chebyshev	DTW
B: Gaussian. μ = 0, $\sigma 2 = 1$ and A: Gaussian. μ = 3, $\sigma 2 = 1$	0	0.7661	0.6354	0.5479
	10	0.6835	0.6415	0.5794
B: Sinus. Amplitude = 1, Fs = 5 and A: Sinus. Amplitude = 3, Fs = 5	0	1.000	1.000	0.9980
	10	0.9980	0.9084	0.9919
B: Sinus. Amplitude = 1, Fs = 5 and A: Sinus. Amplitude = 1, Fs = 15	0	0.7189	0.6863	1.000
	10	0.5305	0.5092	0.8870

mental effect on the performance of anomaly detection in the datasets. Therefore, applying a noise reduction operation before the anomaly detection process could potentially improve the performance. Noise reduction techniques such as filtering or denoising algorithms can help remove unwanted noise from the data and enhance the detectability of anomalies. However, it's important to consider the computational complexity of these techniques, especially in real-time analyses where efficiency is crucial. Noise reduction operations can introduce additional computational overhead, which may not be desirable in real-time applications where quick response times are required.

Finally, the computational times of the dissimilarity metrics were compared to analyze their performance. The results, presented in Table 5.12, provide the average computational times in seconds for the window sizes of 1000 and 2000.

Table 5.12: Comparison of average computational times (in s) of dissimilarity metrics for the number of windows.

Dissimilarity metric	Window size	
	1000	2000
Euclidean	167.54	178.57
Chebyshev	164.41	171.05
Dynamic time warping	198.90	216.73

From the findings in Table 5.12, it can be observed that the Chebyshev dissimilarity metric exhibits the fastest computational requirements among the evaluated metrics. This makes it a favorable choice when computational efficiency is a primary concern. However, it is important to note that the selection of a dissimilarity metric should not solely depend on computational time. The accuracy and performance of the metric in detecting anomalies must also be taken into account. While the Chebyshev metric offers faster computation, it may not provide the highest accuracy compared to the Euclidean or Dynamic Time Warping (DTW) metrics, especially when dealing with datasets containing periodic or quasi-periodic patterns.

5.2 Real Data Analyses

The datasets used in this study are sourced from publicly available data on the Physionet website [328]. Three real electrocardiogram (ECG) datasets are selected to evaluate the performance of the proposed approach. Anomalies are intentionally introduced in specific segments of the data to assess the effectiveness of the proposed algorithm in detecting anomalies.

5.2.1 Real Data Analysis 1: Introducing Displacement Noise to an Interval

This part of the analysis includes the introduction of the random Gaussian Noise to the fixed interval of the real regular ECG datasets. This modification simulates the electrode displacement during a measurement, and results in a collective anomaly interval. This section is based on the author's publication [21].

5.2.1.1 Real Datasets and Experimental Design

The data utilized in this study were sourced from three publicly available benchmark datasets accessible on the Physionet website [328]. These datasets include the MIT-BIH Long Term ECG dataset, the MIT-BIH Normal Sinus Rhythm dataset [333], and the European ST-T database [334]. The selection of these benchmark datasets was driven by the requirement for quasi-periodic data and the availability of standardized datasets for comparative analysis. The Physionet website was deemed a suitable resource for obtaining such standardized quasi-periodic datasets [328].

Each dataset consisted of recordings with 1,000,000 samples per subject, captured at varying sampling rates. The MIT-BIH Long Term ECG dataset comprised 7 subjects with recordings at a sampling rate of 128 Hz, the MIT-BIH normal sinus rhythm ECG dataset comprised 18 subjects with recordings at 128 Hz, and the European ST-T database comprised 90 subjects with recordings at 250 Hz. The data from each subject's recordings were extracted into '.mat' files and checked for any missing or corrupted samples.

A unique modification introduced in this study involved replacing 5 percent of sequential samples, specifically between the halfway point and towards the end of the data, with random Gaussian noise. This modification was carried out under four experimental conditions that varied the relative data range values. The Gaussian normal distribution, known as random white noise, was considered an anomalous distribution due to its association with faulty measurements and natural imperfections. Sensor displacement or disconnection, a common measurement error, often results in the emission of hum noise with a Gaussian distribution. Consequently, the decision was made to utilize the Gaussian distribution as the primary anomaly condition for detection.

Considering the nature of the data modality (i.e., ECG), the four anomalous conditions were defined as anomalous intervals and served as the test anomalies. These intervals were specific segments of samples in the subjects' data, ranging from the middle sample (50%) to the samples corresponding to 55% of the total data length. Synthetic Gaussian normal distribution data was generated to replace these intervals. The anomalous conditions were represented by outlier-rich cumulative anomaly intervals, with S as a scale factor determined by the minimum and maximum values of the subjects' data. The noise samples were generated using a normal probabilistic data distribution with a mean of 0 and a standard deviation of 1, which were then rescaled to match the subject-wise data range. Four different anomalous conditions were created by employing scale factors of 2, 1, 0.5, and 0.25, with 0 representing a DC signal where hum noise is suppressed or very low compared to the base data.

Figure 5.10 displays a visualization of the analysis-ready datasets for the MIT-BIH

Long Term ECG dataset. Similarly, Figure 5.11 showcases the analysis-ready datasets for the MIT-BIH normal sinus rhythm dataset, and Figure 5.12 illustrates the analysis-ready datasets for the European ST-T change dataset. These visualizations provide a comprehensive overview of the datasets, including the presence of anomalies and their variations based on the different scale factors applied.

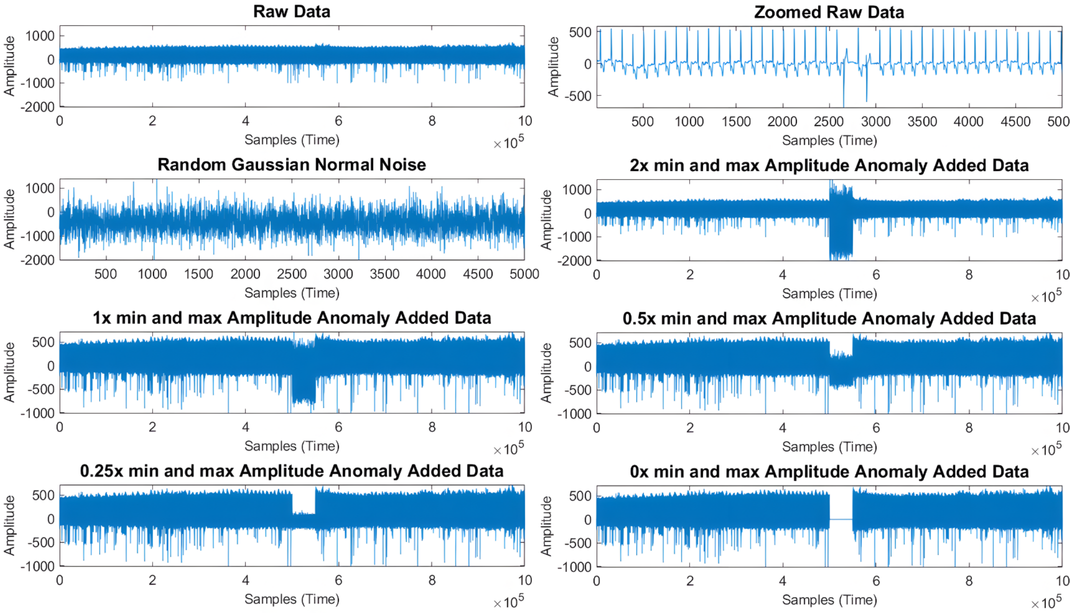
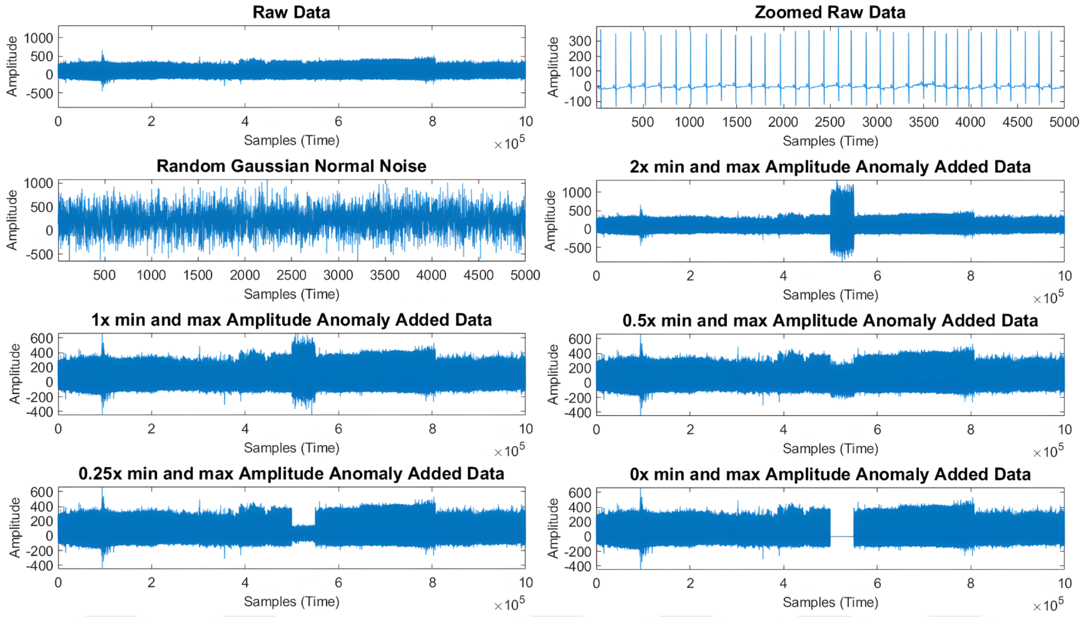


Figure 5.10: Sample data from the MIH-BIH Long Term ECG dataset and examples of the modified data from this dataset.

Each of the figures presents eight sub-figures in a grid layout, representing the different visualizations for the corresponding dataset. The sub-figures are arranged from left to right and top to bottom, depicting the following: the complete raw data, a zoomed-in view of the raw data, the anomaly interval, and the modified data with anomaly additions. The scale factors used for the anomaly additions are 2, 1, 0.5, 0.25, and 0, respectively, for the datasets mentioned above.

To evaluate the performance of the proposed anomaly detection approach, we compare it with other benchmark approaches commonly used for sequential anomaly detection tasks in the literature. The selected benchmark approaches for comparison are HOTSAX (Heuristically Ordered Time Series using Symbolic Aggregate ApproXimation), LDOF (Local Distance-Based Outlier Factor), and Grubbs' algorithm.

The HOTSAX algorithm utilizes symbolic representation to label intervals in time


Raw Data

Zoomed Raw Data

Random Gaussian Normal Noise

2x min and max Amplitude Anomaly Added Data

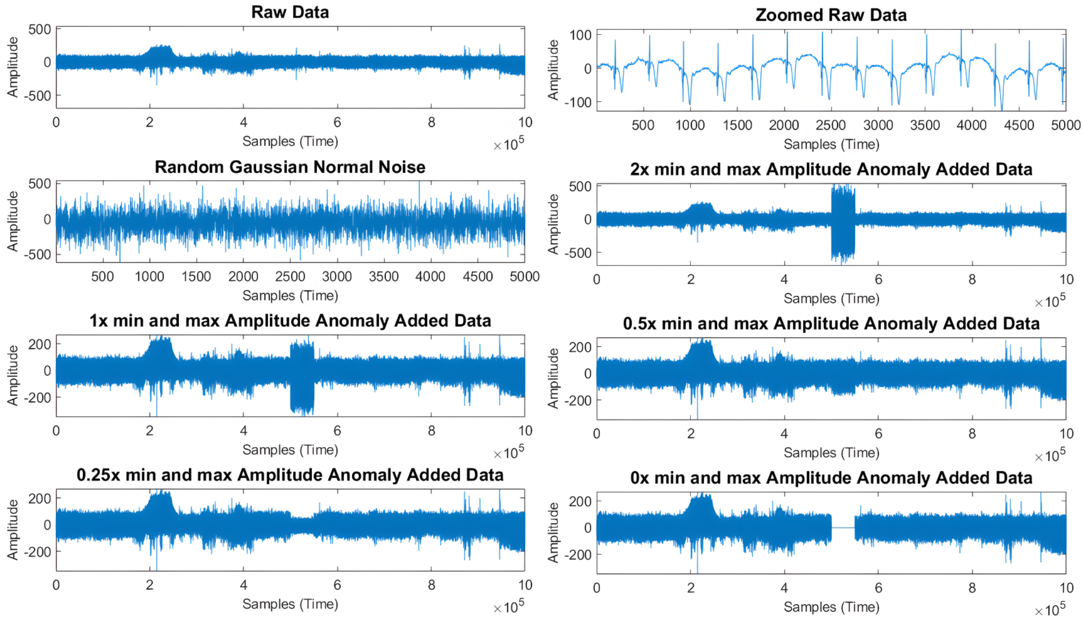
1x min and max Amplitude Anomaly Added Data

0.5x min and max Amplitude Anomaly Added Data

0.25x min and max Amplitude Anomaly Added Data

0x min and max Amplitude Anomaly Added Data

Figure 5.11: Sample data from the MIT-BIH normal sinus rhythm dataset and examples of the modified data from this dataset.


Raw Data

Zoomed Raw Data

Random Gaussian Normal Noise

2x min and max Amplitude Anomaly Added Data

1x min and max Amplitude Anomaly Added Data

0.5x min and max Amplitude Anomaly Added Data

0.25x min and max Amplitude Anomaly Added Data

0x min and max Amplitude Anomaly Added Data

Figure 5.12: Sample data from the European ST-T change dataset and examples of the modified data from this dataset.

series data and then compares them to detect anomalous intervals. On the other hand, the LDOF algorithm calculates the LDOF scores of each sample by dividing the sum of the K-nearest neighbor distances of a sample by the inner K-nearest neighbor distance. Similarly to the proposed anomaly hypothesis, anomalous samples often yield higher LDOF values, making them distinguishable. Finally, Grubbs' approach is a statistical algorithm that computes test statistics for each sample based on data standardization. It uses the absolute difference between a sample value and the data mean, divided by the standard deviation. The more anomalous a sample is, the higher its test score becomes.

Based on the observations from Figure 5.10, Figure 5.11, and Figure 5.12, both datasets can be assumed to be wide-sense stationary (WSS) since they exhibit negligible trend, a constant mean, and quasi-constant variance throughout the data. Therefore, these datasets can be analyzed using parametric statistical approaches such as Grubbs' approach.

The experimental parameters used in the study include a data size of 1,000,000 samples per subject obtained from the Physionet public access database pool [328]. For the moving windows approach, the window size and slide size are set to 5000 samples per window, ensuring that there is no overlap between windows and that all samples in the datasets are covered by windowing-based approaches like HOTSAX. The selection of these window parameters is based on the requirement of covering at least 10 PQRST structures in each window for both datasets and ensuring computational efficiency suitable for the hardware used. Having more than 10 PQRST structures in the ECG data is crucial for spectral domain estimation and the pitch frequency detection process employed in the proposed algorithm. The remaining benchmark approaches are applied to the entire set of subjects' data without predefined window and slide size parameters. The specific parameters for the benchmark approaches are selected based on their default or commonly used values. The significance score for Grubbs' approach is set to 0.05, the K-value for the LDOF approach is set to 100, and the number of symbolic segments for the HOTSAX algorithm is set to 4.

5.2.1.2 Results and Discussion for Real Data Analysis 1

The performance evaluation of the anomaly detection applications is carried out using a confusion matrix. The true positives represent correctly labeled anomaly intervals, which account for 5% of the overall data. False negatives occur when anomaly intervals are incorrectly labeled. True negatives are obtained by correctly classifying the remaining 95% of the samples as non-anomalies. False positives arise when regular data intervals are incorrectly classified as anomalies. The confusion matrix scores are computed for each iteration of the applied approaches, based on their detection of anomalies in the iterated data window. The scores in the confusion matrix are cumulatively added for each subject, resulting in the final confusion matrix after analyzing all the subjects for their anomalous intervals.

Several performance metrics are computed using the final confusion matrix for each applied approach. These metrics include accuracy, sensitivity, specificity, the F1-score, and Matthew's correlation coefficient. Due to the class imbalance caused by the heterogeneous number of anomalous and non-anomalous intervals, accuracy is not a reliable metric. However, it is included in the results tables as it is a popular performance measure. The performance metrics for each dataset and method are recorded and reported.

Following the application of the selected algorithms under the experimental conditions, Table 5.13 presents the performance metric results for the applications on the MIT-BIH Long Term ECG dataset. Similarly, Table 5.14 displays the results for the MIT-BIH normal sinus rhythm dataset, and Table 5.15 shows the results for the European ST-T change dataset.

According to the findings in Table 5.13, the proposed algorithm appears to be highly sensitive to the anomaly scaling factor. As the size of the anomalies decreases, so does the performance of the proposed algorithm. The proposed algorithm has poor sensitivity and F1-scores, especially for scaling factors of $0.25 * Min - Max$ and $0 * Min - Max$, which can be considered inlier-like anomalies. These findings suggest that the proposed method may produce a high number of false negatives for inlier-like anomalies, and that the effectiveness of the proposed algorithm may be heavily dependent on the amplitudes of the cumulative anomalies. Nonetheless, the overall results in Table 5.13, show that the proposed algorithm outperforms HOT-SAX, LDOF, and Grubbs in terms of accuracy, sensitivity, specificity, F1-score, and MCC metrics.

Table 5.13: The performance results of the methods for the modified MIT-BIT Long Term ECG dataset. *MCC* stands for Matthew's Correlation Coefficient.

Anomaly Scaling	**Method**	**Performance Metric**				
		Accuracy	**Sensitivity**	**Specificity**	**F1-Score**	**MCC**
2*Min-Max	HOTSAX	0.91	0.69	0.96	0.79	0.71
	LDOF	0.63	0.12	**1.00**	0.21	0.26
	Grubbs	0.96	0.94	0.96	0.95	0.91
	Proposed alg.	**0.99**	**1.00**	0.99	**0.99**	**0.99**
1*Min-Max	HOTSAX	0.68	0.40	0.75	0.47	0.26
	LDOF	0.64	0.11	0.98	0.20	0.15
	Grubbs	0.95	0.34	0.95	0.51	0.58
	Proposed alg.	**0.99**	**0.98**	**0.99**	**0.98**	**0.97**
0.5*Min-Max	HOTSAX	0.60	0.08	0.78	0.14	0.05
	LDOF	0.66	0.13	**1.00**	0.23	0.00
	Grubbs	**0.95**	0.03	0.95	0.06	0.00
	Proposed Alg.	0.90	**0.35**	0.99	**0.52**	**0.61**
0.25*Min-Max	HOTSAX	0.65	**0.17**	0.85	**0.26**	0.14
	LDOF	0.62	0.10	**0.98**	0.18	**0.16**
	Grubbs	**0.95**	0.00	0.95	0.00	0.00
	Proposed alg.	0.62	0.09	0.97	0.16	0.14
0*Min-Max	HOTSAX	0.90	**0.15**	0.93	**0.26**	**0.26**
	LDOF	0.61	0.10	**1.00**	0.18	0.00
	Grubbs	**0.95**	0.00	0.95	0.00	0.00
	Proposed alg.	0.59	0.11	**1.00**	0.20	0.00

According to Table 5.14, when dealing with larger anomalies, the proposed algorithm on the modified MIT-BIT Normal Sinus ECG dataset achieves perfect scores for accuracy, sensitivity, specificity, F1-score, and MCC. However, for inlier-like anomalies with smaller amplitudes, it's performance drops significantly, resulting in poor sensitivity and F1-scores. The amplitudes of the anomalies in the dataset heavily influence the algorithm's effectiveness. For larger anomaly scaling factors, it consistently outperforms Grubbs' and LDOF methods. However, for smaller inlier-like anomalies, the algorithm may produce a high number of false negatives, indicating its limitations in detecting subtle anomalies effectively.

Table 5.14: The performance results of the methods for the modified MIT-BIT Normal Sinus ECG dataset. *MCC* stands for Matthew's Correlation Coefficient.

Anomaly Scaling	**Method**	**Performance Metric**				
		Accuracy	**Sensitivity**	**Specificity**	**F1-Score**	**MCC**
2*Min-Max	HOTSAX	0.93	0.85	0.92	0.89	0.80
	LDOF	0.65	0.12	**1.00**	0.21	0.0
	Grubbs	0.96	0.99	0.96	0.99	0.93
	Proposed alg.	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
1*Min-Max	HOTSAX	0.59	0.10	0.61	0.17	0.00
	LDOF	0.60	0.11	**1.00**	0.19	0.00
	Grubbs	0.95	0.35	0.95	0.49	0.41
	Proposed alg.	**0.99**	**1.00**	0.99	**0.99**	**0.98**
0.5*Min-Max	HOTSAX	0.73	0.45	0.74	0.53	0.27
	LDOF	0.76	0.18	**1.00**	0.30	0.00
	Grubbs	0.94	0.05	0.95	0.10	0.13
	Proposed alg.	**0.98**	**1.00**	0.98	**0.74**	**0.62**
0.25*Min-Max	HOTSAX	0.82	0.71	0.82	0.76	0.62
	LDOF	0.63	0.12	**1.00**	0.21	0.00
	Grubbs	0.94	0.00	0.95	0.00	0.00
	Proposed alg.	**0.98**	**0.81**	0.99	**0.89**	**0.82**
0*Min-Max	HOTSAX	0.88	**0.75**	0.87	**0.80**	**0.69**
	LDOF	0.55	0.00	0.92	0.00	0.00
	Grubbs	**0.94**	0.00	**0.95**	0.00	0.00
	Proposed alg.	0.68	0.17	0.93	0.25	0.17

Finally, The proposed algorithm's performance was evaluated on the modified European ST-T Change ECG dataset, and the results are shown in Table 5.15. Overall, the algorithm showed promising performance, outperforming LDOF in most cases, and other methods for higher anomaly scaling cases in terms of accuracy, sensitivity, specificity, F1-score, and MCC metrics.

The modified simulation conditions applied to the MIT-BIH Long Term and European ST-T Change ECG datasets, which have the same data modality, yield similar outcomes for the employed approaches, as shown in Table 5.13, Table 5.14, and Table 5.15, aligning with our expectations. Notably, as evidenced by higher F1-score and

Table 5.15: The performance results of the methods for the modified European ST-T Change ECG dataset. *MCC* stands for Matthew's Correlation Coefficient.

