UNIVERSIDAD DE CÁDIZ

TESIS DOCTORAL

Automatic Feature Extraction for Time Series Analysis using Deep and Machine Learning

Análisis de series temporales mediante enfoques de aprendizaje profundo y aprendizaje automático

Autor:

Fatima Sajid Butt

Directores:

Jörg Schäfer

David Gómez-Ullate Oteiza

Escuela Superior de Ingeniería

Programa de Doctorado en Ingeniería Informática

Fecha: June 14, 2024

Conformidad de los Directores

David Gómez-Ullate Oteiza, profesor del Departamento de Ingeniería Informática de la Universidad de Cádiz, y Jörg Schäfer, profesor del Departamento de Ciencias de la Computación e Ingeniería y Director del Programa de Máster en Sistemas de Alta Integridad de la Frankfurt University of Applied Sciences, siendo Directores de la Tesis titulada *Automatic Feature Extraction for Time Series Analysis using Deep and Machine Learning*, realizada por Fatima Sajid Butt y enmarcada en el Programa de Doctorado en Ingeniería Informática, para proceder a los trámites conducentes a la presentación y defensa de la tesis doctoral arriba indicada, informan que se autoriza la tramitación de la tesis.

Los directores de tesis

Jörg Schäfer David Gómez-Ullate Oteiza

Frankfurt am Main, 14 de junio de 2024

To Allah the Almighty, for everything
To Papa and Ammi, forever indebted to your love and kindness,
To Amir, for your (mostly)unstaggering support,
And to Soha and Ibrahim, part of my heart and my motivation, without whom I would have completed this thesis two years ago :)

Acknowledgements

"Keep away from people who belittle your ambitions. Small people always do that, but the really great make you feel that you, too, can become great."–Mark Twain

I would like to extend my immense appreciation and gratitude towards my amazing thesis supervisor, **Jörg Schäfer** who has been a source of constant learning, motivation and unwavering support for me during during my Ph.D. I have learnt some of the most valuable lessons both in the capacity of a researcher and in the capacity of a leader from him. He has taught me to never accept any results without question and the difference between good and great scientific practices. The straightforwardness and kindness he has shown while communicating feedback is exceptional. His openness to new research ideas and directions is one of the major reasons I have enjoyed my research journey so much. Thanks to his counselling, I have learned to appreciate and at the same time critically re-evaluate everything scientifically. I look forward to many future collaborations with you.

It is difficult as such to find one gracious mentor but how about having two of them? **Matthias F. Wagner** has been that mentor and collaborator throughout my Ph.D. program and, before that during my Masters thesis as well. His expertise, encouragement, and mentorship have been instrumental in shaping the direction of this work and fostering my academic as well as personal growth. I am forever indebted to his intellectual insights and efforts invested in my academic and personal extension. His advises and approaches have contributed significantly to enhancing the value of my research and broadening my perspective both intellectually and personally. Matthias has shown me, among many other things, about how to be a profound scientist along with being a compassionate human being. Looking forward to have many more insightful pragmatic discussions along the banks of Rhine river.

I would also like to express my gratitude to **David Gómez-Ullate Oteiza**, my co-supervisor, for his honest opinion and valuable inputs on the results and direction of the work. He has always provided significant feedback and a fresh perspective to the research results and future goals. His input has been vital in designing experiments with public datasets.

I am grateful to my tutor and collaborator **Inmaculada Medina-Bulo** for her continuous guidance and support during my Ph.D. She has always welcomed all the questions and issues with an open smile. It is only due to her warm reception and encouragement that Cádiz feels like second home.

I would also like to extend my gratitude towards my esteemed collaborator **Dirk Stegelmeyer** from whom, I have learnt a lot about industrial processes and its challenges in Germany. Our frequent discussions about industrial practices, human behaviours and use-case analysis has provided me with a original outlook for the real world operations.

I would also like to thank my fellow researchers both at UCASE research group,WSN-IOT and at INDAS. I also want to express my appreciation **Luigi La Blunda** at WSN-IOT research group, who provided valuable insights and input along with a lot of support to the initial human activity using ECG signals experiments. I am grateful to **Juan Boubeta-Puig** at UCASE Software Engineering Research Group for his collaboration and fruitful discussions. Their collective expertise and intellectual contributions have enriched my research experience.

I extend my heartfelt appreciation to all my masters and bachelor thesis students. The discussions and insights with each of them lead me to identify, target and reform the new issues in both theory and practise. Special thanks to **Kylie Pusch** who contributed some ideas in her bachelor thesis which contributed as the experimental setup for CNN-LSTM experiements. I would also like to thank both industrial partners **Bühler Leybold optics/Bühler Alzenau GmbH and Optovision Moderne Brillenglastechnik GmbH** for their data and the domain insights.

I am forever indebted to my teachers in Pakistan International School, Jeddah, Professors during my bachelors in the University of Punjab and during Masters in Frankfurt University of Applied Sciences. My heartfelt appreciation for their tireless efforts in imparting knowledge and instilling in me a thirst for knowledge that continues to drive my pursuit of scholarly endeavors.

I would also like to thank my thesis committee members who have taken the time out to evaluate and give their valuable feedback to my dissertation.

Nothing can beat the support I had from my family. I would like to start by thanking my parents who have provided me with the best they had to offer and taught me all the good things I know. My father **Sajid Pervaiz Butt** has been the source of my perseverance and the reason I work so hard. As it is said, 'on my best days, I am my fathers daughter'. You have taught me that honesty, truth and hard work takes one further than the talent. My mother **Tahira Jabeen**, the kindest woman I know and my source of inspiration and comfort. I am writing this dissertation because of your continuous encouragement, sacrifices and prayers.

I am indebted to my siblings who have been my cheerleaders for as long as I can remember. **Khadeja Yasir**, my sister has encouraged me at every low and cheered me up at every high of my career. Many thanks for your unconditional support and patience with

me. I cannot express how much your support means to me. My brother, **Shahrukh Sajid**, has been really supportive of everything I have ever done. Thank you for everything ; from picking and dropping me to the university to accompanying me to Islamabad to sitting hours in different offices for all the bureaucratic hell I made you went through.

I could not have done this Ph.D. without the moral and personal support from my mother in law, **Yasmin Arshad**. You have been at the back of every decision I made. It is because of you, I could work late hours without any worry. You have been there for me and my family unconditionally and I am forever obliged to you for that.

The two persons who have adjusted and sacrificed the most during my Ph.D. journey were my children **Soha** and **Ibrahim**. It was them who had to constantly reschedule things around my schedule and had to cancel plans last minute because I had to do some 'more experiments'. I will try to make it up for those times, I promise. You are my source of motivation, love and inspiration.

At the end I would like to thank my person, **Amir Saqib Ali Khan**. You have always encouraged me to chase my dreams even if it came at the expense of your own. I do not say it enough but thank you for being patient with me and to be the source of calmness in my life. You have been my backbone and support in all the endeavours and difficult paths I have tried to tread. This thesis is a testament to your love, support, and encouragement. I am forever grateful for your presence in my life and the countless ways you have shaped my journey. I am excited to start the next chapter of our life where hopefully I am not that anxious and irritated all the time :).

March 2024

Fatima Sajid Butt

"He (Moses) said : O my Lord! Expand me my heart (for knowledge), and ease my task for me and remove the impediment from my speech, so they may understand what I say"

(Quran, 20:25-28)

"By far, the greatest danger of Artificial Intelligence is that people conclude too early that they understand it."

– Eliezer Yudkowsky, co-founder and research fellow at the Machine Intelligence Research Institute

Abstract

Time series analysis is crucial in understanding and extracting valuable insights from temporal data, capturing the inherent patterns, trends, and dependencies that evolve over time. This study concisely overviews the state of the art key components and methodologies involved in time series analysis and classification for both industrial and Electrocardiogram (ECG) signals using deep learning approaches like CNNs, LSTMs and transformers etc.

The analysis begins with a use case study of industrial collaboration with the lens manufacturing industry where the process models called CRISP-DM and DMME for industrial data science are implemented to an industrial production dataset. The collaboration improved industrial production by decreasing the downtime for cleaning the machinery. The study also delves into the shortcomings of the process models and the actual challenges faced during the implementation of data science in real industrial projects. Furthermore, the study explores fundamental concepts such as the methodology of analyzing ECG in particular and time series in general. To lay the groundwork for subsequent advanced techniques of classification by deep learning, the basic construct of the human heart and ECG signals is also presented in detail.