Anomaly Scaling	**Method**	**Performance Metric**				
		Accuracy	**Sensitivity**	**Specificity**	**F1-Score**	**MCC**
2*Min-Max	HOTSAX	0.95	0.89	0.95	0.91	0.85
	LDOF	0.58	0.07	0.96	0.12	0.16
	Grubbs	0.96	0.99	0.96	0.91	0.90
	Proposed alg.	**0.99**	**1.00**	**0.99**	**0.99**	**0.99**
1*Min-Max	HOTSAX	0.57	0.12	0.58	0.18	0.00
	LDOF	0.61	0.10	0.98	0.17	0.19
	Grubbs	0.95	0.54	0.95	0.69	0.59
	Proposed alg.	**0.96**	**0.58**	**0.99**	**0.73**	**0.74**
0.5*Min-Max	HOTSAX	0.60	0.17	0.72	0.24	0.19
	LDOF	0.66	0.11	**0.98**	0.20	0.28
	Grubbs	**0.95**	0.04	0.95	0.07	0.07
	Proposed alg.	0.84	**0.21**	**0.98**	**0.35**	**0.35**
0.25*Min-Max	HOTSAX	0.68	**0.16**	0.81	0.24	0.16
	LDOF	0.59	0.07	0.97	0.13	0.00
	Grubbs	**0.95**	0.00	0.95	0.00	0.00
	Proposed alg.	0.70	0.13	**0.98**	**0.25**	**0.18**
0*Min-Max	HOTSAX	0.93	**0.70**	0.94	**0.80**	**0.69**
	LDOF	0.48	0.00	0.91	0.00	0.00
	Grubbs	**0.95**	0.00	**0.95**	0.00	0.00
	Proposed alg.	0.92	0.04	0.94	0.08	0.06

Matthew's correlation coefficient metrics, the proposed approach consistently outperforms the other benchmark methods in detecting induced anomalous intervals. When the simulation conditions are examined, it is discovered that as the amplitude of the anomalous time intervals increases, the proposed algorithm achieves higher F1-score and Matthew's correlation coefficient values, achieving perfection (score of 1) in both evaluation criteria. Its effectiveness, however, decreases for smaller inlier-like anomalies. In summary, the proposed algorithm produced promising overall results but had limitations in handling smaller anomalies and the absence of anomalies.

In addition to the performance metrics, the computational times for each algorithm were recorded and averaged during the simulations for each subject. The simulations were conducted on a personal computer equipped with an 11th Gen i7-11700KF processor running at 3.60 GHz. The computational times were then rescaled relative to the proposed algorithm, which served as the baseline (assigned a value of 1x), and the results can be found in Table 5.16.

Table 5.16: The relative average computational times corresponding to the proposed algorithm through the simulations.

Algorithm	Relative Average Computational Time
Proposed alg.	1x
HOTSAX	5.8x
LDOF	0.76x
Grubbs’	0.03x

According to Table 5.16, the relative average computational times reveal that the proposed algorithm outperforms other non-parametric approaches, such as HOTSAX, in terms of speed. Specifically, the proposed algorithm demonstrates a significant speed advantage over HOTSAX, with a relative computational time of 1x compared to 58x. On the other hand, the LDOF algorithm shows a slightly faster performance than the proposed algorithm, with a relative computational time of 0.76x. Meanwhile, Grubbs’ approach exhibits the fastest computational time among the methods, with a relative value of 0.03x. However, it is important to note that Grubbs’ approach requires stationary datasets to function properly.

5.2.2 Real Data Analysis 2: Frequency Search Range Effect on Pre-annotated Benchmark ECG Datasets

This part of the analysis consists of testing the algorithm’s performance on the pre-annotated anomalies in some benchmark ECG datasets. Here, two of the benchmark datasets are selected based on their containing the type III anomaly from the Physionet Databank [328]. Hence, the two datasets, namely, the MIT-BIH Malignant Ventricular Ectopy [332] and the CU Ventricular Tachyarrhythmia [75] are selected to observe the frequency range selection effects to locate the pitch frequency within the

CEP [210], pitch frequency detection algorithm. Moreover, both of those benchmark datasets include annotations for both the ventricular anomalies and some other signal-related discords with shorter durations, which can be considered type II anomalies.

5.2.2.1 Real Datasets and the Usage of the Annotated Data

The first dataset, namely the MIT-BIH Malignant Ventricular Ectopy [332] consists of 22 different ECG recordings from subjects who have ventricular disorders that can be considered anomalies, such as ventricular tachycardia, flutter, or fibrillation. Each subject in this dataset consists of 525000 samples provided on the website. On the other hand, the CU Ventricular Tachyarrhythmia Dataset [75] consists of recordings from 35 subjects, each with around 127000 samples. Both of the benchmark datasets have sampling rates of 250 Hz.

In this part of the analysis, all of the pre-annotated intervals in the benchmark, including ventricular anomalies are considered anomaly intervals, and the remaining data are considered non-anomaly data. On the other hand, both datasets include some other annotations, such as changes in signal quality, short artifacts, premature beats, and other short-time rhythm-related anomalies. Hence, the proposed algorithm has been tested in both of the annotation cases. The first test includes all of the non-normal annotations as given as the benchmark, and the second test includes only the ventricle-related annotations such as ventricular flutter, ventricular tachycardia, and ventricular fibrillation. Since ventricle-related disorders have relatively higher changes in the signal baseline, it is expected that those anomalies will be detected with higher performance by the proposed approach.

To perform the experiments and obtain the classification results, the data intervals corresponding to the pre-annotated data samples on the Physionet Website are labeled as 1, and the rest of the data samples are labeled as 0. Such labeling is required to form the ground truth for the data samples. Hence, the proposed algorithm is operated through the data of all 22 subjects for the MIT-BIH Malignant Ventricular Ectopy Dataset [332], and 35 subjects for the CU Ventricular Tachyarrhythmia Dataset [75] without any manipulation, subject removal, or data modification. Here, in the analyses of both datasets, the window and slide sizes are selected as 2000 samples, arbitrarily to cover at least 8 seconds of the recordings, which corresponds to 5–8 QRS complexes in each window, as the requirement for those parameters is mentioned in Section 4.1.

5.2.2.2 Results and Discussion for Real Data Analysis 2

The experiments are performed to set the frequency search ranges for the CEP algorithm to locate the pitch frequencies within. Hereby, four scenarios are formed, namely, 1-50 Hz, 1-75 Hz, 1-100 Hz, and 1-125 Hz. The aim of this part is to observe the effects of the frequency search ranges on the classification performance while testing the proposed approach on untouched benchmark ECG datasets.

The results presented in Table 5.17 depict the outcomes of using the CEP algorithm for pitch frequency detection on real annotated benchmark ECG datasets as described in Section 5.2.2.1. The table showcases different frequency search ranges in Hz and the corresponding classification performance metrics such as accuracy and f1-scores for the selected frequency ranges, datasets, and types of annotations.

Table 5.17: The frequency range effect of the pitch frequency detection using the CEP algorithm on real annotated benchmark ECG datasets.

Dataset	Freq. Search Range (Hz)	All Annotations		Only Ventricular Anomalies	
		Accuracy	f1-score	Accuracy	f1-score
MIT-BIH Malignant Ventricular Ectopy	1-50	0.704	0.329	0.830	0.519
	1-75	0.700	0.328	0.823	0.507
	1-100	0.692	0.324	0.822	0.504
	1-125	0.690	0.323	0.815	0.502
CU Ventricular Tachyarrhythmia	1-50	0.792	0.452	0.892	0.631
	1-75	0.780	0.449	0.880	0.607
	1-100	0.774	0.438	0.853	0.599
	1-125	0.772	0.432	0.840	0.592

From Table 5.17, it can clearly be observed that the narrower frequency search range tends to provide slightly better classification performances when the CEP algorithm is used for the pitch detection algorithm. The results show that the performance metric values, i.e., accuracy and f1-score, slightly increase through the narrower frequency search ranges. On the other hand, the table also compares the algorithm's performance between all annotations and only ventricular anomalies and finds that the proposed algorithm provides better results for the ventricular anomalies in the data, as expected due to their more distinctive behavior. This highlights the importance of accurate annotation and targeted analysis to improve the algorithm's performance in detecting specific conditions. It is found that the proposed approach reaches up to 89% accuracy to detect ventricular anomalies, and up to 79% accuracy for all annotated anomalies.

When the datasets are compared, the CU Ventricular Tachyarrhythmia dataset is found to yield better anomaly detection scores than the MIT-BIH Malignant Ventricular Ectopy Dataset. These differences can be attributed to variations in dataset characteristics such as data size, diversity of anomalies, noise levels, and other data-recording conditions.

CHAPTER 6

CONCLUSION AND FUTURE WORKS

6.1 Conclusion

This thesis introduces a novel anomaly detection method designed to detect collective anomalies in quasi-periodic time series data. The proposed algorithm leverages pitch frequency estimation in the spectral domain, a technique commonly used in audio signal processing. By combining the strengths of both the time and frequency domains, the proposed approach provides a comprehensive perspective on locating anomalous patterns. Moreover, the proposed algorithm is highly customizable and adaptable, seamlessly integrating new features and adjusting to evolving data characteristics in a multivariate anomaly detection pipeline. This flexibility ensures its applicability across various domains and datasets, enhancing its practicality not only in the biomedical engineering field but also for all quasi-periodic time series data modalities.

One key feature of the proposed algorithm is its real-time anomaly detection capability. It employs a customizable sliding window approach, allowing it to continuously analyze incoming data for anomalies. Additionally, the algorithm utilizes previous data information in time series data, enabling it to compare new data with previous values and dynamically learn structural patterns. This adaptability makes the proposed algorithm highly effective in detecting anomalies specific to the subject at hand, rendering it a subject-specific anomaly detection algorithm with the linearly increasing time complexity by the data length, providing an $o(n)$ complexity function.

However, it is important to note that the proposed algorithm may not be suitable for locating single-sample outliers that are not significant enough to affect window properties. Furthermore, as the algorithm is specifically designed for quasi-periodic data behavior, its theoretical suitability for time series data that does not exhibit quasi-periodic behavior may be limited.

To evaluate the effectiveness of the proposed algorithm, this study tests it in both synthetic and real data analyses to observe its characteristics. While the synthetic data analyses involve randomly generated data under probabilistic distributions that induce some extreme conditions, the real data analyses focus on testing it on the electrocardiogram (ECG) data modality, which exhibits quasi-periodic time series behavior and is a primary research interest in biomedical signal analysis.

The synthetic data analysis part is further divided into two sub-analyses. The first sub-analysis aims to examine the dissimilarity metric features utilized in the proposed algorithm and compare them with other commonly used features belonging to statistical, spectral, and transformational feature families. On the other hand, the second sub-analysis focuses on investigating the impact of sliding window parameters, particularly when the window size and slide size are set to specific ratios aligned with the baseline data periodicity.

In the real data analysis part 1, three benchmark ECG datasets, namely, the MIT-BIH long-term ECG dataset [328], the MIT-BIH Normal Sinus Rhythm dataset [333], and the European ST-T dataset [334], are adapted to compare the performance of the proposed algorithm against three other benchmark time series anomaly detection algorithms: HOTSAX [312], LDOF [335], and Grubbs' algorithm [336]. The datasets are augmented with random noise across four different range scales for each subject, covering the data interval between half and 5% percent towards the end sample. The performance of the simulations is evaluated using confusion matrices and their associated measures to assess the proposed approach's ability to detect anomaly intervals.

The results indicate that the proposed approach consistently outperforms the benchmark time series anomaly detection algorithms in most simulation conditions and performance metrics. This demonstrates the superiority of the proposed approach in detecting both outlier-like and inlier-like anomalies. Moreover, the proposed algorithm exhibits favorable computational efficiency compared to other benchmark non-parametric approaches designed for non-stationary data. However, it may not outperform the parametric Grubbs' approach in terms of computational time. Nonetheless, the combination of utilizing both spectral and time domains and achieving superior performance values makes the proposed algorithm a promising approach for the detection of quasi-periodic collective anomalies in quasi-periodic time series data.

Finally, the proposed approach is tested in the detection of benchmark annotated real datasets, namely, the MIT-BIH Malignant Ventricular Ectopy [332] and the CU Ventricular Tachyarrhythmia [75] Datasets without any modification or alteration in the annotations or the data. It is found that the proposed approach can locate the annotated data samples with acceptable rates, providing up to 89% accuracy in detection of the ventricular anomalies.

6.2 Future Works

In terms of future work, the proposed algorithm can be further improved by extending the feature extraction, selection, and testing processes. This expansion aims to enhance the classification performance of the algorithm. Additionally, the proposed algorithm's multivariate dynamic clustering approach can be evaluated using various dissimilarity metrics from the time series literature. Computational efficiency can also be enhanced through code and algorithm optimization.

In subsequent research, efforts will be made to make the proposed algorithm smarter by selecting the appropriate methods from the available options based on the encountered data. Additionally, the algorithm parameters will be automatically optimized to improve its performance. Implementing a smart selection mechanism for pitch frequency computation algorithms will enable the algorithm to choose the most suitable approach based on the data characteristics. This enhancement is expected to increase the algorithm's performance at the expense of slightly longer computation times.

Moreover, a major future endeavor involves transferring the entire codebase to the Python environment. This transition will enable the comparison of the proposed approach with a larger set of existing methods. Additionally, a toolbox written in Python will facilitate the adoption of the proposed algorithm, making it more accessible and widely applicable. And another future work plan involves developing a time series anomaly detection pipeline that incorporates most benchmark anomaly detection methods, including the proposed algorithm. The Python repository will be available at "https://github.com/EErkus".

REFERENCES

[1] P. Flandrin. *Time-frequency/time-scale analysis*. Academic press, 1998.

[2] E. Keogh, S. Chu, D. Hart, and M. Pazzani. Segmenting time series: A survey and novel approach. In *Data mining in time series databases*, pages 1–21. World Scientific, 2004.

[3] P. Spirtes, C. N. Glymour, and R. Scheines. *Causation, prediction, and search*. MIT press, 2000.

[4] R. A. Yaffee and M. McGee. *An introduction to time series analysis and forecasting: with applications of SAS® and SPSS®*. Elsevier, 2000.

[5] R. Begg, D. T. Lai, and M. Palaniswami. *Computational intelligence in biomedical engineering*. CRC Press, 2007.

[6] J. S. Richman and J. R. Moorman. Physiological time-series analysis using approximate entropy and sample entropy. *American Journal of Physiology-Heart and Circulatory Physiology*, 2000.

[7] S. L. Hyland, M. Faltys, M. Hüser, X. Lyu, T. Gumbsch, C. Esteban, C. Bock, M. Horn, M. Moor, B. Rieck, et al. Early prediction of circulatory failure in the intensive care unit using machine learning. *Nature medicine*, 26(3):364–373, 2020.

[8] C. Cobelli and E. Carson. *Introduction to modeling in physiology and medicine*. Academic Press, 2019.

[9] S. R. Devasahayam. *Signals and systems in biomedical engineering: signal processing and physiological systems modeling*. Springer Science & Business Media, 2012.

[10] P. A. Abhang, B. Gawali, and S. C. Mehrotra. *Introduction to EEG-and speech-based emotion recognition*. Academic Press, 2016.

[11] R. Shiavi. *Introduction to applied statistical signal analysis: Guide to biomedical and electrical engineering applications*. Elsevier, 2010.

[12] S. S. Haykin. *Adaptive filter theory*. Pearson Education India, 2002.

[13] M. Shelhamer. *Nonlinear dynamics in physiology: a state-space approach*. World Scientific, 2007.

[14] W. Ebeling and I. Sokolov. *Statistical thermodynamics and stochastic theory of nonequilibrium systems*, volume 8. World Scientific Publishing Company, 2005.

[15] A. E. Gelfand, M. Fuentes, J. A. Hoeting, and R. L. Smith. *Handbook of environmental and ecological statistics*. CRC Press, 2019.

[16] R. A. M. Gregson. *Time series in psychology*. Psychology Press, 2014.

[17] S. R. McPherson. *Event based measurement and analysis of internet network traffic*. University of Southern California, 2011.

[18] E. Erkuş, V. Purutçuoğlu, and E. Purutçuoğlu. Detection of abnormalities in heart rate using multiple fourier transforms. *International Journal of Environmental Science and Technology*, pages 1–6, 2019.

[19] B. Lee, J. Han, H. J. Baek, J. H. Shin, K. S. Park, and W. J. Yi. Improved elimination of motion artifacts from a photoplethysmographic signal using a kalman smoother with simultaneous accelerometry. *Physiological measurement*, 31(12):1585, 2010.

[20] M. S. Tootooni, P. K. Rao, C.-A. Chou, and Z. J. Kong. A spectral graph theoretic approach for monitoring multivariate time series data from complex dynamical processes. *IEEE Transactions on Automation Science and Engineering*, 15(1):127–144, 2016.

[21] E. C. Erkuş and V. Purutçuoğlu. A new collective anomaly detection approach using pitch frequency and dissimilarity: Pitchy anomaly detection (pad). *Journal of Computational Science*, page 102084, 2023.

[22] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.

[23] H. Wang, F. Nie, H. Huang, S. L. Risacher, A. J. Saykin, L. Shen, and A. D. N. Initiative. Identifying disease sensitive and quantitative trait-relevant biomarkers from multidimensional heterogeneous imaging genetics data via sparse multimodal multitask learning. *Bioinformatics*, 28(12):i127–i136, 2012.

[24] S. Chauhan and L. Vig. Anomaly detection in ecg time signals via deep long short-term memory networks. In *2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 1–7. IEEE, 2015.

[25] A. Belle, R. Thiagarajan, S. Soroushmehr, F. Navidi, D. A. Beard, and K. Najarian. Big data analytics in healthcare. *BioMed research international*, 2015, 2015.

[26] S. Sanei, D. Jarchi, and A. G. Constantinides. *Body sensor networking, design and algorithms*. John Wiley & Sons, 2020.

[27] G. A. Kaplan and J. E. Keil. Socioeconomic factors and cardiovascular disease: a review of the literature. *Circulation*, 88(4):1973–1998, 1993.

[28] S. C. Hayes, K. G. Wilson, E. V. Gifford, V. M. Follette, and K. Strosahl. Experiential avoidance and behavioral disorders: A functional dimensional approach to diagnosis and treatment. *Journal of consulting and clinical psychology*, 64(6):1152, 1996.

[29] C. J. Boushey, A. M. Coulston, C. L. Rock, and E. Monsen. *Nutrition in the Prevention and Treatment of Disease*. Elsevier, 2001.

[30] G. D. Clifford, F. Azuaje, P. McSharry, et al. *Advanced methods and tools for ECG data analysis*. Artech house Boston, 2006.

[31] D. Dimoudis, T. Vafeiadis, A. Nizamis, D. Ioannidis, and D. Tzovaras. Utilizing an adaptive window rolling median methodology for time series anomaly detection. *Procedia Computer Science*, 217:584–593, 2023.

[32] P. J. Rousseeuw and A. M. Leroy. *Robust regression and outlier detection*. John wiley & sons, 2005.

[33] K. G. Mehrotra, C. K. Mohan, and H. Huang. *Anomaly detection principles and algorithms*. Springer, 2017.

[34] C. C. Aggarwal. *Outlier Analysis*. Springer, New York, 2013.

[35] M. O. Mansur, M. Noor, and M. Sap. Outlier detection technique in data mining: A research perspective. In *Proc. of the Postgrad. Annu. Res. Semin.*, pages 23–31, 2005.

[36] L. Akoglu, H. Tong, and D. Koutra. Graph based anomaly detection and description: a survey. *Data mining and knowledge discovery*, 29(3):626–688, 2015.

[37] T. Lane and C. E. Brodley. Temporal sequence learning and data reduction for anomaly detection. *ACM Transactions on Information and System Security (TISSEC)*, 2(3):295–331, 1999.

[38] R. A. A. Habeeb, F. Nasaruddin, A. Gani, I. A. T. Hashem, E. Ahmed, and M. Imran. Real-time big data processing for anomaly detection: A survey. *International Journal of Information Management*, 45:289–307, 2019.

[39] M. A. Samara, I. Bennis, A. Abouaissa, and P. Lorenz. A survey of outlier detection techniques in iot: review and classification. *Journal of Sensor and Actuator Networks*, 11(1):4, 2022.

[40] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3), 2009.

[41] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection for discrete sequences: A survey. *IEEE transactions on knowledge and data engineering*, 24(5):823–839, 2010.

[42] M. Gupta, J. Gao, C. C. Aggarwal, and J. Han. Outlier detection for temporal data: A survey. *IEEE Transactions on Knowledge and data Engineering*, 26(9):2250–2267, 2013.

[43] H. Ren, Z. Ye, and Z. Li. Anomaly detection based on a dynamic markov model. *Information Sciences*, 411:52–65, 2017.

[44] E. C. Erkuş and V. Purutçuoğlu. Outlier detection and quasi-periodicity optimization algorithm: Frequency domain based outlier detection (fod). *European Journal of Operational Research*, 291(2):560–574, 2021.

[45] X. Song, M. Wu, C. Jermaine, and S. Ranka. Conditional anomaly detection. *IEEE Transactions on knowledge and Data Engineering*, 19(5):631–645, 2007.

[46] H. Zenati, M. Romain, C.-S. Foo, B. Lecouat, and V. Chandrasekhar. Adversarially learned anomaly detection. In *2018 IEEE International conference on data mining (ICDM)*, pages 727–736. IEEE, 2018.

[47] N. Laptev, S. Amizadeh, and I. Flint. Generic and scalable framework for automated time-series anomaly detection. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1939–1947, 2015.

[48] V. Hodge and J. Austin. A survey of outlier detection methodologies. *Artif. Intell. Rev.*, 22(2):85–126, 2004.

[49] P.-N. Tan. *Introduction to data mining*. Pearson Education India, 2018.

[50] A. Ukil, S. Bandyoapdhyay, C. Puri, and A. Pal. Iot healthcare analytics: The importance of anomaly detection. In *2016 IEEE 30th international conference on advanced information networking and applications (AINA)*, pages 994–997. IEEE, 2016.

[51] P. Gogoi, D. K. Bhattacharyya, B. Borah, and J. K. Kalita. A survey of outlier detection methods in network anomaly identification. *The Computer Journal*, 54(4):570–588, 2011.

[52] V. Chandola, A. Banerjee, and V. Kumar. Outlier detection: A survey. *ACM Computing Surveys*, 14:15, 2007.

[53] K. Singh and S. Upadhyaya. Outlier detection: applications and techniques. *International Journal of Computer Science Issues (IJCSI)*, 9(1):307, 2012.

[54] S. Basu and M. Meckesheimer. Automatic outlier detection for time series: an application to sensor data. *Knowledge and Information Systems*, 11(2):137–154, 2007.

[55] M. Gupta, J. Gao, C. Aggarwal, and J. Han. Outlier detection for temporal data. *Synthesis Lectures on Data Mining and Knowledge Discovery*, 5(1):1–129, 2014.