The next vital TSA subject probed was fall detection using ECG signals. The paper introduces a novel approach using electrocardiogram (ECG) signals for fall detection and activity classification. An algorithm employing pre-trained convolutional neural networks (AlexNet and GoogLeNet) as classifiers is proposed, achieving a significant validation accuracy of 98.08% for distinguishing between fall and no-fall scenarios in the first model. The signals are pre-processed to reduce noise, and frequency-time representations (scalograms), which are obtained through continuous wavelet transform, serve as feature extractors. The trained model accurately distinguishes ECGs with fall activity from those without at an accuracy of 98.02%. The robustness of the algorithm is verified by augmenting the experimental dataset with publicly available datasets, achieving a classification accuracy of 98.44% in the second model, which classifies fall, daily activities, and no activities. The models, developed through transfer learning from real images to medical images, offer a lightweight solution compared to traditional deep learning approaches, avoiding redundant computational efforts.

In recent studies on electrocardiogram (ECG) signal classification using deep learning (DL), the focus has been on complex DL methods like transfer learning or feature extraction based on domain knowledge. As the next steps, this study challenges the common assumption that deeper and more complex DL models lead to better learning. Instead, the authors propose two novel DL models: a CNN-LSTM hybrid and an attention/transformer-based model with wavelet transform for dimensional embedding. These models extract features from ECG signals in the initial layers, demonstrating performance on par with or surpassing many contemporary deep neural networks. Validation using three publicly available datasets shows benchmark accuracy, reaching 99.92% for fall detection and 99.93% for PTB database classification of myocardial infarction versus normal heartbeat.

While transformer models have demonstrated superior performance in natural language processing, their careful adoption is essential for achieving comparable results in the realm of time series classification. This study, to the best of our knowledge, for the first time, explores the impact of various dimensional embedding techniques in time series classifications. The exploration includes the use of wavelet transformation, discrete and continuous wavelets, scattering, and feature maps from convolutional neural networks for performance comparison. Several ECG datasets from UCR dataset and Physionet are employed for both multi-class and binary classification. The experimental results consistently reveal that incorporating relevant feature extraction techniques as dimensional embedding outperforms a plain transformer approach.

In conclusion, this dissertation offers a comprehensive overview of the multifaceted realm of automatic feature extraction for time series classification, serving as a guide for researchers, practitioners, and enthusiasts seeking to work with temporal data and harness its capability for informed decision-making.

UNIVERSIDAD DE CÁDIZ

Resumen

Escuela Superior de Ingeniería
Departmento de Ingeniería Informática

por Fatima Sajid Butt

El análisis de series temporales es crucial para comprender y extraer información valiosa sobre datos temporales, capturando patrones inherentes, tendencias y dependencias que evolucionan con el tiempo. En este estudio se describen de forma concisa los principales componentes y las metodologías más recientes que se aplican en el análisis y la clasificación de series temporales, prestando atención particular a series provenientes de procesos industriales y señales de electrocardiogramas (ECG) y empleando enfoques de aprendizaje profundo como redes convolucionales (CNN), redes recurrentes (LSTM), arquitecturas de tipo *transformer*, etc. El análisis comienza con un estudio de colaboración industrial con la industria de fabricación de lentes donde los modelos de proceso CRISP-DM y DMME, que se originan en la ciencia de datos aplicada a procesos industriales, se aplican a un conjunto de datos de producción de lentes. La colaboración mejoró la producción industrial al disminuir el tiempo de inactividad para la limpieza de la maquinaria. El estudio también profundiza en las deficiencias de los modelos de procesos y en los retos reales a los que se enfrenta la aplicación de la ciencia de datos en proyectos industriales reales. Además, el estudio explora conceptos fundamentales como la metodología de análisis de ECG en particular y de series temporales en general. Para sentar las bases de las posteriores técnicas avanzadas de clasificación mediante aprendizaje profundo, también se presenta en detalle la construcción básica de modelos del corazón humano y la obtención de señales de ECG.

El siguiente tema abordado en análisis de series temporales fue la detección de caídas mediante señales de ECG. El artículo presenta un enfoque novedoso que utiliza señales de electrocardiograma (ECG) para la detección de caídas y la clasificación de la actividad. Se ha desarrollado un algoritmo que emplea redes neuronales convolucionales preentrenadas (AlexNet and GoogLeNet) y que obtiene una precisión del 98% en el problema de clasificación binaria entre escenarios de caída / no caída. Las señales temporales de ECG son preprocesadas para eliminar ruido y pasar a una representación en frecuencias (escalogramas) que se obtiene mediante la aplicación de transformadas de ondículas (*wavelets*) contínuas y suponen una extracción de características novedosa. La robustez del algoritmo se valida al ampliar el dataset experimental con nuevas fuentes de datos públicas, llegando a una precisión de 98.4% en el segundo modelo que clasifica actividades en tres clases: caída, reposo y actividades diarias. Estos modelos se han desarrollado mediante la aplicación de aprendizaje por transferencia (*transfer learning*) sobre modelos desarrollados para imágenes reales (ImageNet) para adaptarlos a imágenes médicas, lo cual supone una mejor eficiencia en el entrenamiento y coste computacional de nuestra solución.En estudios recientes sobre clasificación de señales de electrocardiograma (ECG) mediante aprendizaje profundo (DL), la atención se ha centrado en métodos de DL complejos como el aprendizaje por transferencia o la extracción de características basada en el conocimiento específico. Como próximos pasos, este estudio desafía la suposición común de que los modelos de DL más profundos y complejos conducen a un mejor aprendizaje. En su lugar, los autores proponen dos nuevos modelos de DL: un modelo híbrido CNN-LSTM y un modelo basado en el mecanismo de atención con transformada *wavelet* para *embedding* dimensional. Estos modelos extraen características de las señales de ECG en las capas iniciales, demostrando un rendimiento a la par o superior al de muchas redes neuronales actuales. La validación con tres conjuntos de datos públicos muestra una precisión en los benchmarks estudiados que alcanza el 99,92% para la detección de caídas y del 99.93% para el problema de clasificación de señales correspondientes a infarto de miocardio frente a señales normales.

Si bien es cierto que los modelos *transformer* han demostrado un rendimiento superior en procesamiento del lenguaje natural, es esencial prestar atención a ciertos detalles en su aplicación para lograr resultados comparables en el ámbito de la clasificación de series temporales. Hasta donde sabemos, este trabajo es el primero en explorar el impacto de varias técnicas de *embedding* dimensional en la clasificación de series temporales. La exploración incluye el uso de la transformada wavelet, wavelets discretas y continuas, dispersión y mapas de características en redes neuronales convolucionales para la comparación del rendimiento. El estudio aborda clasificación binaria y multi-clase en dos conjuntos de datos de ECG provenientes de UCR y Physionet. Los resultados demuestran de manera consistente que la incorporación de técnicas de extracción de características

como el *dimensional embedding* mejoran el rendimiento de los modelos *transformer*. En resumen, este trabajo ofrece una panorámica sobre el ámbito de la extracción automática de características para abordar problemas de clasificación de series temporales, siendo de interés para investigadores y usuarios que trabajen con datos temporales para mejorar su toma de decisiones.

Contents

Conformidad de los Directores ii

Acknowledgements iv

Abstract xi

Resumen xiii

List of Figures xxii

List of Tables xxv

Abbreviations xxvii

1 Introduction 1
- 1.1 Motivation . . . 3
- 1.2 Research Questions and Goals . . . 4
- 1.3 Contributions . . . 5
- 1.4 Publications . . . 6
- 1.5 Structure of the PhD Thesis . . . 7

2 Time Series Analysis 11
- 2.1 Probability Spaces and Time Series . . . 11
- 2.2 Time Series Analysis . . . 12
- 2.3 Time Series Classification . . . 13
- 2.4 Bayes Decision Theory . . . 15

3 Time Series Analysis - An Industrial Application 18
- 3.1 TSA for Industry 4.0 . . . 18
- 3.2 Background and Motivation . . . 21
- 3.3 Related Work and Our Contribution . . . 22
- 3.4 Process Models: CRISP-DM and DMME . . . 24
  - 3.4.1 Business Understanding . . . 24
  - 3.4.2 Technical Understanding and Conceptualization . . . 25
    - 3.4.2.1 Determine Technical Objectives . . . 25
    - 3.4.2.2 Analysis of the Technical Situation . . . 26
    - 3.4.2.3 Conceptualization . . . 27