[56] F. Strasser, M. Muma, and A. M. Zoubir. Motion artifact removal in ecg signals using multi-resolution thresholding. In *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO)*, pages 899–903. IEEE, 2012.

[57] Z. Zhao, Y. Zhang, Z. Comert, and Y. Deng. Computer-aided diagnosis system of fetal hypoxia incorporating recurrence plot with convolutional neural network. *Frontiers in physiology*, 10:255, 2019.

[58] K. H. Ong, D. Ramachandram, R. Mandava, and I. L. Shuaib. Automatic white matter lesion segmentation using an adaptive outlier detection method. *Magnetic resonance imaging*, 30(6):807–823, 2012.

[59] M. Prastawa, E. Bullitt, S. Ho, and G. Gerig. A brain tumor segmentation framework based on outlier detection. *Medical image analysis*, 8(3):275–283, 2004.

[60] J. Lee, D. D. McManus, S. Merchant, and K. H. Chon. Automatic motion and noise artifact detection in holter ecg data using empirical mode decomposition and statistical approaches. *IEEE Transactions on Biomedical Engineering*, 59(6):1499–1506, 2011.

[61] A. R. Magnano, S. Holleran, R. Ramakrishnan, J. A. Reiffel, and D. M. Bloomfield. Autonomic modulation of the u wave during sympathomimetic stimulation and vagal inhibition in normal individuals. *Pacing and clinical electrophysiology*, 27(11):1484–1492, 2004.

[62] C. C. Aggarwal and S. Sathe. *Outlier Ensembles: An Introduction*. Springer, Berlin, 2017.

[63] N. Heim and J. E. Avery. Adaptive anomaly detection in chaotic time series with a spatially aware echo state network. *arXiv preprint arXiv:1909.01709*, 2019.

[64] G. M. Ljung and G. E. Box. On a measure of lack of fit in time series models. *Biometrika*, 65(2):297–303, 1978.

[65] A. Sagoolmuang and K. Sinapiromsaran. Median-difference window subseries score for contextual anomaly on time series. In *2017 8th International Conference of Information and Communication Technology for Embedded Systems (IC-ICTES)*, pages 1–6. IEEE, 2017.

[66] M. Munir, S. Erkel, A. Dengel, and S. Ahmed. Pattern-based contextual anomaly detection in hvac systems. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 1066–1073. IEEE, 2017.

[67] J. Wang, R. Li, R. Li, and B. Fu. A knowledge-based deep learning method for ecg signal delineation. *Future Generation Computer Systems*, 109:56–66, 2020.

[68] C. Dora and P. K. Biswal. Robust ecg artifact removal from eeg using continuous wavelet transformation and linear regression. In *2016 International Conference on Signal Processing and Communications (SPCOM)*, pages 1–5. IEEE, 2016.

[69] J. Ye, T. Kobayashi, M. Murakawa, T. Higuchi, and N. Otsu. Anomaly detection using multi-channel flac for supporting diagnosis of ecg. *IEEJ Transactions on Electronics, Information and Systems*, 132(1):111–119, 2012.

[70] J. R. Pinto, J. S. Cardoso, A. Lourenço, and C. Carreiras. Towards a continuous biometric system based on ecg signals acquired on the steering wheel. *Sensors*, 17(10):2228, 2017.

[71] Z. Chen and Y. F. Li. Anomaly detection based on enhanced dbscan algorithm. *Procedia Engineering*, 15:178–182, 2011.

[72] S. Elmougy, M. S. Hossain, A. S. Tolba, M. F. Alhamid, and G. Muhammad. A parameter based growing ensemble of self-organizing maps for outlier detection in healthcare. *Cluster Computing*, 22(1):2437–2460, 2019.

[73] R. Vecht, M. A. Gatzoulis, and N. Peters. *ECG diagnosis in clinical practice*. Springer Science & Business Media, 2009.

[74] R. Chalapathy and S. Chawla. Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407*, 2019.

[75] F. Nolle, F. Badura, J. Catlett, R. Bowser, and M. Sketch. Crei-gard, a new concept in computerized arrhythmia monitoring systems. *Computers in Cardiology*, 13(1):515–518, 1986.

[76] O. Salem, A. Guerassimov, A. Mehaoua, A. Marcus, and B. Furht. Anomaly detection in medical wireless sensor networks using svm and linear regression models. *International Journal of E-Health and Medical Communications (IJEHMC)*, 5(1):20–45, 2014.

[77] A. Goshvarpour and A. Goshvarpour. Poincaré's section analysis for ppg-based automatic emotion recognition. *Chaos, Solitons & Fractals*, 114:400–407, 2018.

[78] T. Charrad, K. Nouira, and A. Ferchichi. Ecg patch monitor: A telemedicine system for remote monitoring and assisting patients during a heart attack. *International Journal of Ad Hoc and Ubiquitous Computing*, 34(1):25–34, 2020.

[79] A. Artemov, E. Burnaev, and A. Lokot. Nonparametric decomposition of quasi-periodic time series for change-point detection. In *Eighth International Conference on Machine Vision (ICMV 2015)*, volume 9875, pages 418–422. SPIE, 2015.

[80] A. Özmen, G. W. Weber, İ. Batmaz, and E. Kropat. Rcmars: Robustification of cmars with different scenarios under polyhedral uncertainty set. *Communications in Nonlinear Science and Numerical Simulation*, 16(12):4780–4787, 2011.

[81] S. A. Haque, M. Rahman, and S. M. Aziz. Sensor anomaly detection in wireless sensor networks for healthcare. *Sensors*, 15(4):8764–8786, 2015.

[82] O. Salem, Y. Liu, A. Mehaoua, and R. Boutaba. Online anomaly detection in wireless body area networks for reliable healthcare monitoring. *IEEE journal of biomedical and health informatics*, 18(5):1541–1551, 2014.

[83] O. Salem, A. Guerassimov, A. Mehaoua, A. Marcus, and B. Furht. Sensor fault and patient anomaly detection and classification in medical wireless sensor networks. In *2013 IEEE international conference on communications (ICC)*, pages 4373–4378. IEEE, 2013.

[84] M. Braei and S. Wagner. Anomaly detection in univariate time-series: A survey on the state-of-the-art. *arXiv preprint arXiv:2004.00433*, 2020.

[85] R. Refinetti, G. Cornélissen, and F. Halberg. Procedures for numerical analysis of circadian rhythms. *Biological rhythm research*, 38(4):275–325, 2007.

[86] K. Choi, J. Yi, C. Park, and S. Yoon. Deep learning for anomaly detection in time-series data: review, analysis, and guidelines. *IEEE Access*, 9:120043–120065, 2021.

[87] H. Banaee, M. U. Ahmed, and A. Loutfi. Data mining for wearable sensors in health monitoring systems: a review of recent trends and challenges. *Sensors*, 13(12):17472–17500, 2013.

[88] V. Hodge and J. Austin. A survey of outlier detection methodologies. *Artificial intelligence review*, 22(2):85–126, 2004.

[89] E. Kropat, A. Özmen, G.-W. Weber, S. Meyer-Nieberg, and O. Defterli. Fuzzy prediction strategies for gene-environment networks–fuzzy regression analysis for two-modal regulatory systems. *RAIRO-Operations Research-Recherche Opérationnelle*, 50(2):413–435, 2016.

[90] H. P. Kriegel, P. Kroger, and A. Zimek. Outlier detection techniques. *Tut. at KDD*, 10, 2010.

[91] Y. Zhang, N. Meratnia, and J. M. P. Havinga. Outlier detection techniques for wireless sensor networks: A survey. *IEEE Comm. Surv. and Tut.*, 12(2):159–170, 2010.

[92] I. Ben-Gal. *Outlier Detection in Data Mining and Knowledge Discovery Handbook: A Complete Guide for Practitioners and Researchers*. Dordrecht, Kluwer Acad. Publ., 2005.

[93] F. Grubbs. Procedures for detecting outlying observations in samples. *Technometrics*, 11(1):1–21, 1969.

[94] S. Seo. *A review and comparison of methods for detecting outliers in univariate data sets*. PhD thesis, Univ. of Pittsburgh, 2006.

[95] G. Williams, R. Baxter, H. He, S. Hawkins, and L. Gu. A comparative study of rnn for outlier detection in data mining. In *2002 IEEE Int. Conf. on Data Min.*, pages 709–712. IEEE, 2002.

[96] A. Z. G. O. Campos, J. Sander, R. J. G. B. Campello, B. Micenkova, E. Schubert, I. Assent, and M. E. Houle. On the evaluation of unsupervised outlier detection: Measures, datasets, and an empirical study. *Data Min. and Knowl. Disco.*, 30(4):891–927, 2016.

[97] J. W. Tukey. *Exploratory Data Analysis*, volume 1. Addison-Wesley Publ. Company, 1977.

[98] P. Indyk, N. Koudas, and S. Muthukrishnan. Identifying representative trends in massive time series data sets using sketches. In *VLDB*, pages 363–372, 2000.

[99] F. Rasheed, M. Alshalalfa, and R. Alhajj. Efficient periodicity mining in time series databases using suffix trees. *IEEE Trans. Knowl. Data Eng.*, 23(1):79–94, 2011.

[100] F. Rasheed and R. Alhajj. A framework for periodic outlier pattern detection in time-series sequences. *IEEE Trans. on Cybern.*, 44(5):569–582, 2014.

[101] J. Lee, C. Jin, Z. Liu, and H. D. Ardakani. Introduction to data-driven methodologies for prognostics and health management. In *Probabilistic prognostics and health management of energy systems*, pages 9–32. Springer, 2017.

[102] X. Gu and P. Angelov. Autonomous anomaly detection. In *2017 Evolving and Adaptive Intelligent Systems (EAIS)*, pages 1–8. IEEE, 2017.

[103] G. L. Tietjen and R. H. Moore. Some grubbs-type statistics for the detection of several outliers. *Technometrics*, 14(3):583–597, 1972.

[104] F. Rasheed, P. Peng, R. Alhajj, and J. Rokne. Fourier transform based spatial outlier mining. In *Int. Conf. on Intell. Data Eng. and Autom. Learn.*, pages 317–324. Springer, 2009.

[105] O. Shittu and D. Shangodoyin. Detection of outliers in time series data: A frequency domain approach. *Asian J. of Sci. Res.*, 1(1):130–137, 2008.

[106] W. Hu and J. Bao. The outlier interval detection algorithms on astronautical time series data. *Math. Prob. in Eng.*, 2013:6, 2013.

[107] G. Tang, K. Wu, J. Lei, Z. Bi, and J. Tang. From landscape to portrait: A new approach for outlier detection in load curve data. *IEEE Trans. on Small Grid*, 5(4):1764–1773, 2014.

[108] M. Aouf and L. A. F. Park. Approximate document outlier detection using random spectral projection. *AI 2012: Adv. in Artif. Intell.*, 7691:579–590, 2012.

[109] X. Bao and L. Dai. Partial least squares with outlier detection in spectral analysis: A tool to predict gasoline properties. *Fuel*, 88(7):1216–1222, 2009.

[110] M. G. Elfeky, W. G. Aref, and A. K. Elmagarmid. Warp: time warping for periodicity detection. In *5th IEEE Int. Conf. on Data Min. (ICDM'05)*, pages 8–pp. IEEE, 2005.

[111] M. Goldstein and S. Uchida. Comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PloS One*, 11(4), 2016.

[112] K. Worden, C. R. Farrar, G. Manson, and G. Park. The fundamental axioms of structural health monitoring. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 463(2082):1639–1664, 2007.

[113] L. Formaggia, A. Quarteroni, and A. Veneziani. *Cardiovascular Mathematics: Modeling and simulation of the circulatory system*, volume 1. Springer Science & Business Media, 2010.

[114] A. El Attaoui, M. Hazmi, A. Jilbab, and A. Bourouhou. Wearable wireless sensors network for ecg telemonitoring using neural network for features extraction. *Wireless Personal Communications*, 111(3):1955–1976, 2020.

[115] M. Zhao, A. Jha, Q. Liu, B. A. Millis, A. Mahadevan-Jansen, L. Lu, B. A. Landman, M. J. Tyska, and Y. Huo. Faster mean-shift: Gpu-accelerated clustering for cosine embedding-based cell segmentation and tracking. *Medical Image Analysis*, 71:102048, 2021.

[116] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *International conference on information processing in medical imaging*, pages 146–157. Springer, 2017.

[117] R. Wu and E. Keogh. Current time series anomaly detection benchmarks are flawed and are creating the illusion of progress. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[118] B. Veeravalli, C. J. Deepu, and D. Ngo. Real-time, personalized anomaly detection in streaming data for wearable healthcare devices. In *Handbook of large-scale distributed computing in smart healthcare*, pages 403–426. Springer, 2017.

[119] B. Kim, M. A. Alawami, E. Kim, S. Oh, J. Park, and H. Kim. A comparative study of time series anomaly detection models for industrial control systems. *Sensors*, 23(3):1310, 2023.

[120] A. Mason, Y. Zhao, H. He, R. Gompelman, and S. Mandava. Online anomaly detection of time series at scale. In *2019 International Conference on Cyber Situational Awareness, Data Analytics And Assessment (Cyber SA)*, pages 1–8. IEEE, 2019.

[121] A. Anandakrishnan, S. Kumar, A. Statnikov, T. Faruquie, and D. Xu. Anomaly detection in finance: editors' introduction. In *KDD 2017 Workshop on Anomaly Detection in Finance*, pages 1–7. PMLR, 2018.

[122] A. E. Glaser, J. P. Harrison, and D. Josephs. Anomaly detection methods to improve supply chain data quality and operations. *SMU Data Science Review*, 6(1):3, 2022.

[123] R. Oucheikh, M. Fri, F. Fedouaki, and M. Hain. Deep real-time anomaly

detection for connected autonomous vehicles. *Procedia Computer Science*, 177:456–461, 2020.

[124] G. Fenza, M. Gallo, and V. Loia. Drift-aware methodology for anomaly detection in smart grid. *IEEE Access*, 7:9645–9657, 2019.

[125] A. Z. Woldaregay, E. Årsand, T. Botsis, D. Albers, L. Mamykina, and G. Hartvigsen. Data-driven blood glucose pattern classification and anomalies detection: machine-learning applications in type 1 diabetes. *Journal of medical Internet research*, 21(5):e11030, 2019.

[126] M. Nawaz and J. Ahmed. Cloud-based healthcare framework for real-time anomaly detection and classification of 1-d ecg signals. *Plos one*, 17(12):e0279305, 2022.

[127] X. Sun, C. Zhang, and L. Li. Dynamic emotion modelling and anomaly detection in conversation based on emotional transition tensor. *Information Fusion*, 46:11–22, 2019.

[128] F. Cui, Y. Yue, Y. Zhang, Z. Zhang, and H. S. Zhou. Advancing biosensors with machine learning. *ACS sensors*, 5(11):3346–3364, 2020.

[129] S. Saba-Sadiya, E. Chantland, T. Alhanai, T. Liu, and M. M. Ghassemi. Unsupervised eeg artifact detection and correction. *Frontiers in Digital Health*, 2:608920, 2021.

[130] M. E. Tschuchnig and M. Gadermayr. Anomaly detection in medical imaging-a mini review. In *Data Science–Analytics and Applications: Proceedings of the 4th International Data Science Conference–iDSC2021*, pages 33–38. Springer, 2022.

[131] M. Mahmud, M. S. Kaiser, T. M. McGinnity, and A. Hussain. Deep learning in mining biological data. *Cognitive Computation*, 13(1):1–33, 2021.

[132] S. Agrawal and J. Agrawal. Survey on anomaly detection using data mining techniques. *Procedia Computer Science*, 60:708–713, 2015.

[133] X. Zhang and H. Huang. A real-time, practical sensor fault-tolerant module

for robust emg pattern recognition. *Journal of neuroengineering and rehabilitation*, 12(1):1–16, 2015.

[134] D. Wulsin, J. Blanco, R. Mani, and B. Litt. Semi-supervised anomaly detection for eeg waveforms using deep belief nets. In *2010 Ninth international conference on machine learning and applications*, pages 436–441. IEEE, 2010.

[135] J. R. Pinto, J. S. Cardoso, and A. Lourenço. Evolution, current challenges, and future possibilities in ecg biometrics. *IEEE Access*, 6:34746–34776, 2018.

[136] K. Al-Jabery, T. Obafemi-Ajayi, G. Olbricht, and D. Wunsch. *Computational learning approaches to data analytics in biomedical applications*. Academic Press, 2019.

[137] V. López, A. Fernández, S. García, V. Palade, and F. Herrera. An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics. *Information sciences*, 250:113–141, 2013.

[138] J. Lafferty, A. McCallum, and F. C. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

[139] M. A. Hall. Correlation-based feature selection for machine learning. 1999.

[140] E. C. Erkus and V. Purutcuoglu. A new frequency domain and dynamic time warping based feature: Wfod feature. In *AIP Conference Proceedings*, volume 2714. AIP Publishing, 2023.

[141] P. Bruce, A. Bruce, and P. Gedeck. *Practical statistics for data scientists: 50+ essential concepts using R and Python*. O'Reilly Media, 2020.

[142] M. Barandas, D. Folgado, L. Fernandes, S. Santos, M. Abreu, P. Bota, H. Liu, T. Schultz, and H. Gamboa. Tsfel: Time series feature extraction library. *SoftwareX*, 11:100456, 2020.

[143] H. Chen, S. Das, J. M. Morgan, and K. Maharatna. Prediction and classification of ventricular arrhythmia based on phase-space reconstruction and fuzzy c-means clustering. *Computers in Biology and Medicine*, 142:105180, 2022.

[144] W. Bomela, S. Wang, C.-A. Chou, and J.-S. Li. Real-time inference and detection of disruptive eeg networks for epileptic seizures. *Scientific Reports*, 10(1):1–10, 2020.

[145] E. C. Erkuş and V. Purutçuoğlu. Feature extraction of hidden oscillation in ecg data via multiple-fod method. In *Artificial Intelligence and Applied Mathematics in Engineering Problems: Proceedings of the International Conference on Artificial Intelligence and Applied Mathematics in Engineering (ICAIAME 2019)*, pages 47–56. Springer, 2020.

[146] S. Aghabozorgi, A. S. Shirkhorshidi, and T. Y. Wah. Time-series clustering–a decade review. *Information systems*, 53:16–38, 2015.

[147] S.-H. Cha. Comprehensive survey on distance/similarity measures between probability density functions. *City*, 1(2):1, 2007.

[148] A. Blázquez-García, A. Conde, U. Mori, and J. A. Lozano. A review on outlier/anomaly detection in time series data. *ACM Computing Surveys (CSUR)*, 54(3):1–33, 2021.

[149] Á. López-Oriona and J. A. Vilar. Outlier detection for multivariate time series: A functional data approach. *Knowledge-Based Systems*, 233:107527, 2021.

[150] V. Rasoulzadeh, E. Erkus, T. Yogurt, I. Ulusoy, and S. A. Zergeroğlu. A comparative stationarity analysis of eeg signals. *Annals of Operations Research*, 258:133–157, 2017.

[151] M. Fahim and A. Sillitti. Anomaly detection, analysis and prediction techniques in iot environment: A systematic literature review. *IEEE Access*, 7:81664–81681, 2019.

[152] K. Fujiwara, E. Abe, K. Kamata, C. Nakayama, Y. Suzuki, T. Yamakawa, T. Hiraoka, M. Kano, Y. Sumi, F. Masuda, et al. Heart rate variability-based driver drowsiness detection and its validation with eeg. *IEEE transactions on biomedical engineering*, 66(6):1769–1778, 2018.

[153] Y. Sun, J. Li, J. Liu, C. Chow, B. Sun, and R. Wang. Using causal discovery for feature selection in multivariate numerical time series. *Machine Learning*, 101:377–395, 2015.

[154] J. E. Ball, D. T. Anderson, and C. S. Chan. Comprehensive survey of deep learning in remote sensing: theories, tools, and challenges for the community. *Journal of applied remote sensing*, 11(4):042609–042609, 2017.

[155] J. Du, F. Chen, and Y.-M. Hu. Automatic defect inspection of patterned fpc board based on 1-d fourier reconstruction. In *2017 36th Chinese Control Conf. (CCC)*, pages 10109–10112. IEEE, 2017.

[156] B. Yegnanarayana and K. S. R. Murty. Event-based instantaneous fundamental frequency estimation from speech signals. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(4):614–624, 2009.

[157] M. Lahat, R. Niederjohn, and D. Krubsack. A spectral autocorrelation method for measurement of the fundamental frequency of noise-corrupted speech. *IEEE transactions on acoustics, speech, and signal processing*, 35(6):741–750, 1987.

[158] B. C. Moore. *An introduction to the psychology of hearing*. Brill, 2012.

[159] W. Palma. *Time series analysis*. John Wiley & Sons, 2016.

[160] J. S. Bendat and A. G. Piersol. *Random data: analysis and measurement procedures*, volume 729. John Wiley & Sons, 2011.

[161] P. M. Clarkson. *Optimal and adaptive signal processing*. Routledge, 2017.

[162] C. Cheng, A. Sa-Ngasoongsong, O. Beyca, T. Le, H. Yang, Z. Kong, and S. T. Bukkapatnam. Time series forecasting for nonlinear and non-stationary processes: a review and comparative study. *Iie Transactions*, 47(10):1053–1071, 2015.

[163] M. Aktaruzzaman and R. Sassi. Parametric estimation of sample entropy in heart rate variability analysis. *Biomedical Signal Processing and Control*, 14:141–147, 2014.

[164] K. C. Chua, V. Chandran, U. R. Acharya, and C. M. Lim. Application of higher order statistics/spectra in biomedical signals—a review. *Medical engineering & physics*, 32(7):679–689, 2010.

[165] S. Saxena, V. K. Gupta, and P. Hrisheekesha. Coronary heart disease detection using nonlinear features and online sequential extreme learning machine. *Biomedical Engineering: Applications, Basis and Communications*, 31(06):1950046, 2019.

[166] R. M. Rangayyan. *Biomedical signal analysis*. John Wiley & Sons, 2015.

[167] B. M. Quandt, L. J. Scherer, L. F. Boesel, M. Wolf, G.-L. Bona, and R. M. Rossi. Body-monitoring and health supervision by means of optical fiber-based sensing systems in medical textiles. *Advanced healthcare materials*, 4(3):330–355, 2015.

[168] A. C. Skanes, R. Mandapati, O. Berenfeld, J. M. Davidenko, and J. Jalife. Spatiotemporal periodicity during atrial fibrillation in the isolated sheep heart. *Circulation*, 98(12):1236–1248, 1998.

[169] C. A. Taylor and C. Figueroa. Patient-specific modeling of cardiovascular mechanics. *Annual review of biomedical engineering*, 11:109–134, 2009.

[170] M. Zivanovic and M. Gonzalez-Izal. Quasi-periodic modeling for heart sound localization and suppression in lung sounds. *Biomedical Signal Processing and Control*, 8(6):586–595, 2013.

[171] P. Ivaturi, M. Gadaleta, A. C. Pandey, M. Pazzani, S. R. Steinhubl, and G. Quer. A comprehensive explanation framework for biomedical time series classification. *IEEE journal of biomedical and health informatics*, 25(7):2398–2408, 2021.

[172] G. D. Clifford, F. Azuaje, and P. Mcsharry. Ecg statistics, noise, artifacts, and missing data. *Advanced methods and tools for ECG data analysis*, 6(1):18, 2006.

[173] M. V. Kamath, M. Watanabe, and A. Upton. Heart rate variability (hrv) signal analysis: clinical applications. 2012.

[174] H. Adeli and S. Ghosh-Dastidar. *Automated EEG-based diagnosis of neurological disorders: Inventing the future of neurology*. CRC press, 2010.

[175] J. G. Webster. *Electrical measurement, signal processing, and displays*. CRC Press, 2003.

[176] L. Tan and J. Jiang. *Digital signal processing: fundamentals and applications*. Academic Press, 2018.

[177] L. Sörnmo and P. Laguna. *Bioelectrical signal processing in cardiac and neurological applications*, volume 8. Academic press, 2005.

[178] G. Li, Y. Li, L. Yu, and Y. Geng. Conditioning and sampling issues of emg signals in motion recognition of multifunctional myoelectric prostheses. *Annals of biomedical engineering*, 39:1779–1787, 2011.

[179] L. Mesin. A neural algorithm for the non-uniform and adaptive sampling of biomedical data. *Computers in Biology and Medicine*, 71:223–230, 2016.

[180] S. K. Berkaya, A. K. Uysal, E. S. Gunal, S. Ergin, S. Gunal, and M. B. Gulmezoglu. A survey on ecg analysis. *Biomedical Signal Processing and Control*, 43:216–235, 2018.

[181] B. Boashash. *Time-frequency signal analysis and processing: a comprehensive reference*. Academic press, 2015.

[182] J. B. J. Fourier. *Théorie analytique de la chaleur*. Gauthier-Villars et fils, 1888.

[183] R. L. Allen and D. Mills. *Signal analysis: time, frequency, scale, and structure*. John Wiley & Sons, 2004.

[184] A. Ishimaru. *Electromagnetic wave propagation, radiation, and scattering: from fundamentals to applications*. John Wiley & Sons, 2017.

[185] S. V. Vaseghi. *Advanced digital signal processing and noise reduction*. John Wiley & Sons, 2008.

[186] M. Akay. *Biomedical signal processing*. Academic press, 2012.

[187] B. Chandrakar, O. Yadav, and V. Chandra. A survey of noise removal techniques for ecg signals. *International Journal of Advanced Research in Computer and Communication Engineering*, 2(3):1354–1357, 2013.

[188] K. Afifah and N. Retdian. Design of n-path notch filter circuits for hum noise suppression in biomedical signal acquisition. *IEICE Transactions on Electronics*, 103(10):480–488, 2020.

[189] E. R. Davies. *Machine vision: theory, algorithms, practicalities*. Elsevier, 2004.

[190] G. Sharma, K. Umapathy, and S. Krishnan. Trends in audio signal feature extraction methods. *Applied Acoustics*, 158:107020, 2020.

[191] H.-T. Wu. Current state of nonlinear-type time–frequency analysis and applications to high-frequency biomedical signals. *Current Opinion in Systems Biology*, 23:8–21, 2020.

[192] U. Satija, B. Ramkumar, and M. S. Manikandan. A review of signal processing techniques for electrocardiogram signal quality assessment. *IEEE reviews in biomedical engineering*, 11:36–52, 2018.

[193] P. Banerjee, B. Chakraborty, and J. Banerjee. Procedure for cepstral analysis in tracing unique voice segments. In *2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom)*, pages 351–356. IEEE, 2015.

[194] J. Jeong. Kepstrum analysis and real-time application to noise cancellation. In *Proceedings of the 8th WSEAS International Conference on Signal Processing, Robotics, and Automation*, pages 149–154, 2009.

[195] D. Goyal and B. Pabla. The vibration monitoring methods and signal processing techniques for structural health monitoring: a review. *Archives of Computational Methods in Engineering*, 23:585–594, 2016.

[196] J. Deng, S. Frühholz, Z. Zhang, and B. Schuller. Recognizing emotions from whispered speech based on acoustic feature transfer learning. *IEEE Access*, 5:5235–5246, 2017.

[197] A. Gabrielsson. The performance of music. In *The psychology of music*, pages 501–602. Elsevier, 1999.

[198] S. O. Sadjadi, S. M. Ahadi, and O. Hazrati. Unsupervised speech/music classification using one-class support vector machines. In *2007 6th International Conference on Information, Communications & Signal Processing*, pages 1–5. IEEE, 2007.

[199] K. R. Anne, S. Kuchibhotla, and H. D. Vankayalapati. *Acoustic modeling for emotion recognition*. Springer, 2015.

[200] F. Tamburini and C. Caini. An automatic system for detecting prosodic prominence in american english continuous speech. *International Journal of speech technology*, 8(1):33–44, 2005.

[201] J.-R. Yeh, J.-S. Shieh, and N. E. Huang. Complementary ensemble empirical mode decomposition: A novel noise enhanced data analysis method. *Advances in adaptive data analysis*, 2(02):135–156, 2010.

[202] A. R. Allam, A. S. Ashour, M. M. Abd Elnaby, and F. E. Abd El-Samie. A novel pitch-frequency-based ecg signal classification approach for abnormality detection. In *2019 7th International Japan-Africa Conference on Electronics, Communications, and Computations,(JAC-ECC)*, pages 106–110. IEEE, 2019.

[203] MathWorks®. pitch: Estimate fundamental frequency of audio signal.

[204] B. S. Atal. Automatic speaker recognition based on pitch contours. *The Journal of the Acoustical Society of America*, 52(6B):1687–1697, 1972.

[205] D. Wang, Y. Wei, Y. Wang, and J. Wang. A robust and low computational cost pitch estimation method. *Sensors*, 22(16):6026, 2022.

[206] B. Ma, C. Van Doorne, Z. Zhang, and F. Nieuwstadt. On the spatial evolution of a wall-imposed periodic disturbance in pipe poiseuille flow at re= 3000. part 1. subcritical disturbance. *Journal of Fluid Mechanics*, 398:181–224, 1999.

[207] S. Gonzalez and M. Brookes. A pitch estimation filter robust to high levels of noise (pefac). In *2011 19th European Signal Processing Conference*, pages 451–455. IEEE, 2011.

[208] S. Gonzalez and M. Brookes. Pefac-a pitch estimation algorithm robust to high

levels of noise. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(2):518–530, 2014.

[209] S. Sarangi, M. Sahidullah, and G. Saha. Optimization of data-driven filterbank for automatic speaker verification. *Digital Signal Processing*, 104:102795, 2020.

[210] A. M. Noll. Cepstrum pitch determination. *The journal of the acoustical society of America*, 41(2):293–309, 1967.

[211] H. Kobayashi and T. Shimamura. A modified cepstrum method for pitch extraction. In *IEEE. APCCAS 1998. 1998 IEEE Asia-Pacific Conference on Circuits and Systems. Microelectronics and Integrating Systems. Proceedings (Cat. No. 98EX242)*, pages 299–302. IEEE, 1998.

[212] A. V. Oppenheim and R. W. Schafer. From frequency to quefrency: A history of the cepstrum. *IEEE signal processing Magazine*, 21(5):95–106, 2004.

[213] F. Huang and T. Lee. Pitch estimation in noisy speech based on temporal accumulation of spectrum peaks. In *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[214] D. J. Hermes. Measurement of pitch by subharmonic summation. *The journal of the acoustical society of America*, 83(1):257–264, 1988.

[215] Z. Lei. *Voice detection and pattern recognition using neck skin vibration signals*. McGill University (Canada), 2019.

[216] A. Huang, P. Sévigny, B. Balaji, and S. Rajan. Fundamental frequency estimation of herm lines of drones. In *2020 IEEE International Radar Conference (RADAR)*, pages 1013–1018. IEEE, 2020.

[217] P. Klaer, A. Huang, P. Sévigny, S. Rajan, S. Pant, P. Patnaik, and B. Balaji. An investigation of rotary drone herm line spectrum under manoeuvering conditions. *Sensors*, 20(20):5940, 2020.

[218] T. Drugman and A. Alwan. Joint robust voicing detection and pitch estimation based on residual harmonics. *arXiv preprint arXiv:2001.00459*, 2019.

[219] S. Finkelstein, S. Scherer, A. Ogan, L.-P. Morency, and J. Cassell. Investigating the influence of virtual peers as dialect models on students' prosodic inventory. In *Third Workshop on Child, Computer and Interaction*, 2012.

[220] A. S. Shirkhorshidi, S. Aghabozorgi, and T. Y. Wah. A comparison study on similarity and dissimilarity measures in clustering continuous data. *PloS one*, 10(12):e0144059, 2015.

[221] G. Xu, Y. Zong, and Z. Yang. *Applied data mining*. CRC Press, 2013.

[222] S. Ramaswamy, R. Rastogi, and K. Shim. Efficient algorithms for mining outliers from large data sets. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 427–438, 2000.

[223] I. E. Frank and R. Todeschini. *The data analysis handbook*. Elsevier, 1994.

[224] T. Thomas, A. P. Vijayaraghavan, and S. Emmanuel. *Machine learning approaches in cyber security analytics*. Springer, 2020.

[225] S. Salvador and P. Chan. Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis*, 11(5):561–580, 2007.

[226] M. Müller. Dynamic time warping. *Information retrieval for music and motion*, pages 69–84, 2007.

[227] M. Schmidt, M. Baumert, A. Porta, H. Malberg, and S. Zaunseder. Two-dimensional warping for one-dimensional signals—conceptual framework and application to ecg processing. *IEEE Transactions on Signal Processing*, 62(21):5577–5588, 2014.

[228] D. F. Silva and G. E. Batista. Speeding up all-pairwise dynamic time warping matrix calculation. In *Proceedings of the 2016 SIAM International Conference on Data Mining*, pages 837–845. SIAM, 2016.

[229] P. Senin. Dynamic time warping algorithm review. *Information and Computer Science Department University of Hawaii at Manoa Honolulu, USA*, 855(1-23):40, 2008.

[230] H. Madsen. *Time series analysis*. CRC Press, 2007.

[231] G. Ciaburro and G. Iannace. Machine learning-based algorithms to knowledge extraction from time series data: A review. *Data*, 6(6):55, 2021.

[232] J. M. De Sa. *Pattern recognition: concepts, methods, and applications*. Springer Science & Business Media, 2001.

[233] H. Ombao and M. Pinto. Spectral dependence. *Econometrics and Statistics*, 2022.

[234] A. K. Tangirala. *Principles of system identification: theory and practice*. Crc Press, 2018.

[235] S. L. Mirtaheri and R. Shahbazian. *Machine Learning: Theory to Applications*. CRC Press, 2022.

[236] A. K. Jain. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666, 2010.

[237] G. Bonaccorso. *Machine learning algorithms*. Packt Publishing Ltd, 2017.

[238] A. Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. " O'Reilly Media, Inc.", 2022.

[239] V. Kotu and B. Deshpande. *Data science: concepts and practice*. Morgan Kaufmann, 2018.

[240] A. Saxena, M. Prasad, A. Gupta, N. Bharill, O. P. Patel, A. Tiwari, M. J. Er, W. Ding, and C.-T. Lin. A review of clustering techniques and developments. *Neurocomputing*, 267:664–681, 2017.

[241] B. Mirkin. *Clustering: a data recovery approach*. CRC press, 2012.

[242] P. J. García-Laencina, J.-L. Sancho-Gómez, and A. R. Figueiras-Vidal. Pattern classification with missing data: a review. *Neural Computing and Applications*, 19:263–282, 2010.

[243] C. C. Aggarwal and C. C. Aggarwal. *An introduction to outlier analysis*. Springer, 2017.

[244] R. P. Duin and E. Pekalska. *Dissimilarity Representation For Pattern Recognition, The: Foundations And Applications*, volume 64. World scientific, 2005.

[245] A. Telikani, A. Tahmassebi, W. Banzhaf, and A. H. Gandomi. Evolutionary machine learning: A survey. *ACM Computing Surveys (CSUR)*, 54(8):1–35, 2021.

[246] Y. Li and X. Zhang. Improving k nearest neighbor with exemplar generalization for imbalanced classification. In *Advances in Knowledge Discovery and Data Mining: 15th Pacific-Asia Conference, PAKDD 2011, Shenzhen, China, May 24-27, 2011, Proceedings, Part II 15*, pages 321–332. Springer, 2011.

[247] P. Dangeti. *Statistics for machine learning*. Packt Publishing Ltd, 2017.

[248] O. Z. Maimon and L. Rokach. *Data mining with decision trees: theory and applications*, volume 81. World scientific, 2014.

[249] S. B. Kotsiantis, I. D. Zaharakis, and P. E. Pintelas. Machine learning: a review of classification and combining techniques. *Artificial Intelligence Review*, 26(3):159–190, 2006.

[250] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas. Supervised machine learning: A review of classification techniques. *Emerging artificial intelligence applications in computer engineering*, 160:3–24, 2007.

[251] I. Steinwart and A. Christmann. *Support vector machines*. Springer Science & Business Media, 2008.

[252] K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of machine learning research*, 2(Dec):265–292, 2001.

[253] G. Mountrakis, J. Im, and C. Ogole. Support vector machines in remote sensing: A review. *ISPRS journal of photogrammetry and remote sensing*, 66(3):247–259, 2011.

[254] K. P. Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[255] P. Baldi, S. Brunak, Y. Chauvin, C. A. Andersen, and H. Nielsen. Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics*, 16(5):412–424, 2000.

[256] K. Kowsari, K. Jafari Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown. Text classification algorithms: A survey. *Information*, 10(4):150, 2019.

[257] A. Merghadi, A. P. Yunus, J. Dou, J. Whiteley, B. ThaiPham, D. T. Bui, R. Avtar, and B. Abderrahmane. Machine learning methods for landslide susceptibility studies: A comparative overview of algorithm performance. *Earth-Science Reviews*, 207:103225, 2020.

[258] A. Tharwat. Classification assessment methods. *Applied Computing and Informatics*, 17(1):168–192, 2020.

[259] D. Chicco and G. Jurman. The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC genomics*, 21:1–13, 2020.

[260] V. Shah, M. Golmohammadi, I. Obeid, and J. Picone. Objective evaluation metrics for automatic classification of eeg events. *Biomedical Signal Processing: Innovation and Applications*, pages 223–255, 2021.

[261] H. Dalianis and H. Dalianis. Evaluation metrics and evaluation. *Clinical text mining: secondary use of electronic patient records*, pages 45–53, 2018.

[262] M. Sokolova, N. Japkowicz, and S. Szpakowicz. Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation. In *AI 2006: Advances in Artificial Intelligence: 19th Australian Joint Conference on Artificial Intelligence, Hobart, Australia, December 4-8, 2006. Proceedings 19*, pages 1015–1021. Springer, 2006.

[263] A. Mesaros, T. Heittola, and D. Ellis. Datasets and evaluation. *Computational Analysis of Sound Scenes and Events*, pages 147–179, 2018.

[264] X. Xu, H. Liu, M. Yao, et al. Recent progress of anomaly detection. *Complexity*, 2019, 2019.

[265] A. Subasi. *Practical machine learning for data analysis using python*. Academic Press, 2020.

[266] M. Bekkar, H. K. Djemaa, and T. A. Alitouche. Evaluation measures for models assessment over imbalanced data sets. *J Inf Eng Appl*, 3(10), 2013.

[267] U. Naseem, I. Razzak, and P. W. Eklund. A survey of pre-processing techniques to improve short-text quality: a case study on hate speech detection on twitter. *Multimedia Tools and Applications*, 80:35239–35266, 2021.

[268] P. Raghavan and N. El Gayar. Fraud detection using machine learning and deep learning. In *2019 international conference on computational intelligence and knowledge economy (ICCIKE)*, pages 334–339. IEEE, 2019.

[269] C.-E. Kuo, G.-T. Chen, and P.-Y. Liao. An eeg spectrogram-based automatic sleep stage scoring method via data augmentation, ensemble convolution neural network, and expert knowledge. *Biomedical Signal Processing and Control*, 70:102981, 2021.

[270] G. C. Cawley and N. L. Talbot. On over-fitting in model selection and subsequent selection bias in performance evaluation. *Journal of Machine Learning Research*, 11(Jul):2079–2107, 2010.

[271] B. Boehmke and B. M. Greenwell. *Hands-on machine learning with R*. CRC press, 2019.

[272] J. Vanerio and P. Casas. Ensemble-learning approaches for network security and anomaly detection. In *Proceedings of the Workshop on Big Data Analytics and Machine Learning for Data Communication Networks*, pages 1–6, 2017.

[273] F. Acebes, M. Pereda, D. Poza, J. Pajares, and J. M. Galán. Stochastic earned value analysis using monte carlo simulation and statistical learning techniques. *International Journal of Project Management*, 33(7):1597–1609, 2015.

[274] K. S. Killourhy and R. A. Maxion. Comparing anomaly-detection algorithms for keystroke dynamics. In *2009 IEEE/IFIP Int. Conf. on Dependable Sys. Net.*, pages 125–134, Lisbon, Portugal, 2009.

[275] R. Schiffler. Maximum z score and outliers. *The Am. Statistician*, 42(1):79–80, 1988.

[276] M. Daszykowski, K. Kaczmarek, Y. Vander Heyden, and B. Walczak. Robust statistics in data analysis—a review: Basic concepts. *Chemometrics and intelligent laboratory systems*, 85(2):203–219, 2007.

[277] L. Chiaramonte, E. Croci, and F. Poli. Should we trust the z-score? evidence from the european banking industry. *Global Finance Journal*, 28:111–131, 2015.

[278] B. Tang and H. He. A local density-based approach for outlier detection. *Neurocomputing*, 241:171–180, 2017.

[279] R. McGill, J. W. Tukey, and W. A. Larsen. Variations of box plots. *The american statistician*, 32(1):12–16, 1978.

[280] J. M. Chambers. *Graphical methods for data analysis*. CRC Press, 2018.

[281] J. L. Myers, A. Well, and R. F. Lorch. *Research design and statistical analysis*. Routledge, 2010.

[282] H. Spratt, H. Ju, and A. R. Brasier. A structured approach to predictive modeling of a two-class problem using multidimensional data sets. *Methods*, 61(1):73–85, 2013.

[283] H. Rezazadegan Tavakoli, E. Rahtu, and J. Heikkilä. Fast and efficient saliency detection using sparse sampling and kernel density estimation. In *Image Analysis: 17th Scandinavian Conference, SCIA 2011, Ystad, Sweden, May 2011. Proceedings 17*, pages 666–675. Springer, 2011.

[284] D. Wang, L. Gao, Y. Qu, X. Sun, and W. Liao. Frequency-to-spectrum mapping gan for semisupervised hyperspectral anomaly detection. *CAAI Transactions on Intelligence Technology*, 2023.

[285] Y. Chen, J. Tao, Q. Zhang, K. Yang, X. Chen, J. Xiong, R. Xia, and J. Xie. Saliency detection via the improved hierarchical principal component analysis method. *Wireless communications and mobile computing*, 2020:1–12, 2020.

[286] S. Sanei and H. Hassani. *Singular spectrum analysis of biomedical signals*. CRC press, 2015.

[287] R. Teti, K. Jemielniak, G. O'Donnell, and D. Dornfeld. Advanced monitoring of machining operations. *CIRP annals*, 59(2):717–739, 2010.

[288] B. Yang, L. Zhong, J. Wang, H. Shu, X. Zhang, T. Yu, and L. Sun. State-of-the-art one-stop handbook on wind forecasting technologies: An overview of classifications, methodologies, and analysis. *Journal of Cleaner Production*, 283:124628, 2021.

[289] V. Oropeza and M. Sacchi. Simultaneous seismic data denoising and reconstruction via multichannel singular spectrum analysis. *Geophysics*, 76(3):V25–V32, 2011.

[290] X. Wang, Q. Zhou, J. Harer, G. Brown, S. Qiu, Z. Dou, J. Wang, A. Hinton, C. A. Gonzalez, and P. Chin. Deep learning-based classification and anomaly detection of side-channel signals. In *Cyber Sensing 2018*, volume 10630, pages 37–44. SPIE, 2018.

[291] J. Yang, K. Zhou, Y. Li, and Z. Liu. Generalized out-of-distribution detection: A survey. *arXiv preprint arXiv:2110.11334*, 2021.

[292] M. Nakanishi, K. Sato, and H. Terada. Anomaly detection by autoencoder based on weighted frequency domain loss. *arXiv preprint arXiv:2105.10214*, 2021.

[293] F. Cai. *Out-of-Distribution Detection in Learning-Enabled Cyber-Physical Systems*. PhD thesis, Vanderbilt University, 2022.

[294] I. Castillo Camacho and K. Wang. A comprehensive review of deep-learning-based methods for image forensics. *Journal of imaging*, 7(4):69, 2021.

[295] H. Zhao, Y. Li, N. He, K. Ma, L. Fang, H. Li, and Y. Zheng. Anomaly detection for medical images using self-supervised and translation-consistent features. *IEEE Transactions on Medical Imaging*, 40(12):3641–3651, 2021.

[296] J. Mao, H. Wang, and B. F. Spencer Jr. Toward data anomaly detection for automated structural health monitoring: Exploiting generative adversarial nets and autoencoders. *Structural Health Monitoring*, 20(4):1609–1626, 2021.

[297] A. Ortiz, P. López, J. L. Luque, F. J. Martínez-Murcia, D. Aquino-Britez, and J. Ortega. An anomaly detection approach for dyslexia diagnosis using eeg signals. In *Understanding the Brain Function and Emotions: 8th International Work-Conference on the Interplay Between Natural and Artificial Computation, IWINAC 2019, Almería, Spain, June 3–7, 2019, Proceedings, Part I 8*, pages 369–378. Springer, 2019.

[298] M. W. Musselman. Monitoring of biomedical systems using non-stationary signal analysis. 2013.

[299] J. Birjandtalab, M. B. Pouyan, D. Cogan, M. Nourani, and J. Harvey. Automated seizure detection using limited-channel eeg and non-linear dimension reduction. *Computers in biology and medicine*, 82:49–58, 2017.

[300] G. D. Fraser, A. D. Chan, J. R. Green, and D. T. MacIsaac. Automated biosignal quality analysis for electromyography using a one-class support vector machine. *IEEE Transactions on Instrumentation and Measurement*, 63(12):2919–2930, 2014.