3.4.2.4 Acquisition Concept . . . 27
3.4.2.5 Experimental Planning . . . 28
3.4.2.6 Specification and Project Plan . . . 28
3.4.3 Technical Realization and Testing . . . 28
3.4.3.1 Prototype Realization . . . 28
3.4.3.2 Test of the Concept . . . 28
3.4.3.3 Perform Experiments and Collect Data . . . 29
3.4.3.4 Documentation . . . 29
3.4.4 Data Understanding . . . 30
3.4.5 Data Preparation . . . 30
3.4.6 Model Building . . . 31
3.4.7 Model Evaluation . . . 32
3.4.8 Results . . . 33
3.5 Case Study Insights . . . 33
3.5.1 Business Case . . . 34
3.5.2 Data Quality Description . . . 34
3.5.3 Role Description . . . 35
3.6 Discussion and Future Work . . . 35
3.7 From Industrial Time Series to Bio-medical Time Series . . . 35
3.8 Conclusion . . . 36

4 ECG Signals - Construct and Analysis 38
4.1 Time Series Classification . . . 38
4.2 ECG Signals . . . 38
4.2.1 Structure And Physiology Of Human Heart . . . 39
4.2.2 Cardiac Muscles And Electrical Activity . . . 39
4.2.3 Sinoatrial (SA) Node . . . 40
4.2.4 Atrioventricular (AV) Node . . . 41
4.3 The Cardiac Cycle . . . 41
4.3.1 Phases Of The Cardiac Cycle . . . 41
4.3.2 Atrial Systole And Diastole . . . 42
4.3.3 Ventricular Systole . . . 43
4.3.4 Ventricular Diastole . . . 43
4.4 The Normal Electrocardiogram . . . 44
4.4.1 The ECG Electrodes And Leads . . . 45
4.4.2 Einthoven's Triangle . . . 45
4.5 Systematic Methodology of ECG Analysis – An Overview . . . 46
4.5.1 Feature Extraction . . . 48
4.5.2 Feature Selection . . . 49
4.5.3 Feature Transformation . . . 49
4.5.4 Classification . . . 52
4.5.5 Explanation . . . 55

5 Fall Detection Using ECG Signals 59
5.1 Introduction . . . 60
5.2 Background . . . 60
5.3 Related Work and Our Contribution . . . 61

5.4 Experimental Setup and Data Collection . . . 63
5.4.1 Inclusion Criteria . . . 64
5.4.2 The HAR Experiment . . . 65
5.4.3 The Collected Data . . . 65
5.5 Our Methodology and Implementation Protocol . . . 66
5.5.1 Proposed Algorithm . . . 66
5.5.2 ECG Signal Filtering . . . 67
5.5.2.1 IIR versus FIR . . . 68
5.5.2.2 Elliptical Filter . . . 70
5.5.3 Implementation Protocol: Hardware And Software . . . 70
5.5.4 Time-Frequency Representations . . . 70
5.5.5 Phases of Our Implementation . . . 71
5.5.5.1 Training The Network : Phase I . . . 71
5.5.5.2 Preparing and Training the Model . . . 71
5.5.5.3 Tuning The AlexNet . . . 73
5.5.5.4 Explainability: Activation's of Different Layers In CNN . 73
5.5.6 Extension of the Algorithm: Phase II . . . 75
5.5.6.1 Data Augmentation and its Challenges . . . 75
5.5.6.2 Tuning the GoogLeNet . . . 76
5.5.6.3 Transfer Learning to the Rescue: GoogLeNet and AlexNet 77
5.5.6.4 k-fold Verification . . . 78
5.6 Analysis of Results . . . 79
5.6.1 Analysis of Fall Vs No-Fall ECG Signals . . . 79
5.6.2 Analysis of Scalograms . . . 82
5.7 Discussion of the Research Question . . . 82
5.8 Conclusion and Future Work . . . 83