[301] A. Rai and S. H. Upadhyay. A review on signal processing techniques utilized in the fault diagnosis of rolling element bearings. *Tribology International*, 96:289–306, 2016.

[302] M. Civera and C. Surace. A comparative analysis of signal decomposition techniques for structural health monitoring on an experimental benchmark. *Sensors*, 21(5):1825, 2021.

[303] S. M. P. Dinakarrao, A. Jantsch, and M. Shafique. Computer-aided arrhythmia diagnosis with bio-signal processing: A survey of trends and techniques. *ACM Computing Surveys (CSUR)*, 52(2):1–37, 2019.

[304] M. Civera, M. Ferraris, R. Ceravolo, C. Surace, and R. Betti. The teager-kaiser energy cepstral coefficients as an effective structural health monitoring tool. *Applied Sciences*, 9(23):5064, 2019.

[305] M. Morgantini, R. Betti, and L. Balsamo. Structural damage assessment through features in quefrency domain. *Mechanical Systems and Signal Processing*, 147:107017, 2021.

[306] L. Wang, G. Sun, Y. Wang, J. Ma, X. Zhao, and R. Liang. Afexplorer: Visual analysis and interactive selection of audio features. *Visual Informatics*, 6(1):47–55, 2022.

[307] P. Chaovalit, A. Gangopadhyay, G. Karabatis, and Z. Chen. Discrete wavelet transform-based time series analysis and mining. *ACM Computing Surveys (CSUR)*, 43(2):1–37, 2011.

[308] A. Cohen and M. A. Atoui. On wavelet-based statistical process monitoring. *Transactions of the Institute of Measurement and Control*, 44(3):525–538, 2022.

[309] Z. K. Peng and F. Chu. Application of the wavelet transform in machine condition monitoring and fault diagnostics: a review with bibliography. *Mechanical systems and signal processing*, 18(2):199–221, 2004.

[310] Z. Du, L. Ma, H. Li, Q. Li, G. Sun, and Z. Liu. Network traffic anomaly detection based on wavelet analysis. In *2018 IEEE 16th International Conference on Software Engineering Research, Management and Applications (SERA)*, pages 94–101. IEEE, 2018.

[311] M. Vlachos, P. S. Yu, V. Castelli, and C. Meek. Structural periodic measures for time-series data. *Data Mining and Knowledge Discovery*, 12:1–28, 2006.

[312] E. Keogh, J. Lin, and A. Fu. Hot sax: Efficiently finding the most unusual time series subsequence. In *Fifth IEEE International Conference on Data Mining (ICDM'05)*, pages 8–pp. Ieee, 2005.

[313] J. Lin, E. Keogh, L. Wei, and S. Lonardi. Experiencing sax: a novel symbolic representation of time series. *Data Mining and knowledge discovery*, 15:107–144, 2007.

[314] S. Kandanaarachchi, M. A. Muñoz, R. J. Hyndman, and K. Smith-Miles. On normalization and algorithm selection for unsupervised outlier detection. *Data Mining and Knowledge Discovery*, 34(2):309–354, 2020.

[315] E. Acuna and C. Rodriguez. A meta analysis study of outlier detection methods in classification. *Technical paper, Department of Mathematics, University of Puerto Rico at Mayaguez*, 1:25, 2004.

[316] X. Song, Q. Wen, Y. Li, and L. Sun. Robust time series dissimilarity measure for outlier detection and periodicity detection. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 4510–4514, 2022.

[317] P. P. Angelov and X. Gu. Anomaly detection: Empirical approach. In *Empirical Approach to Machine Learning*, pages 157–173. Springer, 2019.

[318] A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu. *Spatial tessellations: concepts and applications of Voronoi diagrams*, volume 501. John Wiley & Sons, 2009.

[319] P. P. Angelov and X. Gu. Applications of autonomous anomaly detection. In *Empirical Approach to Machine Learning*, pages 249–259. Springer, 2019.

[320] A. V. Oppenheim, A. S. Willsky, and I. Y. Withian. *Signals and systems*. Prentice-Hall Int., New Jersey, USA, 1983.

[321] E. C. Erkuş and V. Purutçuoğlu Gazi. Detection of hidden patterns in time series data via multiple-time fod method. 2019.

[322] E. C. Erkuş, V. Purutçuoğlu, and E. Purutçuoğlu. Detection of abnormalities in heart rate using multiple fourier transforms. *International Journal of Environmental Science and Technology*, 16:5237–5242, 2019.

[323] H. Nyquist. Certain topics in telegraph transmission theory. *Transactions of the American Institute of Electrical Engineers*, 47(2):617–644, 1928.

[324] T. A. Sorensen. A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on danish commons. *Biol. Skar.*, 5:1–34, 1948.

[325] D. S. Baim, W. S. Colucci, E. S. Monrad, H. S. Smith, R. F. Wright, A. Lanoue, D. F. Gauthier, B. J. Ransil, W. Grossman, and E. Braunwald. Survival of patients with severe congestive heart failure treated with oral milrinone. *Journal of the American College of Cardiology*, 7(3):661–670, 1986.

[326] P. Indyk and R. Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 604–613, 1998.

[327] V. Behravan, N. E. Glover, R. Farry, P. Y. Chiang, and M. Shoaib. Rate-adaptive compressed-sensing and sparsity variance of biomedical signals. In *2015 IEEE 12th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, pages 1–6. IEEE, 2015.

[328] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley. Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *circulation*, 101(23):e215–e220, 2000.

[329] E. O. Brigham. *The fast Fourier transform and its applications*. Prentice-Hall, Inc., 1988.

[330] E. C. Erkuş and V. Purutçuoğlu. Anomaly detection in sliding windows using dissimilarity metrics in time series data. In *4th International Conference on Artificial Intelligence and Applied Mathematics in Engineering (ICAIAME 2022), Baku, Azerbaijan*, volume in press, pages 1–16, 2022.

[331] A. P. Reynolds, G. Richards, B. de la Iglesia, and V. J. Rayward-Smith. Clustering rules: a comparison of partitioning and hierarchical clustering algorithms. *Journal of Mathematical Modelling and Algorithms*, 5(4):475–504, 2006.

[332] S. D. Greenwald. *The development and analysis of a ventricular fibrillation detector*. PhD thesis, Massachusetts Institute of Technology, 1986.

[333] G. B. Moody and R. G. Mark. The impact of the mit-bih arrhythmia database. *IEEE Engineering in Medicine and Biology Magazine*, 20(3):45–50, 2001.

[334] A. Taddei, G. Distante, M. Emdin, P. Pisani, G. Moody, C. Zeelenberg, and C. Marchesi. The european st-t database: standard for evaluating systems for the analysis of st-t changes in ambulatory electrocardiography. *European heart journal*, 13(9):1164–1172, 1992.

[335] K. Zhang, M. Hutter, and H. Jin. A new local distance-based outlier detection approach for scattered real-world data. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 813–822. Springer, 2009.

[336] F. E. Grubbs. Sample criteria for testing outlying observations. *The Annals of Mathematical Statistics*, pages 27–58, 1950.

[337] A. Ruha, S. Sallinen, and S. Nissila. A real-time microprocessor qrs detector system with a 1-ms timing accuracy for the measurement of ambulatory hrv. *IEEE Transactions on Biomedical Engineering*, 44(3):159–167, 1997.

[338] G. Moody. A new method for detecting atrial fibrillation using rr intervals. *Computers in Cardiology*, pages 227–230, 1983.

Appendix A

APPENDIX

A.1 Extension Works

A.1.1 Comparison of Dissimilarity Metrics as Features

In this section, a comparison is made between the discriminability of the linear dissimilarity metrics including Euclidean, square Euclidean, city block, Chebyshew, and their generalization, namely, Minkowski and dynamic time warping (DTW) metric as a non-linear feature are tested in terms of their usability as a dissimilarity feature to classify the anomalous intervals in the data. To perform the classification, a synthetic ECG dataset is generated using a publicly available ECG data generation toolbox [337] obtained from the Physionet website [338] under various experimental conditions.

The dataset is subjected to different levels of noise, represented by Signal-to-Noise Ratio (SNR) values ranging from 10dB to -10dB, resulting in a total of 9 SNR values. The purpose is to observe the impact of noise on the performance of the dissimilarity metrics within the range of 10dB to -10 dB. However, for simplicity, this report only presents the results for SNR values of 10dB, 0dB, and -10 dB. Results for the remaining SNR values can be provided upon request.

For each SNR case, two sets of data are generated for each of the four sub-experimental conditions, representing the control and anomaly groups. The default parameter values correspond to a regular heartbeat sequence with a heart rate of 80 bpm, proper amplitude, and PQRST structure intervals. The parameter settings are as follows: amplitude of 1000 units, QRS width of 0.1 seconds, T-wave amplitude of 500 units, data length of 10 seconds, and a sampling rate of 500 Hz. The generated data intervals are concatenated to simulate stimulus conditions applied at equal intervals. The four sub-experimental conditions are defined independently by altering the default parameters for each condition. These conditions include:

1. SNR = 10dB
    (a) Arrhythmia, with the change in the heart beat rate, representing type III outliers.

    The control group has 60 bpm and the event group has 110 bpm. The rest of the parameters are unchanged and remained the default values. The data can be visualized in Figure A.1, overall and zoomed-in formats, respectively for a) and b).

    (b) Anomaly in the amplitude of R peaks, representing type I outliers.

    The control group has R peak amplitudes of about 1000 units, and the event group has 2000 units. The rest of the parameters are unchanged and remained the default values. The data can be visualized in Figure A.2, overall and zoomed-in formats, respectively for a) and b).

    (c) Extended and narrowed QRS width, representing type II outliers.

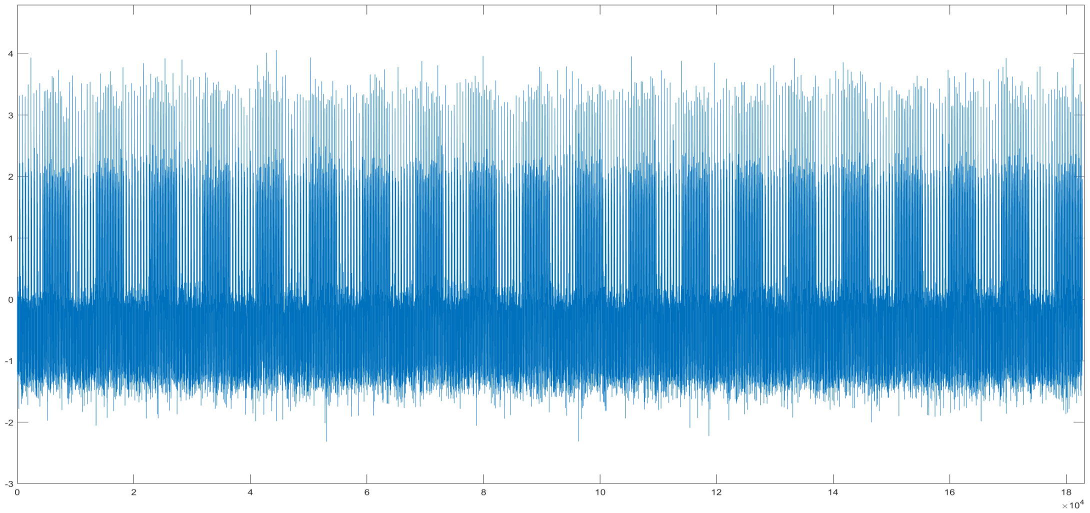
    The control group has 0.07 seconds of the QRS structure width, whereas the event group has 0.12 seconds. The rest of the parameters are unchanged and remained the default values. The data can be visualized in Figure A.3, overall and zoomed-in formats, respectively for a) and b).

    (d) Abnormal t wave amplitude, representing type II outliers.

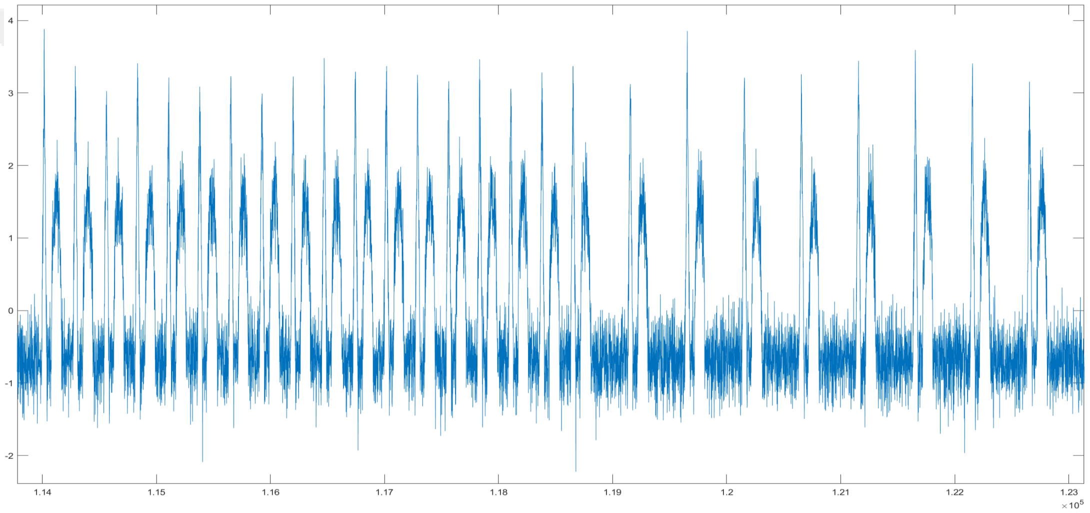
    The control group has t wave amplitudes of around 500 units, while the event group has around 1000 units. The rest of the parameters are unchanged and remained the default values. The data can be visualized in Figure A.4, overall and zoomed-in formats, respectively for a) and b).

    The same data generation process is applied for the remaining 8 different SNR values as well. The following part includes the visualizations for the SNR values of -10dB, representing noisy data.

2. SNR = -10dB



(a) Overall data.

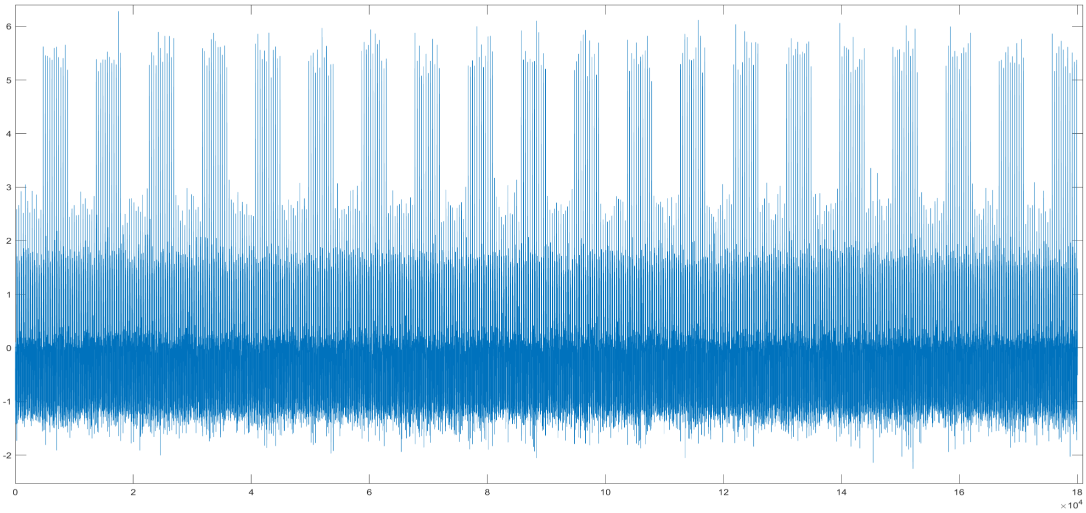


(b) Zoomed data with a random interval.

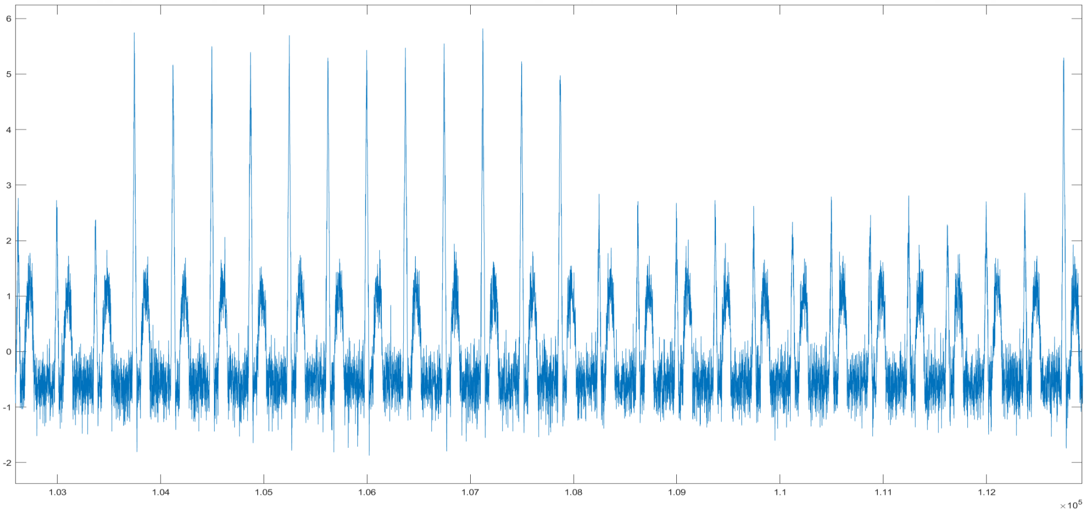
Figure A.1: Synthetic ECG data, illustrating arrhythmia with 60 and 110 bpm values, generated under SNR of 10dB.

(a) Arrhythmia, with the change in the heart beat rate, representing type III outliers.

The control group has 60 bpm and the event group has 110 bpm. The rest of the parameters are unchanged and remained the default values. The data can be visualized in Figure A.5, overall and zoomed-in formats, respectively for a) and b).



(a) Overall data.



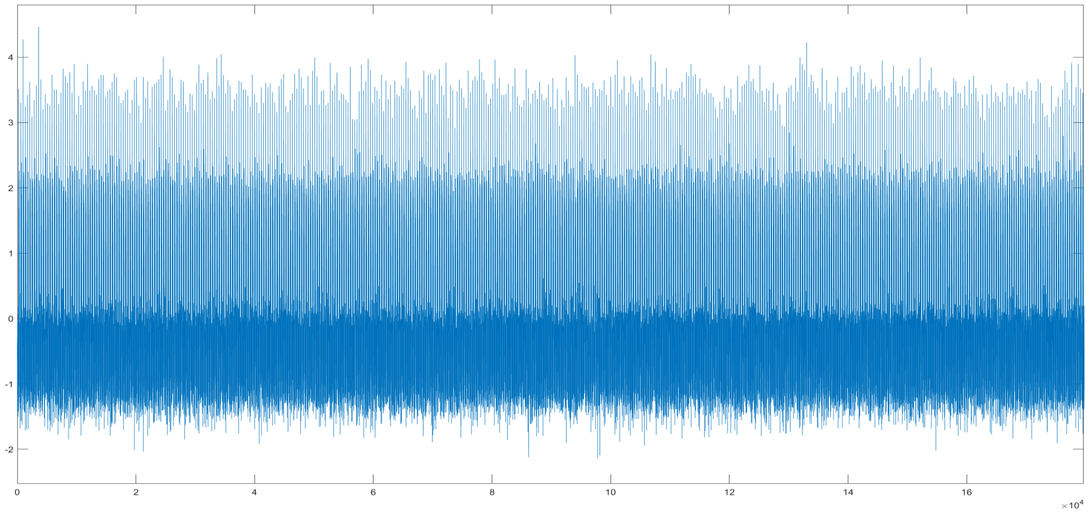
(b) Zoomed data with a random interval.

Figure A.2: Synthetic ECG data, illustrating type I outliers with 1000 and 2000 units in R peak amplitudes, generated under SNR of 10dB.

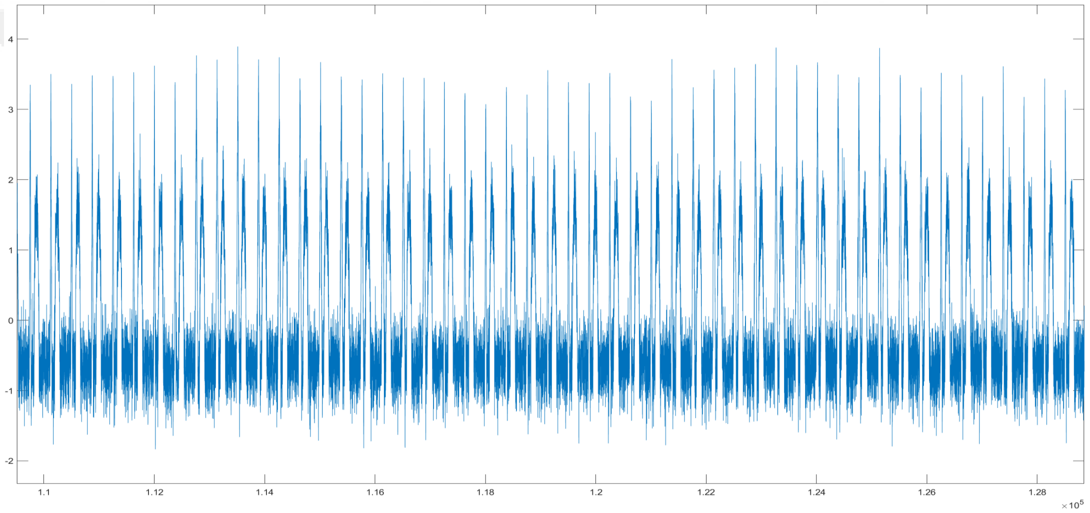
(b) Anomaly in the amplitude of R peaks, representing type I outliers.

The control group has R peak amplitudes of about 1000 units, and the event group has 2000 units. The rest of the parameters are unchanged and remained the default values. The data can be visualized in Figure A.6, overall and zoomed-in formats, respectively for a) and b).

(c) Extended and narrowed QRS width, representing type II outliers.



(a) Overall data.