6 Towards Automated Feature Extraction 85
6.1 Motivation . . . 86
6.2 Related Work and Our Contribution . . . 87
6.2.1 CNN-LSTM Architectures . . . 88
6.2.2 Attention and Transformer Architectures . . . 89
6.3 Algorithms . . . 90
6.3.1 CNN-LSTM Model . . . 90
6.3.1.1 CNNs and LSTMs . . . 90
6.3.1.2 CNN-LSTM Architecture and Algorithm . . . 92
6.3.2 Attention Model . . . 93
6.3.2.1 Attention . . . 93
6.3.2.2 Attention and Dimensional Embedding . . . 94
6.3.2.3 Transformer Architecture and Algorithm . . . 96
6.3.3 Complexity Analysis . . . 97
6.3.3.1 Runtime Complexity Analysis . . . 97
6.3.3.2 Memory Complexity Analysis . . . 98
6.4 Data Preparation and Experimental Setup . . . 98
6.4.1 ECG Data Set for Fall Detection . . . 99
6.4.2 PTB Diagnostics Data Set . . . 99
6.4.3 PTB XL Diagnostics . . . 100

6.5 Results . . . 100
6.5.1 ECG HAR Data Set . . . 101
6.5.2 PTB Diagnostics . . . 102
6.5.3 PTB XL Diagnostics . . . 105
6.5.4 Statistical Analysis . . . 106
6.5.4.1 McNemar's test . . . 107
6.5.4.2 The 5x2 cv t test . . . 108
6.5.4.3 Interpretation . . . 109
6.5.5 Summary . . . 109
6.6 Discussion . . . 109
6.7 Conclusion . . . 111
6.7.1 Data and Code Availability . . . 112
6.7.2 5x2 cv t-test table . . . 112

7 Feature Extraction using Wavelet Transformation 114
7.1 Feature Extraction for Time Series and Bio Medical Signals . . . 114
7.2 Wavelets – An Introduction . . . 115
7.2.1 Discrete Wavelet Transformation (DWT) for Dimensional Embedding . . . 117
7.3 Wavelets as Feature Extractors . . . 118
7.3.1 Groups and Group Representations . . . 119
7.3.2 Quantization . . . 121
7.3.3 Admissible Embeddings . . . 121
7.3.4 Wavelets as Admissible Embeddings . . . 122

8 Dimensional Embeddings for TSC in Transformers 126
8.1 Introduction . . . 126
8.2 State of the Art and Our Contribution . . . 127
8.2.1 Transformers and Attention Mechanism . . . 128
8.3 Dimensional Embeddings . . . 129
8.3.1 Transformers and Embeddings for Time Series . . . 130
8.4 Proposed Architecture and Experimental Setup . . . 132
8.5 Datasets . . . 134
8.6 Results . . . 135
8.7 Discussion . . . 137
8.8 Future Work and Conclusion . . . 137

9 Findings with Respect to the Research Questions 140
9.1 Are existing deep learning methods for time series classification as effective as they are for other domains like Natural language processing (NLP) and computer vision? . . . 140
9.2 What are the hindrances in the actual realization of Industry 4.0 despite the availability of modern data analysis techniques? . . . 141
9.3 How can data science be effectively used in an industrial setup to decrease production downtime? . . . 141
9.4 How features could be automatically extracted from different time series for classification problems? . . . 142
9.5 Can human activity including fall be detected from ECG signals? . . . 142

9.6 Can wavelet transforms act as feature extractors for time series to be used in deep learning models? . . . 142
9.7 How can transformers be adapted to time series classification? . . . 143
9.8 Does positional encoding play any vital role in TS classification, if we already have a wavelet transformation representation of our TS? . . . 143
9.9 Challenges and Limitations . . . 144

10 Conclusion & Future Work **147**
10.1 Conclusion . . . 147
10.2 Future Work . . . 148

A Mathematical Notations **151**
A.1 Vectors . . . 151
A.2 Hilbert Space . . . 151
A.3 Tensor Product . . . 152
A.4 Sets . . . 152
A.5 Signals . . . 152
A.6 Probability . . . 152

B Code and Data Availability **153**

C Code and Note to the Technology **154**
C.1 Experimental Set up . . . 154
C.2 Libraries . . . 154

Bibliography **158**

List of Figures

1.1 The comparison in an increase in the data generated annually [70] 1
1.2 The conceptual flow of a model generation for time series [117] 2