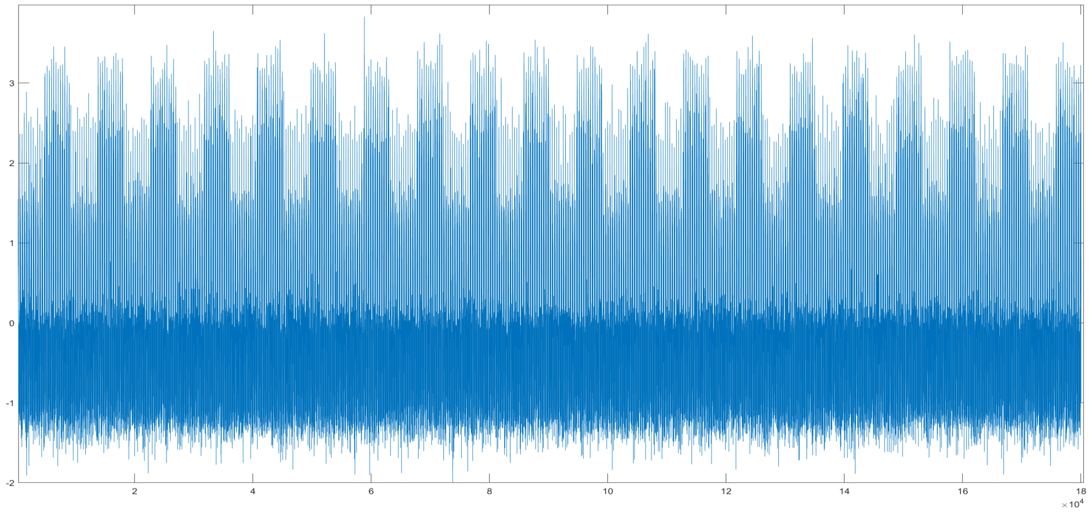
(b) Zoomed data with a random interval.

Figure A.3: Synthetic ECG data, illustrating narrowed and extended duration of QRS structures with 0.07 and 0.12 seconds values, generated under SNR of 10dB.

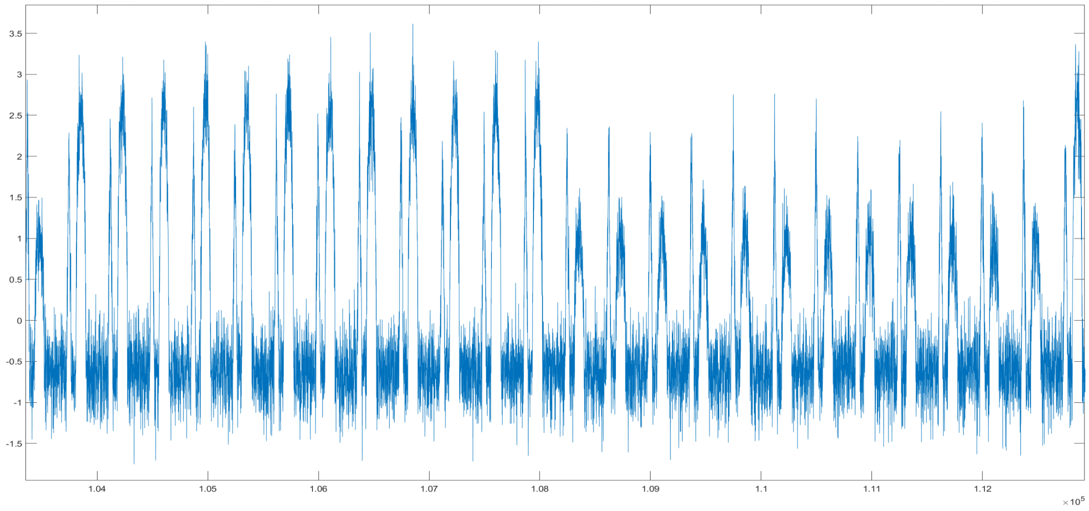
The control group has 0.07 seconds of the QRS structure width, whereas the event group has 0.12 seconds. The rest of the parameters are unchanged and remained the default values. The data can be visualized in Figure A.7, overall and zoomed-in formats, respectively for a) and b).

(d) Abnormal t wave amplitude, representing type II outliers.

The control group has t wave amplitudes of around 500 units, while the



(a) Overall data.

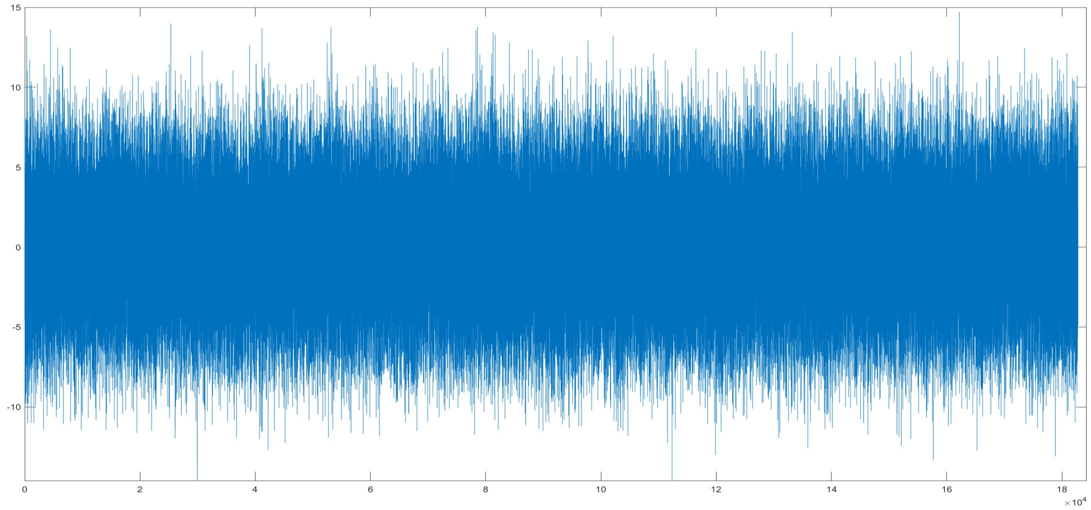


(b) Zoomed data with a random interval.

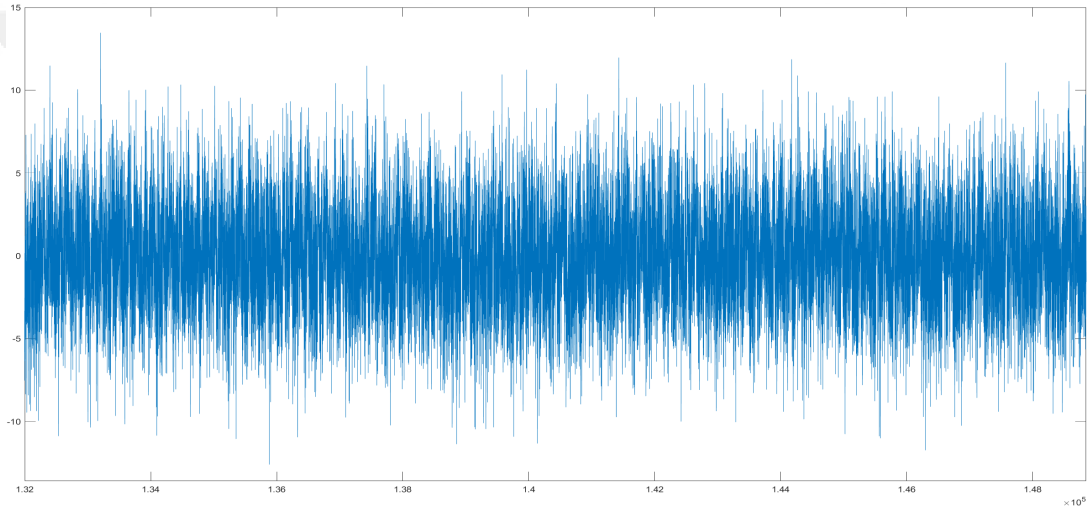
Figure A.4: Synthetic ECG data, illustrating t-wave abnormality with 500 and 1000 units in amplitude, generated under SNR of 10dB.

event group has around 1000 units. The rest of the parameters are unchanged and remained the default values. The data can be visualized in Figure A.8, overall and zoomed-in formats, respectively for a) and b).

In this study, a total of 10 virtual subjects were generated for each of the four sub-experimental conditions and for each signal-to-noise ratio (SNR) value. The generated data were divided into control and event intervals, with an imaginary stimulus in



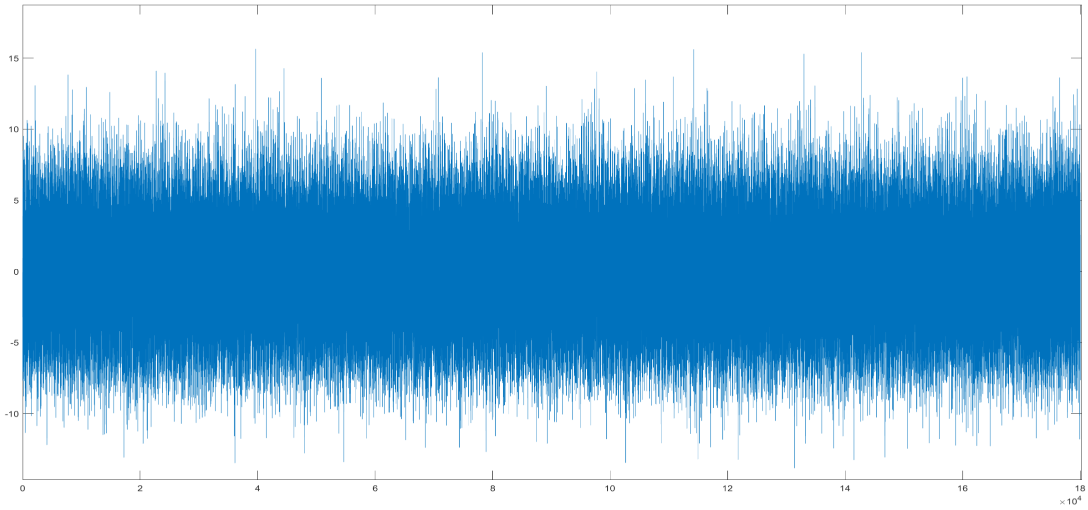
(a) Overall data.



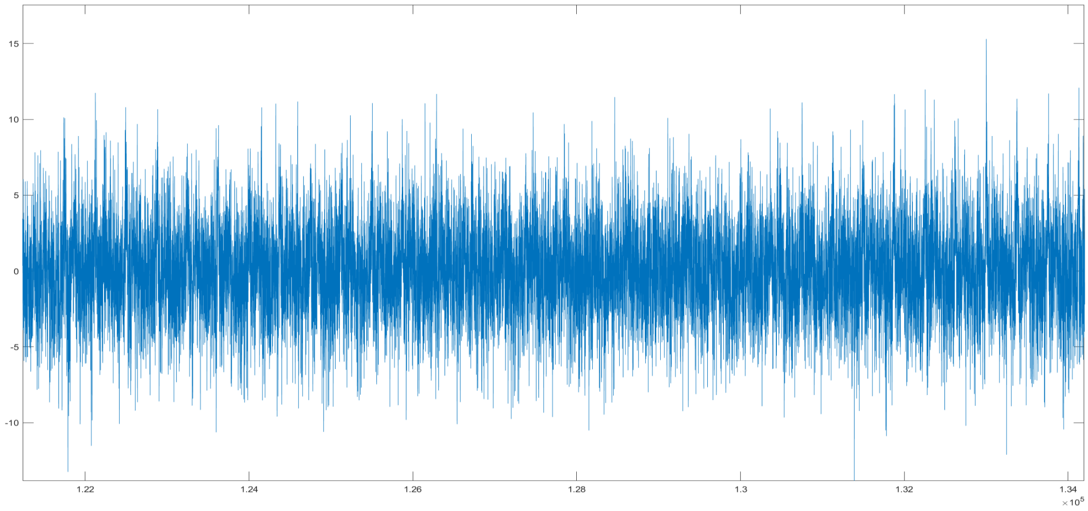
(b) Zoomed data with a random interval.

Figure A.5: Synthetic ECG data, illustrating arrhythmia with 60 and 110 bpm values, generated under SNR of -10dB.

between, enabling the performance of classification studies. The classification study followed a supervised approach, involving the extraction of control and event intervals from the data and the computation of their grand average. The feature extraction step iteratively examined each data interval without considering its label. During each iteration, dissimilarity metrics were calculated as features by comparing the current data interval with the grand average. After performing feature extraction, dissimilarity values were computed for each interval and dissimilarity measure.



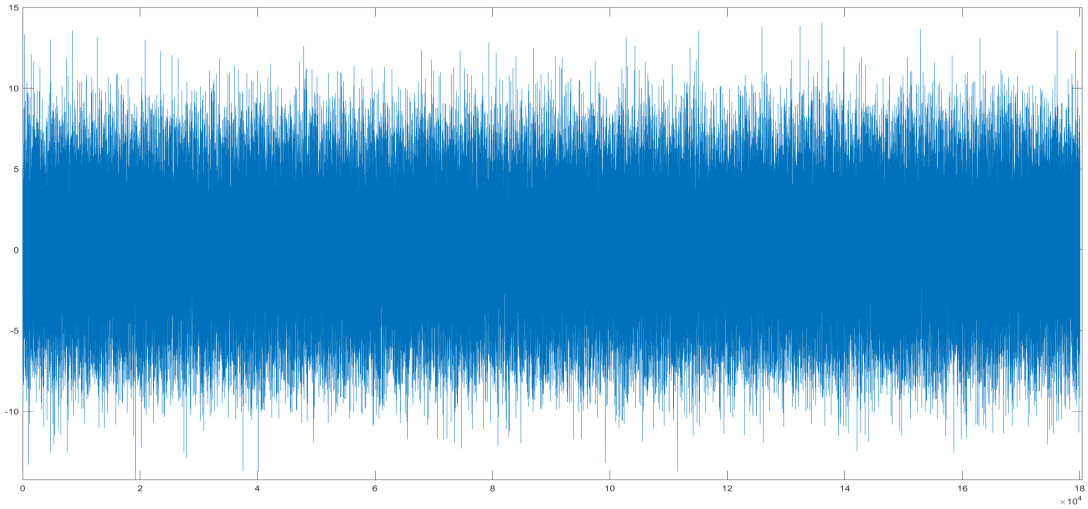
(a) Overall data.



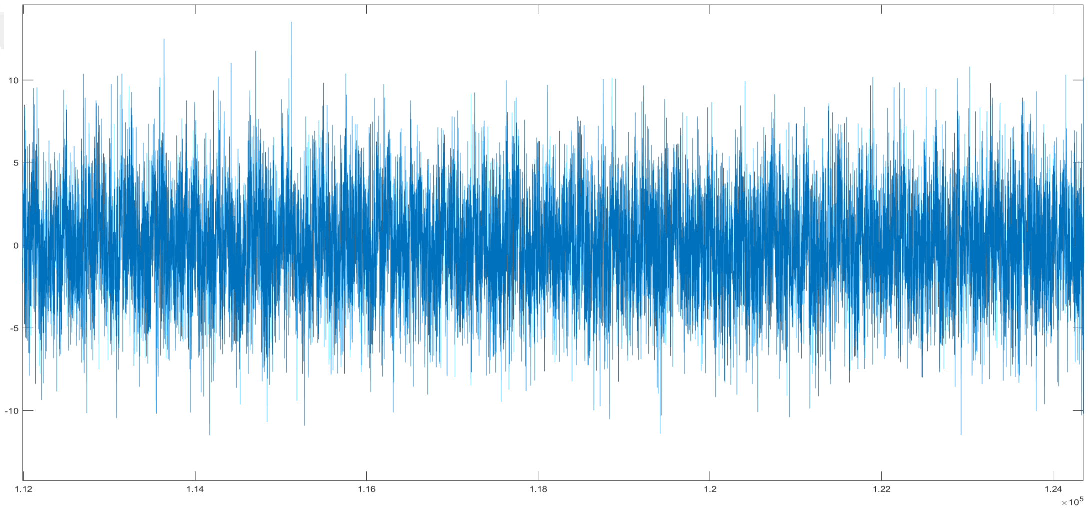
(b) Zoomed data with a random interval.

Figure A.6: Synthetic ECG data, illustrating type I outliers with 1000 and 2000 units in R peak amplitudes, generated under SNR of -10dB.

The classification step involved training the extracted features using classifier algorithms such as Tree classifier, linear discriminant analysis (LDA), K-nearest neighbor classifier (K-NN), and Naive Bayes classifier. These algorithms were used to validate the data. Testing was performed using the k-fold validation algorithm, resulting in a confusion matrix. Various measures including accuracy, sensitivity, specificity, F-Score, and Matthew's correlation coefficient were computed, but only accuracy is reported in this study for brevity.



(a) Overall data.

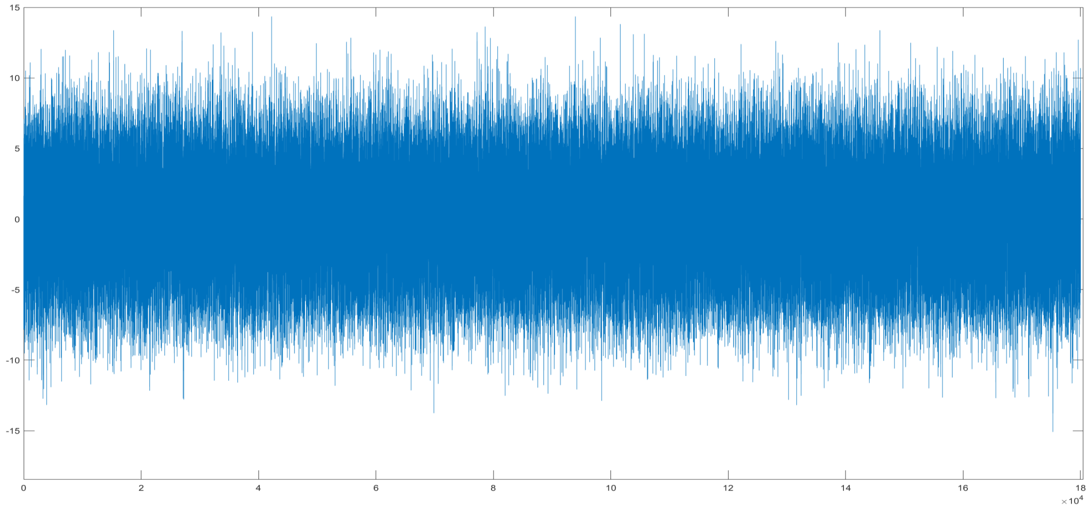


(b) Zoomed data with a random interval.

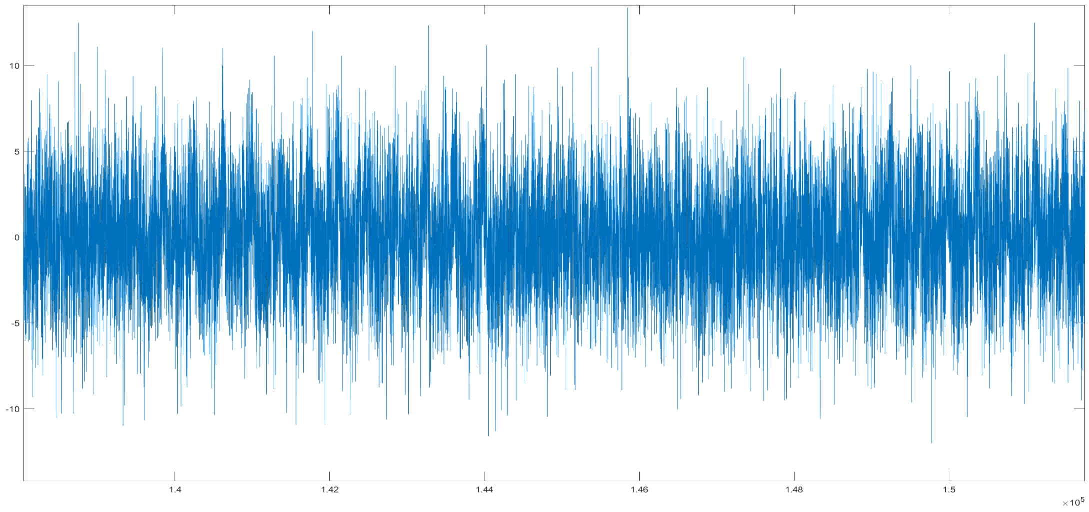
Figure A.7: Synthetic ECG data, illustrating narrowed and extended duration of QRS structures with 0.07 and 0.12 seconds values, generated under SNR of -10dB.

The provided tables, namely Table A.1, Table A.2, Table A.3, and Table A.4, present classification accuracy results for different dissimilarity metrics, classifiers, and signal-to-noise ratio (SNR) values in the context of synthetic ECG data analysis.

Table A.1 focuses on the effect of heart rate variation (ranging from 60 to 110 bpm) on classification accuracy. The dissimilarity metrics evaluated include Euclidean distance, square Euclidean distance, city block distance, Minkowski distance, Cheby-



(a) Overall data.



(b) Zoomed data with a random interval.

Figure A.8: Synthetic ECG data, illustrating t-wave abnormality with 500 and 1000 units in amplitude, generated under SNR of -10dB.

shev distance, and Dynamic Time Warping (DTW) distance. The classifiers used are Tree, LDA, KNN, and NBayes. The results demonstrate that DTW distance performs less effectively compared to other dissimilarity metrics when dealing with purely arrhythmic anomalies.

Table A.2 investigates the impact of R peak amplitude differences (ranging from 1000 to 2000 units) on classification accuracy. The dissimilarity metrics, classifiers, and

Table A.1: The classification results of synthetic ECG database with 60 to 110 bpm.

Dissimilarity Distance Feature	Classifier	Accuracy (in range of 0-1)		
		SNR 10dB	SNR 0dB	SNR -10dB
Euclidean Distance	Tree	1	1	0.78
	LDA	1	1	0.78
	KNN	1	1	0.77
	NBayes	1	1	0.79
Square Euclidean Distance	Tree	1	1	0.76
	LDA	1	1	0.79
	KNN	1	1	0.77
	NBayes	1	1	0.79
City Block Distance	Tree	1	1	0.77
	LDA	1	1	0.78
	KNN	1	1	0.77
	NBayes	1	1	0.78
Minkowski Distance	Tree	1	1	0.78
	LDA	1	1	0.78
	KNN	1	1	0.76
	NBayes	1	1	0.78
Chebyshev Distance	Tree	1	1	0.79
	LDA	1	1	0.78
	KNN	1	1	0.75
	NBayes	1	1	0.78
DTW Distance	Tree	1	0.99	0.67
	LDA	1	0.99	0.68
	KNN	1	0.99	0.62
	NBayes	1	0.99	0.68

SNR values remain the same as in the previous table. Similar to the previous table, DTW distance exhibits lower accuracy values, indicating its unsuitability for pure Type I outliers.

Table A.3 examines the classification results for different QRS width variations (ranging from 0.07 to 0.12 seconds). The dissimilarity metrics, classifiers, and SNR values remain consistent with the previous tables. Once again, DTW distance performs inferiorly compared to other dissimilarity metrics in distinguishing synthetic ECG data with varying QRS structure widths.

Lastly, Table A.4 presents the classification accuracy results for different T wave amplitude differences (ranging from 500 to 1000 units). The dissimilarity metrics, classifiers, and SNR values remain unchanged. The findings align with the previ-

Table A.2: The classification results of synthetic ECG database with R peak amplitude of 1000 to 2000 units.

Dissimilarity Distance Feature	Classifier	Accuracy (in range of 0-1)		
		SNR 10dB	SNR 0dB	SNR -10dB
Euclidean Distance	Tree	1	1	0.96
	LDA	1	1	0.96
	KNN	1	1	0.95
	NBayes	1	1	0.96
Square Euclidean Distance	Tree	1	1	0.98
	LDA	1	1	0.97
	KNN	1	1	0.98
	NBayes	1	1	0.98
City Block Distance	Tree	1	1	0.95
	LDA	1	1	0.96
	KNN	1	1	0.96
	NBayes	1	1	0.96
Minkowski Distance	Tree	1	1	0.95
	LDA	1	1	0.96
	KNN	1	1	0.96
	NBayes	1	1	0.96
Chebyshev Distance	Tree	1	1	0.95
	LDA	1	1	0.96
	KNN	1	1	0.96
	NBayes	1	1	0.96
DTW Distance	Tree	0.71	0.70	0.60
	LDA	0.71	0.69	0.61
	KNN	0.70	0.69	0.55
	NBayes	0.71	0.69	0.59

ous tables, with DTW distance demonstrating lower accuracy values, indicating its inadequacy as a sole dissimilarity metric for discriminating synthetic ECG data.

Despite the lower performance of DTW distance in these experiments, it should be noted that DTW offers advantages in capturing non-linear mappings between time series and comparing time intervals of different lengths. This property can be particularly valuable for analyzing biomedical responses with varying durations, which is often encountered in real-time scenarios. Therefore, rather than dismissing DTW distance entirely, there is a need to improve and incorporate it into a more robust feature extraction algorithm for ECG data analysis.

Table A.3: The classification results of synthetic ECG database with QRS width of 0.07 seconds to 0.12 seconds.

Dissimilarity Distance Feature	Classifier	Accuracy (in range of 0-1)		
		SNR 10dB	SNR 0dB	SNR -10dB
Euclidean Distance	Tree	1	1	0.62
	LDA	1	0.99	0.63
	KNN	1	0.99	0.58
	NBayes	1	1	0.63
Square Euclidean Distance	Tree	1	1	0.62
	LDA	1	1	0.62
	KNN	1	1	0.59
	NBayes	1	1	0.62
City Block Distance	Tree	1	0.99	0.60
	LDA	1	0.99	0.63
	KNN	1	1	0.57
	NBayes	1	1	0.62
Minkowski Distance	Tree	1	1	0.62
	LDA	1	0.99	0.63
	KNN	1	1	0.57
	NBayes	1	0.99	0.63
Chebyshev Distance	Tree	1	1	0.62
	LDA	1	0.99	0.63
	KNN	1	0.99	0.56
	NBayes	1	1	0.63
DTW Distance	Tree	1	0.54	0.50
	LDA	1	0.58	0.52
	KNN	1	0.55	0.54
	NBayes	1	0.57	0.48

A.1.2 Comparison of Pitch Frequency Algorithms

The purpose of this study is to compare the pitch estimation methods defined in Section 2.2.2 that are available in a single MATLAB library: 'pitch.m' [203]. Their performances are compared to each other, as well as some other statistical and spectral features, under a variety of experimental conditions, including univariate and multivariate usages in conjunction with the three most commonly used machine learning classifiers: the Tree classifier, LDA classifier, and K-NN classifier. The brief background information for these classifiers can be found in Section 2.4.

Four different synthetic datasets have been generated, each consisting of 50 subjects.

Table A.4: The classification results of synthetic ECG database with t wave amplitude of 500 to 1000 units.

Dissimilarity Distance Feature	Classifier	Accuracy (in range of 0-1)		
		SNR 10dB	SNR 0dB	SNR -10dB
Euclidean Distance	Tree	1	1	0.97
	LDA	1	1	0.98
	KNN	1	1	0.97
	NBayes	1	1	0.98
Square Euclidean Distance	Tree	1	1	0.98
	LDA	1	1	0.98
	KNN	1	1	0.98
	NBayes	1	1	0.99
City Block Distance	Tree	1	1	0.97
	LDA	1	1	0.98
	KNN	1	1	0.97
	NBayes	1	1	0.98
Minkowski Distance	Tree	1	1	0.96
	LDA	1	1	0.98
	KNN	1	1	0.97
	NBayes	1	1	0.98
Chebyshev Distance	Tree	1	1	0.97
	LDA	1	1	0.97
	KNN	1	1	0.97
	NBayes	1	1	0.97
DTW Distance	Tree	1	0.83	0.55
	LDA	1	0.85	0.56
	KNN	1	0.86	0.51
	NBayes	1	0.85	0.55

The datasets have a total of 1,000,000 samples, with equal numbers of sequentially concatenated control and experimental data samples, each containing 1,000 samples. All of the control data intervals are generated using a Gaussian Normal distribution with zero mean and unit variance.

On the other hand, the experimental data sequences for the first synthetic dataset are generated using a Gaussian distribution with a mean value of 3 and unit variance to test the capturability of fold changes. The second dataset is generated using an exponential distribution with a rate of 5 for the experimental sequence of samples. The third dataset is similarly generated but using the log-normal distribution with zero mean and a standard deviation of 5. The fourth dataset is generated using the

Student's t-distribution with a degree of freedom of 5. And finally, a real, publicly available ECG dataset, namely 'MIT-BIH Long Term ECG Dataset', obtained from Physionet website [338, 328] is used to test the experimental setup to compare the pitch estimation algorithms on the proposed algorithm, PAD, feature performances. Here, the intervals between P to T waves are considered as the experimental, and the intervals from T to P waves are considered the control samples.

The accuracy and computational time results of the above-mentioned experiments are presented in Table A.5, Based on the performance results, one of the pitch estimation methods is chosen to proceed with the initial experimental tests of the proposed algorithm.

The accuracy results in Table A.5 show that the majority of feature groups achieve high accuracy (A=1) in both univariate and multivariate analyses. In both types of analyses, the statistical features consistently produce perfect accuracy, indicating their effectiveness in distinguishing between classes. The spectral features perform well as well, with A=1 for the majority of feature types. However, the accuracy of PAD features ranged from 0.69 to 1, depending on the specific feature and distance metric used. It implies that the discriminative power of PAD features may be variable.

In terms of computational time, statistical and spectral features have generally shorter processing times, ranging from 3.43 to 8.61 units on average. The PAD features, on the other hand, require more computation time, with the average ranging from 19.64 to 37.71 units. It implies that PAD feature extraction and analysis may be more computationally intensive than statistical and spectral features. The results show that statistical and spectral features are effective in achieving high classification accuracy on the synthetic dataset. The PAD features, while slightly less accurate, still provide adequate performance. It is worth noting, however, that the PAD features require more computational time than the other feature groups.

Looking at the accuracy results as presented in Table A.6, the Statistical feature group achieves the best performance, as all univariate and multivariate features achieve perfect accuracy (A = 1). The Spectral feature group achieves perfect accuracy for the Band Power feature but less so for Mean Frequency and Median Frequency. With different distance metrics, the PEF, CEP, and NCF groups also achieve relatively high

Table A.5: Synthetic Dataset: S1, Univariate and multivariate results. ’A’ stands for accuracy, and ’t’ is for computational time.

Feature Group	Feature	Tree Classifier				LDA Classifier				K-NN Classifier			
		Univar.		Multivar.		Univar.		Multivar.		Univar.		Multivar.	
		A	t	A	t	A	t	A	t	A	t	A	t
Statistical	Mean	1	4.59	1	5.86	1	3.89	1	5.49	1	5.49	1	15.76
	Variance	1	3.98			0.99	3.95			1	5.09		
	Skewness	1	3.63			1	3.82			1	6.47		
	Kurtosis	0.94	4.26			0.85	3.98			0.94	6.03		
	Median	1	4.32			1	3.66			1	5.99		
	Range	1	4.1			1	3.83			1	5.98		
	RMS	1	3.75			1	3.78			1	6.38		
Spectral	Band power	1	3.43	1	8.61	1	3.79	1	8.13	1	5.42	1	13.21
	Mean freq.	1	3.54			1	3.78			1	6.91		
	Median freq.	1	3.86			1	3.69			1	5.28		
PAD - PEF	Euclidean	1	4.18	1	34.21	0.98	4.16	1	33.53	1	6.59	1	37.71
	Square Eucl.	1	4.16			0.93	4.03			1	7.04		
	Manhattan	0.99	4.37			0.94	3.8			0.99	6.3		
	Chebyshev	1	3.88			0.99	3.89			1	6.5		
	DTW	0.99	4.28			0.95	3.96			0.99	8.11		
PAD - CEP	Euclidean	0.98	7.73	1	19.64	0.92	7.89	0.98	19.44	0.98	10.49	1	21.18
	Square Eucl.	0.99	7.8			0.86	7.31			0.99	10.24		
	Manhattan	0.86	7.68			0.84	7.5			0.85	9.63		
	Chebyshev	1	7.81			0.96	7.59			1	9.44		
	DTW	0.88	8.97			0.85	8.18			0.88	11.34		
PAD - NCF	Euclidean	0.95	8.88	1	20.69	0.87	8.85	0.99	20.88	0.95	16.79	1	23.5
	Square Eucl.	0.98	9.3			0.82	8.95			0.97	16.49		
	Manhattan	0.69	9.72			0.67	9.13			0.69	20.8		
	Chebyshev	1	9.44			0.95	9.24			0.99	14.27		
	DTW	0.71	10.85			0.71	10.46			0.7	19.72		
PAD - LHS	Euclidean	1	9.09	1	20.93	0.97	8.43	1	20.77	1	12.53	1	40.5
	Square Eucl.	1	8.01			0.91	8			0.97	13		
	Manhattan	0.94	8.8			0.87	9.36			0.94	17.52		
	Chebyshev	1	9.49			0.99	8.24			1	14.49		
	DTW	0.96	9.58			0.9	9.51			0.95	16.91		

accuracy (A > 0.8).

The Statistical feature group has the shortest computational time requirements for both univariate and multivariate features. The PAD Features groups, specifically PEF, on the other hand, require the most computational time for both univariate and multi-

Table A.6: Synthetic Dataset: S2, Univariate and multivariate results. ’A’ stands for accuracy, and ’t’ is for computational time.

Feature Group	Feature	Tree Classifier				LDA Classifier				K-NN Classifier			
		Univar.		Multivar.		Univar.		Multivar.		Univar.		Multivar.	
		A	t	A	t	A	t	A	t	A	t	A	t
Statistical	Mean	0.89	5.32	1	5.75	0.51	6.61	1	5.77	0.89	12.11	1	15.2
	Variance	1	5.06			1	4.98			1	9.99		
	Skewness	0.52	7.17			0.5	4.83			0.51	17.37		
	Kurtosis	0.52	7.03			0.5	5.83			0.52	12.74		
	Median	0.89	6			0.53	6.48			0.72	11.93		
	Range	1	6.06			1	4.88			1	13.54		
	RMS	1	5.52			1	4.77			1	13.01		
Spectral	Band power	1	5.53	1	8.33	1	5.69	1	8.13	1	10.56	1	16.58
	Mean freq.	0.5	8.09			0.5	5.9			0.5	14.31		
	Median freq.	0.52	7.46			0.5	6.62			0.52	15.53		
PAD - PEF	Euclidean	1	11.85	1	45.11	1	11.24	1	44.44	1	18.32	1	54.59
	Square Eucl.	1	10.42			0.98	10.5			1	17.12		
	Manhattan	1	10.71			0.98	10.4			1	15.85		
	Chebyshev	1	11.18			1	11.02			1	21.51		
	DTW	1	16.52			0.98	16.42			1	22.2		
PAD - CEP	Euclidean	1	7.21	1	19.08	0.95	7.35	1	18.95	1	13.17	1	25.04
	Square Eucl.	1	7.04			0.89	7.03			1	12.62		
	Manhattan	0.96	8			0.87	7.65			0.96	17.19		
	Chebyshev	1	8.15			1	7.99			1	13.53		
	DTW	0.97	10.76			0.88	10.06			0.95	30.95		
PAD - NCF	Euclidean	1	8.52	1	20.99	0.95	8.22	1	21.06	1	19.25	1	29.69
	Square Eucl.	1	7.66			0.9	7.86			1	17.88		
	Manhattan	0.96	7.63			0.87	7.06			0.96	14.73		
	Chebyshev	1	7.07			0.99	7.18			1	16.83		
	DTW	0.97	8.53			0.88	8.25			0.97	15.36		
PAD - LHS	Euclidean	1	6.73	1	18.91	1	6.96	1	18.83	1	16.49	1	32.18
	Square Eucl.	1	7.13			0.96	7.34			1	14.37		
	Manhattan	1	8.83			0.93	9.5			1	15.19		
	Chebyshev	1	6.95			1	7.54			1	16.44		
	DTW	1	8.4			0.96	8.81			1	13.14		

variate cases.

Overall, the Statistical feature group outperforms the other feature groups in terms of accuracy and computational time, making it a good choice for classification tasks on synthetic datasets. It is important to note, however, that the selection of feature group and classifier may be influenced by the specific characteristics and requirements of the dataset and the problem at hand.

Table A.7: Synthetic Dataset: S3, Univariate and multivariate results. ’A’ stands for accuracy, and ’t’ is for computational time.

Feature Group	Feature	Tree Classifier				LDA Classifier				K-NN Classifier			
		Univar.		Multivar.		Univar.		Multivar.		Univar.		Multivar.	
		A	t	A	t	A	t	A	t	A	t	A	t
Statistical	Mean	1	4.72	1	4.78	1	4.39	1	4.84	1	8.07	1	9.67
	Variance	1	4.82			1	4.45			1	14		
	Skewness	0.95	5.08			0.94	4.48			0.94	10.72		
	Kurtosis	0.76	5.24			0.76	4.59			0.76	12.23		
	Median	1	4.78			1	4.28			1	11.51		
	Range	0.97	4.97			0.97	4.38			0.97	11.37		
	RMS	1	4.96			1	4.83			1	10.56		
Spectral	Band power	1	4.74	1	7.59	1	4.25	1	7.56	1	12.18	1	14.84
	Mean freq.	1	5.96			1	5.9			1	10.28		
	Median freq.	0.97	6.26			0.95	6			0.97	10.73		
PAD - PEF	Euclidean	0.86	10.91	0.99	37.08	0.85	10.46	0.98	36.48	0.86	18.01	1	46.94
	Square Eucl.	0.89	11.15			0.88	10.92			0.89	15.41		
	Manhattan	0.78	11.2			0.78	9.97			0.78	19.38		
	Chebyshev	0.91	10.13			0.9	9.28			0.91	16.63		
	DTW	0.78	13.21			0.78	13.06			0.78	28.25		
PAD - CEP	Euclidean	0.66	8.81	0.96	24.92	0.66	7.79	0.92	24.5	0.67	23.94	0.96	34.25
	Square Eucl.	0.7	9.63			0.7	7.78			0.7	19.26		
	Manhattan	0.58	9.12			0.59	7.72			0.59	23.03		
	Chebyshev	0.76	7.85			0.76	7.4			0.76	15.55		
	DTW	0.6	11.38			0.6	10.65			0.61	23.93		
PAD - NCF	Euclidean	0.74	9.56	0.99	31.4	0.67	9.35	0.99	31.08	0.74	22.85	1	37.36
	Square Eucl.	0.79	9.25			0.72	8.62			0.79	20.1		
	Manhattan	0.69	9.99			0.51	10.87			0.69	24		
	Chebyshev	0.83	10.51			0.82	9.35			0.82	17.96		
	DTW	0.69	13.04			0.51	13.78			0.68	25.25		
PAD - LHS	Euclidean	0.69	8.59	0.9	20.03	0.66	7.59	0.69	19.28	0.69	27.68	0.94	31.31
	Square Eucl.	0.71	10			0.67	8.46			0.71	23.91		
	Manhattan	0.67	8.09			0.6	7.93			0.67	31.33		
	Chebyshev	0.7	8.2			0.65	8.17			0.7	23.22		
	DTW	0.69	9.43			0.61	9.21			0.69	23.42		

According to Table A.7, for most feature groups and classifiers, the results show that the multivariate approach outperforms the univariate approach. In terms of accuracy, the multivariate method achieves an average of 0.97, while the univariate method achieves an average of 0.81. This suggests that combining multiple features improves discrimination and classification performance when compared to using individual features separately.

In terms of computational time, the multivariate approach takes slightly longer than the univariate approach. However, the differences are minor, with the multivariate approach taking an average of 20.97 seconds and the univariate approach taking an average of 8.39 seconds. As a result, despite slightly longer processing times, the multivariate approach is still practical for practical implementation.

According to the results in Table A.8, in terms of accuracy, the multivariate approach outperforms the univariate approach across all feature groups and classifiers. In most cases, the multivariate approach's accuracy is close to one, indicating good classification performance. In all classifiers, the statistical feature group achieves perfect accuracy (A=1) for both univariate and multivariate approaches. Similarly, the PEF features achieves high accuracy (A>0.9) in most classifiers for both approaches. The CEP and NCF features are also accurate, albeit slightly less so than the other two groups.

In terms of computational time, the multivariate approach consistently takes longer than the univariate approach. This is to be expected because the multivariate approach deals with features of higher dimensionality, resulting in increased computational complexity. Despite the longer computation time, the multivariate approach is still viable, with an average time of 21 to 24 seconds. The univariate approach, on the other hand, takes significantly less time to compute, ranging from 7 to 11 seconds on average.

In terms of accuracy, the multivariate approach consistently outperforms the univariate approach for all feature groups and classifiers. In most cases, multivariate accuracy is significantly higher (greater than 0.9) than univariate accuracy (around 0.6-0.8). This suggests that combining multiple features yields more discriminative information for classification tasks. However, as indicated by the higher values of computational time for multivariate results, the multivariate approach requires more computational time than the univariate approach.

When compared to the Statistical and Spectral feature groups, the PAD features

Table A.8: Synthetic Dataset: S4, Univariate and multivariate results. ’A’ stands for accuracy, and ’t’ is for computational time.

Feature Group	Feature	Tree Classifier				LDA Classifier				K-NN Classifier			
		Univar.		Multivar.		Univar.		Multivar.		Univar.		Multivar.	
		A	t	A	t	A	t	A	t	A	t	A	t
Statistical	Mean	1	4.58	1	4.94	1	4.47	1	5.05	1	6.65	1	10.75
	Variance	1	4.05			1	4.11			1	8.06		
	Skewness	0.72	5.42			0.72	5.66			0.73	8.81		
	Kurtosis	0.96	5.38			0.94	4.71			0.96	8.28		
	Median	1	4.41			1	4.39			1	5.98		
	Range	0.99	4.5			0.99	4.88			0.98	6.07		
	RMS	0.52	5.49			0.51	5.92			0.52	9.1		
Spectral	Band power	0.52	6.31	1	7.15	0.51	6.1	1	6.57	0.52	9.33	1	11.8
	Mean freq.	0.7	5.99			0.69	5.71			0.7	8.38		
	Median freq.	1	5.4			1	5.32			1	7.81		
PAD - PEF	Euclidean	0.88	9.85	0.98	34.18	0.88	9.37	0.97	33.96	0.88	12.04	0.99	37.49
	Square Eucl.	0.93	9.32			0.9	9.05			0.93	11.53		
	Manhattan	0.8	9.71			0.8	9.41			0.8	13.12		
	Chebyshev	0.92	11.24			0.92	9.37			0.92	15.51		
	DTW	0.8	13.29			0.81	12.2			0.81	16.99		
PAD - CEP	Euclidean	0.63	8.23	0.93	22.46	0.63	8.33	0.92	22.23	0.64	10.8	0.94	23.33
	Square Eucl.	0.67	8.04			0.68	7.26			0.68	12.01		
	Manhattan	0.56	8.27			0.55	7.34			0.57	10.82		
	Chebyshev	0.74	9.2			0.74	8.34			0.74	12.58		
	DTW	0.57	10.03			0.57	9.01			0.58	11.19		
PAD - NCF	Euclidean	0.75	8.8	0.98	30.66	0.71	8.43	0.98	30.19	0.75	13.08	0.99	31.44
	Square Eucl.	0.82	9.09			0.77	9.16			0.81	12.39		
	Manhattan	0.69	10.75			0.53	10.95			0.68	15.53		
	Chebyshev	0.83	9.41			0.83	9.09			0.83	12.21		
	DTW	0.69	12.01			0.53	12.41			0.69	15.1		
PAD - LHS	Euclidean	0.65	6.9	0.91	16.27	0.58	6.6	0.76	16.38	0.64	14.81	0.93	17.34
	Square Eucl.	0.63	7.12			0.58	6.67			0.62	11.09		
	Manhattan	0.72	6.76			0.65	7.23			0.66	10.24		
	Chebyshev	0.76	7.11			0.53	8.08			0.76	11.97		
	DTW	0.73	8.1			0.67	7.63			0.68	13.27		

achieve higher accuracy. This suggests that the phase-amplitude duration features extracted from ECG signals contain useful information for classifying patients. However, the PAD features require more computational time, as evidenced by higher computational time values when compared to the Statistical and Spectral features.

The results of the analysis suggest that utilizing multivariate feature sets leads to

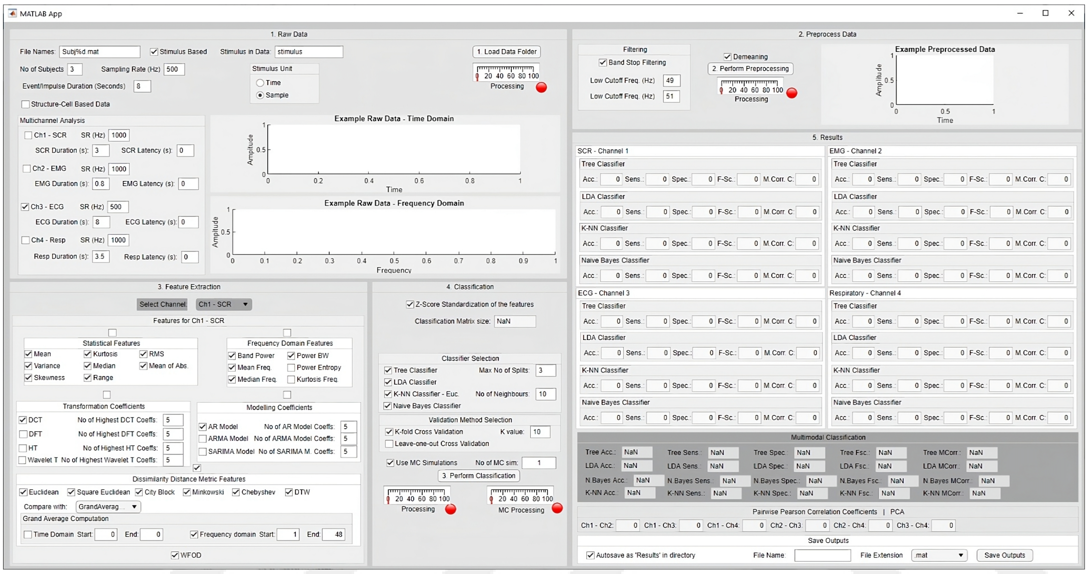
Table A.9: Real Dataset: MIT-BIH Long Term ECG, Univariate and multivariate results. 'A' stands for accuracy, and 't' is for computational time.

Feature Group	Feature	Tree Classifier				LDA Classifier				K-NN Classifier			
		Univar.		Multivar.		Univar.		Multivar.		Univar.		Multivar.	
		A	t	A	t	A	t	A	t	A	t	A	t
Statistical	Mean	0.61	5.22	0.98	5.82	0.61	3.93	0.89	5.11	0.61	20.33	0.99	11.2
	Variance	0.87	4.81			0.8	3.96			0.86	16.17		
	Skewness	0.81	4.8			0.5	6.15			0.81	27.49		
	Kurtosis	0.85	4.99			0.84	4.43			0.84	21.29		
	Median	0.58	5.44			0.57	4.41			0.58	18.07		
	Range	0.95	4.72			0.83	4.44			0.95	11.35		
	RMS	0.83	4.72			0.82	4.04			0.83	10.73		
Spectral	Band power	0.83	4.26	0.95	11.1	0.76	3.91	0.88	10.57	0.83	15.06	0.95	30.26
	Mean freq.	0.8	7.68			0.8	7.17			0.8	23.51		
	Median freq.	0.84	7.52			0.84	6.88			0.84	21.58		
PAD - PEF	Euclidean	0.8	8.78	0.92	28.24	0.77	8.35	0.85	27.57	0.81	28.72	0.95	38.61
	Square Eucl.	0.8	8.74			0.69	8.33			0.8	40.3		
	Manhattan	0.82	9.12			0.77	8.43			0.82	18.23		
	Chebyshev	0.83	8.84			0.77	8.32			0.82	35.35		
	DTW	0.81	9.76			0.78	9.24			0.81	15.33		
PAD - CEP	Euclidean	0.7	10.93	0.96	39.56	0.66	10.41	0.82	39.04	0.69	24.08	0.97	50.56
	Square Eucl.	0.77	10.72			0.71	9.87			0.77	17.85		
	Manhattan	0.68	11.89			0.62	11.23			0.67	22.28		
	Chebyshev	0.71	11.2			0.7	10.78			0.71	19.06		
	DTW	0.67	14.15			0.62	13.71			0.67	32.67		
PAD - NCF	Euclidean	0.71	10.47	0.95	38.49	0.68	9.68	0.78	37.79	0.71	26.91	0.96	43.37
	Square Eucl.	0.78	9.89			0.73	9.46			0.78	20.91		
	Manhattan	0.7	11.09			0.65	9.88			0.68	23.94		
	Chebyshev	0.72	10.67			0.7	9.95			0.71	19.47		
	DTW	0.69	11.81			0.66	13.02			0.7	34.77		
PAD - LHS	Euclidean	0.51	7.56	0.51	23.68	0.5	6.96	0.51	17.41	0.52	11.95	0.51	32.82
	Square Eucl.	0.52	8.49			0.5	7.54			0.51	15.19		
	Manhattan	0.51	9.04			0.5	7.09			0.52	14.43		
	Chebyshev	0.51	7.41			0.5	6.46			0.51	19.46		
	DTW	0.52	9.49			0.5	7.01			0.51	13.51		

improved classification accuracy compared to using individual features separately. Although the multivariate approach may require slightly longer computational times, the performance gain justifies the additional processing effort. These findings emphasize the importance of considering multiple features simultaneously for achieving more accurate and reliable classification results, with a trade-off between accuracy and computational time.

A.2 The Classification Graphical User Interface (GUI)

All of the analyses of the thesis study are performed in MATLAB software, using custom-made scripts over the built-in or pre-proven functions. Moreover, in order to perform the analyses, a new GUI for feature extraction and classification is made. Hereby, Figure A.9 represents the snapshot of the current version of the GUI which is made for this thesis study.



1. Raw Data
File Names: Subj%d.mat ☑ Stimulus Based Stimulus in Data: stimulus
No of Subjects: 3 Sampling Rate (Hz): 500
Event/Impulse Duration (Seconds): 8
Stimulus Unit: ○ Time ● Sample
☐ Structure-Cell Based Data

Multichannel Analysis
☐ Ch1 - SCR SR (Hz): 1000 SCR Duration (s): 3 SCR Latency (s): 0
☐ Ch2 - EMG SR (Hz): 1000 EMG Duration (s): 0.8 EMG Latency (s): 0
☑ Ch3 - ECG SR (Hz): 500 ECG Duration (s): 8 ECG Latency (s): 0
☐ Ch4 - Resp SR (Hz): 1000 Resp Duration (s): 3.5 Resp Latency (s): 0

2. Preprocess Data
Filtering: ☑ Band Stop Filtering
Low Cutoff Freq. (Hz): 49
Low Cutoff Freq. (Hz): 51
☑ Demeaning

3. Feature Extraction
Select Channel: Ch1 - SCR
Features for Ch1 - SCR
Statistical Features: ☑ Mean ☑ Kurtosis ☑ RMS ☑ Variance ☑ Median ☑ Mean of Abs. ☑ Skewness ☑ Range
Frequency Domain Features: ☑ Band Power ☑ Power BW ☑ Mean Freq. ☐ Power Entropy ☑ Median Freq. ☐ Kurtosis Freq.
Transformation Coefficients:
☑ DCT No of Highest DCT Coeffs: 5
☐ DFT No of Highest DFT Coeffs: 5
☐ HT No of Highest HT Coeffs: 5
☐ Wavelet T No of Highest Wavelet T Coeffs: 5
Modelling Coefficients:
☑ AR Model No of AR Model Coeffs: 5
☐ ARMA Model No of ARMA Model Coeffs: 5
☐ SARIMA Model No of SARIMA M. Coeffs: 5
Dissimilarity Distance Metric Features: ☑ Euclidean ☑ Square Euclidean ☑ City Block ☑ Minkowski ☑ Chebyshev ☑ DTW
Compare with: GrandAverag...
Grand Average Computation: ☐ Time Domain Start: 0 End: 0 ☑ Frequency domain Start: 1 End: 48
☑ WFOD

4. Classification
☑ Z-Score Standardization of the features
Classification Matrix size: NaN
Classifier Selection:
☑ Tree Classifier Max No of Splits: 3
☑ LDA Classifier
☑ K-NN Classifier - Euc. No of Neighbours: 10
☑ Naive Bayes Classifier
Validation Method Selection:
☑ K-fold Cross Validation K value: 10
☐ Leave-one-out Cross Validation
☑ Use MC Simulations No of MC sim: 1

5. Results
SCR - Channel 1
Tree Classifier: Acc.: 0 Sens.: 0 Spec.: 0 F-Sc.: 0 M.Corr. C: 0
LDA Classifier: Acc.: 0 Sens.: 0 Spec.: 0 F-Sc.: 0 M.Corr. C: 0
K-NN Classifier: Acc.: 0 Sens.: 0 Spec.: 0 F-Sc.: 0 M.Corr. C: 0
Naive Bayes Classifier: Acc.: 0 Sens.: 0 Spec.: 0 F-Sc.: 0 M.Corr. C: 0

EMG - Channel 2
Tree Classifier: Acc.: 0 Sens.: 0 Spec.: 0 F-Sc.: 0 M.Corr. C: 0
LDA Classifier: Acc.: 0 Sens.: 0 Spec.: 0 F-Sc.: 0 M.Corr. C: 0
K-NN Classifier: Acc.: 0 Sens.: 0 Spec.: 0 F-Sc.: 0 M.Corr. C: 0
Naive Bayes Classifier: Acc.: 0 Sens.: 0 Spec.: 0 F-Sc.: 0 M.Corr. C: 0

ECG - Channel 3
Tree Classifier: Acc.: 0 Sens.: 0 Spec.: 0 F-Sc.: 0 M.Corr. C: 0
LDA Classifier: Acc.: 0 Sens.: 0 Spec.: 0 F-Sc.: 0 M.Corr. C: 0
K-NN Classifier: Acc.: 0 Sens.: 0 Spec.: 0 F-Sc.: 0 M.Corr. C: 0
Naive Bayes Classifier: Acc.: 0 Sens.: 0 Spec.: 0 F-Sc.: 0 M.Corr. C: 0

Respiratory - Channel 4
Tree Classifier: Acc.: 0 Sens.: 0 Spec.: 0 F-Sc.: 0 M.Corr. C: 0
LDA Classifier: Acc.: 0 Sens.: 0 Spec.: 0 F-Sc.: 0 M.Corr. C: 0
K-NN Classifier: Acc.: 0 Sens.: 0 Spec.: 0 F-Sc.: 0 M.Corr. C: 0
Naive Bayes Classifier: Acc.: 0 Sens.: 0 Spec.: 0 F-Sc.: 0 M.Corr. C: 0

Multimodal Classification
Tree Acc.: NaN Tree Sens.: NaN Tree Spec.: NaN Tree Fsc.: NaN Tree MCorr.: NaN
LDA Acc.: NaN LDA Sens.: NaN LDA Spec.: NaN LDA Fsc.: NaN LDA MCorr.: NaN
N.Bayes Acc.: NaN N.Bayes Sens.: NaN N.Bayes Spec.: NaN N.Bayes Fsc.: NaN N.Bayes MCorr.: NaN
K-NN Acc.: NaN K-NN Sens.: NaN K-NN Spec.: NaN K-NN Fsc.: NaN K-NN MCorr.: NaN

Pairwise Pearson Correlation Coefficients | PCA
Ch1 - Ch2: 0 Ch1 - Ch3: 0 Ch1 - Ch4: 0 Ch2 - Ch3: 0 Ch2 - Ch4: 0 Ch3 - Ch4: 0

Save Outputs
☑ Autosave as 'Results' in directory File Name: File Extension: .mat

Figure A.9: A snapshot of the GUI that is made for analyses.

The GUI is capable of properly reading multimodal datasets, preprocessing them with several options, performing feature extraction with a variety of features, classifying and validating the extracted features, outputting the performance measures, performing multimodal classification, and saving the outputs. However, it will be improved even further with the needs and detailed analyses.

Appendix B

CURRICULUM VITAE

Last update: July, 2023

EDUCATION

Years	Degree	Department, University
2017-2023	**Ph.D.**	Biomedical Engineering (Bioelectrical Track), Middle East Technical University
2014-2017	**M.Sc.**	Biomedical Engineering (Bioelectrical Track), Middle East Technical University
2012-2014	**B.Sc.**	Business Management, Anadolu University
2009-2013	**B.Sc. (Main)**	Electrical and Electronics Engineering, Hacettepe University

WORK EXPERIENCE

PROFESSIONAL EXPERIENCE

Years	Position and Role	Company/University and Department
2022-ongoing	Senior AI Research Engineer Research and Compliance Manager	Huawei Telecommunications Co. Ltd., Turkey R&D Center, Intelligent Application DC Department
2014-2022	Research and Teaching Assistant	Middle East Technical University (METU), Biomedical Engineering

TEACHING EXPERIENCE

As Teaching Assistant

- BME 501 – Introduction to Biomedical Engineering, METU (2016 - 2022)
- BME 502 – Human Physiology, METU (2016 - 2022)
- BME 590-591-690-691 Seminars, METU (2014 - 2022)

RESEARCH INTERESTS and PROFICIENCY

- **Time series data analysis** (2009 - ongoing) – All sorts, but highest experience in biomedical data analyses.
    - Anomaly and outlier detection (2018 - ongoing)
- **Machine learning** (2014 - ongoing) – Specialized in feature extraction and feature engineering for classification and forecasting (for time series).
    - Feature engineering (2014 - ongoing)
    - Deep learning (2021 - ongoing)
    - Recommender systems and Ranking (2022 - ongoing)
- **Biomedical signal analysis and medical imaging** (2010 - ongoing) – Have experiences of signal analyses with data modalities in different projects, studies and personal interests (from highest to lowest): ECG, EEG, EMG, HRV, pupil diameter, SCR (GSR), fNIRS, blood tests (biochemistry measures, haemogram), fMRI.
- **Data modelling** (2015 - ongoing) – Statistical data modelling and analyses.
- **Graph theory and connectivity analyses** (2015 - ongoing) – Specialized in multimodal biomedical signals. But also conducted studies on economics (Stock market analyses).
- **Natural language processing** (2022 - ongoing) – Working on frequency of frequency estimations of the word distributions in corpuses.

RESEARCH

PROJECTS

- **Researcher**, January 2020 - February 2022, Scientific and Technological Research Council (TÜBİTAK- 1003 Programme). Project title: Development of hardware and software infrastructure for physiological human data for use in human-machine applications. Project no: 117E650.
- **Researcher**, January 2020 - February 2023, Scientific Research Project (BAP1), METU. Project title: Approximate stochastic simulation algorithms in protein-protein interaction networks. Project no: BAP-10282.

PUBLICATIONS IN JOURNALS

- **Erkuş, E. C.** and Purutçuoğlu, V. (2023) "A New Collective Anomaly Detection Approach Using Pitch Frequency and Dissimilarity: Pitchy Anomaly Detection (PAD)". *Journal of Computational Science*, 102084. https://doi.org/10.1016/j.jocs.2023.102084.
- **Erkuş, E. C.** and Purutçuoğlu, V. (2021) "Outlier Detection and Quasi-periodicity Optimization Algorithm: Frequency Domain Based Outlier Detection (FOD)". *European Journal of Operational Research*, 1-15. https://doi.org/10.1016/j.ejor.2020.01.014.
- **Erkuş, E. C.**, V. Purutçuoğlu, and E. Purutçuoğlu. (2019) "Detection of

abnormalities in heart rate using multiple Fourier transforms." International Journal of Environmental Science and Technology. 16, 5237–5242. https://doi.org/10.1007/s13762-019-02252-3.

- Rasoulzadeh, V., **Erkuş, E. C.**, Yogurt, T. A., Ulusoy, I., & Zergeroğlu, S. A. (2017). A comparative stationarity analysis of EEG signals. Annals of Operations Research, 258(1), 133-157.

CHAPTERS IN BOOKS

- Arı, F., Akan, E., Aktaş-Dinçer, H., **Erkuş, E. C.**, Farzin M., Gökçay, D., İleri, F., Purutçuoğlu, V., Somuncuoğlu, A., (2022) Evaluation of Data Compression Methods for Efficient Transport and Classification of Facial EMG Signals, Chapter In: Operations Research: New Paradigms and Emerging Applications, Editors: V. Purutçuoğlu, G.W. Weber, H. Farnoudkia, CRC Taylor and Francis, 2022 (In print).
- **Erkuş, E. C.** and Purutçuoğlu, V. (2021) "Outlier detection in biomedical data: ECG focused approaches". Chapter in: *Advanced Optimization, Optimal Control and Meta-heuristics for Operational Research, Analytics and Decision-Making in Emerging Markets*. Elsevier (In print).

PUBLICATIONS IN PROCEEDINGS

FULL PAPERS

- Yıldız, A., Er, M. E., Gencer, M. and **Erkuş, E. C.** (2023) "An Investigation of the Quality Assurance Approaches and Performance Evaluation in Break Point Anomaly Detection", The 11th International Conference on Advanced Technologies (ICAT 2023), İstanbul, Turkey.
- Özer, M. A., Kaplan, E., Özbey, C., Çetiner, M. and **Erkuş, E. C.** (2023) "An Investigation of the Usage of Dynamic Time Warping in String Similarity Estimation", The 11th International Conference on Advanced Technologies (ICAT 2023), İstanbul, Turkey.
- Aciksöz, S., Altınok, H., Bursalı, A. and **Erkuş, E. C.** (2023) "Dissimilarity Metric Score Estimation for Time Series with Missing Values", The 11th International Conference on Advanced Technologies (ICAT 2023), İstanbul, Turkey.
- Yıldız, A., Er, M. E., Bursalı, A., Çolakoğlu, T. and **Erkuş, E. C.** (2023) "The Effects of Data Standardization and Normalization Techniques in Ranking Performances", The 11th International Conference on Advanced Technologies (ICAT 2023), İstanbul, Turkey.
- Özbey, C., and **Erkuş, E. C.** (2023) "Extending Heaps' Law for Sublinear Vocabulary Growth in a Logarithmic Scale, The 31th 31. IEEE Sinyal İşleme ve İletişim Uygulamaları Kurultayı (SIU 2023), İstanbul, Turkey.
- Özbey, C., Çolakoglu, T., Bilici, M. Ş. and **Erkuş, E. C.** (2023) "A Unified Formulation for the Frequency Distribution of Word Frequencies Using the Inverse Zipf's Law", The 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2023), Taipei, Taiwan.
- **Erkuş, E. C.**, Er, M. E., Yıldız, A. and Gencer, M. (2022) "An Investigation of the

Effects of the Numerical Missing Value Imputation Methods for Click-through Rate Estimation Performances", The 11th International Symposium on Digital Forensics and Security (ISDFS 2023), Tennessee, USA.

- **Erkuş, E. C.** and Purutçuoğlu, V. (2022) "Anomaly Detection in Sliding Windows Using Dissimilarity Metrics in Time Series Data", 4th International Conference on Artificial Intelligence and Applied Mathematics in Engineering (ICAIAME 2022), Baku, Azerbaijan.
- **Erkuş, E. C.** and Purutçuoğlu, V. (2023) "A New Frequency Domain and Dynamic Time Warping Based Feature: WFOD Feature", Proceeding of 2nd International Conference on Advanced Information Scientific Development (ICAISD 2021), Jakarta, Indonesia.
- **Erkuş, E. C.** and Purutçuoğlu, V., Arı, F. and Gökçay, D. (2020) "Comparison of several machine learning classifiers for arousal classification: A preliminary study", Proceeding of Medical Technologies Congress (TIPTEKNO 2020), Virtual International Congress, Turkey.
- **Erkuş, E. C.** and Purutçuoğlu, V. (2019) "Feature extraction of hidden oscillation in ECG data via multiple-FOD method". 47-56. *Artificial Intelligence and Applied Mathematics in Engineering Problems*. Editors: D. J. Hemanth, U. Kose. Springer. https://doi.org/10.1007/978-3-030-36178-5_5
- Gökçay, D., Arı, F., Avenoğlu, B., İleri, F., **Erkuş, E. C.**, Balık, M., ... & Hacıhabiboğlu, H. (2019, July). Preliminary Results in Evaluating the Pleasantness of an Interviewing Candidate Based on Psychophysiological Signals. In 15th International Summer Workshop on Multimodal Interfaces (p. 45).
- **Erkuş, E. C.** and Purutçuoğlu, V. (2019) "Description of Turkish construction sector via İstanbul stock market data", Proceeding of the 5th International Conference on Natural and Engineering Sciences (ICNES 2019), İstanbul, Turkey.
- **Erkuş, E. C.** and Purutçuoğlu, V. (2019) "Calculation of Optimal Number of Monte Carlo Runs for Normally Distributed Datasets", International Conference on Applied Analysis and Mathematical Modeling (ICAAMM 2019), İstanbul, Turkey.
- **Erkuş, E. C.** and Purutçuoğlu, V. (2018) "Two-stage outlier detection algorithm based on Fourier transform: Real data applications", Proceeding of the International Conference on Innovative Engineering Applications (CIEA 2018), Sivas, Turkey.
- **Erkuş, E. C.**, & Ulusoy, İ. (2016, November). "The interpretation of the effective connectivity maps obtained by using Dynamic Bayesian Networks on EEG data". In 2016 20th National Biomedical Engineering Meeting (BIYOMUT) (pp. 1-4). IEEE.

PUBLISHED ABSTRACTS and PRESENTATIONS

- **Erkuş, E. C.** and Purutçuoğlu, V. (2020) "Dinamik Zaman Işınlaması (DTW) Yönteminin Motor Görev fNIRS Verilerinin Sınıflandırılmasında Özellik Olarak Kullanımının Etkisi", Poster Presentation in 18. Ulusal Sinirbilim Kongresi (18. USK), Ankara, Turkey.
- **Erkuş, E. C.** and Purutçuoğlu, V. (2019) "Detection of Hidden Patterns in Time Series Data via Multiple-time FOD Method", Proceeding of the 30th European Conference on Operational Research (EURO 2019), Dublin, Ireland.

- **Erkuş, E. C.** and Purutçuoğlu, V. (2018) “Detection of abnormalities in heart rate using multiple Fourier transforms”, Proceeding of the 5th International Conference on Computational and Experimental Science and Engineering (ICCESEN 2018), Antalya, Turkey.
- **Erkuş, E. C.** and Purutçuoğlu, V. (2018) “A New R Programming Package for the Detection of Outliers in Univariate Time Series Data”, International Conference on Applied Mathematics in Engineering, Balıkesir, Turkey. 27th June 2018, pp.165.
- **Erkuş, E. C.** and Purutçuoğlu, V. (2018) “Outlier Detection Methods for Time Series Datasets”, 4th International Researchers, Statisticians and Young Statisticians Congress, İzmir, Turkey. 28-30th April 2018, pp.94
- **Erkuş, E. C.**, Purutçuoğlu, V. and Ağraz, M. (2017) “Detection of outliers using Fourier transform”, Proceeding of the 10th International Statistics Congress (ISC2017), Ankara, Turkey.
- **Erkuş, E. C.**, Ulusoy İ., (2016, November) “EEG Verilerinde Güç ve Spektrum Analizleri Kullanılarak Farklı Frekans Bantlarında Disleksi Hastalığı Olan Bireylerin Tespit Edilmesi”, In 2016 20th National Biomedical Engineering Meeting (BIYOMUT).
- **Erkuş, E. C.**, Rasoulzadeh V., Ulusoy İ., (2016) “The anatomical comparison of the EEG data collected from dyslexic and control groups using functional and effective brain connections obtained by Partial Directed Coherence (PDC) and Dynamic Bayesian Networks (DBN)”. Poster presentation in 2016 14th National Neuroscience Congress (Best poster award)

LANGUAGES

- **Spoken:** Turkish (native), English (fluent-professional), German (intermediate), Russian (Beginner), Spanish (Beginner).
- **Programming**: MATLAB (Expert – actively using), Python (Advanced – actively using).