2.1 Standard Model for Statistical Pattern Classification [192] 14

3.1 Schematic illustration of the four industrial revolutions and projection into Industry 5.0. 19
3.2 Graphical representation of the DMME process [108] 22
3.3 Phases of the reference model of DMME (data mining methodology for engineering applications). (a) State of CRISP-DM, (b) holistic extension for engineering application.[108] 25
3.4 A Coater Machine by Bühler Leybold Optics for Air Coating Lenses. . . . 26
3.5 Correlation matrix displaying the correlation between important different quality parameters 29
3.6 Diagram representing the distribution of (aggregated) pressure between good and bad quality processes. The x-axis represents (aggregated) pressure in milliBar (mbar) and the y-axis represents chamber pressure i.e. pre-vacuum line (mbar). Green stars represent good quality and red bad quality batches. Where 0: $<$ 10 % bad lenses per batch(in green) , 1: $>$70% bad lenses per batch(red) 30
3.7 Flow diagram depicting the initial data cleaning process 31
3.8 Decision tree to identify the threshold for each parameter for different classes 32
3.9 Accuracies of different convolutional neural network models where each model was trained with different input parameters 33

4.1 Structure of the heart, and course of blood flow through the heart chambers and heart valves.[15] 39
4.2 The cardiac cycle starting with atrial systole and progressing to ventricular systole,atrial diastole, and ventricular diastole. Corresponding ECG correlation is highlighted[21] 42
4.3 A normal ECG [140] 44
4.4 Conventional arrangement of electrodes for recording the standard electrocardiographic leads superimposing the Einthoven's triangle[21] 46
4.5 Systematic Methodology for Analyzing a Time Series in general and ECG in particular 47
4.6 An overview of the Feature extraction techniques fro ECG signals 48
4.7 The three principal approaches of feature selection. The shades show the components used by the three approaches: filters, wrappers and embedded methods [93] 50

4.8 A unified deep learning framework for time series classification [111] . . . 53
4.9 The proposed taxonomy categorizes the reviewed XAI approaches in different explanation types based on their explanations [240]. 56

5.1 The System Architecture Diagram for a Data Acquisition System – Hardware and Software Components [135] 63
5.2 The Rollover Fall Process 65
5.3 Examples of Baseline Wander. 69
5.4 Daily Activities 72
5.5 Resting 72
5.6 Fall 72
5.7 Scalograms with different classes of ECG signal in them 72
5.8 Scalogram of Fall ECG and its corresponding activation (Left to right): Figure (a): Scalogram with a distinct Fall. Figure (b): Strongest Activation Of the Image in Layer 5 (`conv5`). Figure (c): Activations in The `conv1` Layer 74
5.9 Scalogram Of Resting ECG: Activation of earlier and deeper Layers : (Left to right): Figure (a): Layer 1. Figure (b): Layer 2 74
5.10 A Graphical Summary of the Training Results for AlexNet Model 78
5.11 Falls in ECG signals in Time Domain 80
5.12 A Closer Look at The Fall Activity in an ECG Signal 80
5.13 A Comparison of different ECG Time Domain Signals with their Corresponding Frequency Spectrum 81

6.1 CNN-LSTM Model for PTB DB 92
6.2 Final CNN-LSTM Architecture for Fall and HAR using ECG signals . . . 93
6.3 Dimensional Embedding 95
6.4 Transformer Architecture of Model att11 96
6.5 Class Distribution for PTB-XL Data Set 101
6.6 Training and Validation Graph over Epochs for the PTB Data Set for Algorithm 3 104
6.7 Training and Validation Loss for the PTB Data Set for Algorithm 4 . . . 105

7.1 Wavelet Transform Pyramid 118

8.1 Input vector of length $d = 6$ is shown along with the linear weighted transformations. The resultant vectors, each of size $s_1 = s = d$ are called Query, key and value. In every sequence handled by the Transformer, there exist n inputs in total, leading to the generation of n query vectors, n key vectors, and n value vectors. 130
8.2 The proposed transformer architecture for classification as compared to an original transformer architecture [249] 133
8.3 The LSTM model(left) and Transformers(right) with feature maps as dimensional embeddings 133
8.4 The transformer model with(left) and without(right) positional encoding and attention mechanism 134

8.5 CSI dataset : The normalized confusion matrix for experiments with (top right) Transformer with feature maps, (top left)transformers with scattering wavelets , (bottom right)scattering wavelet transformation with LSTM 136
8.6 HAR dataset : The normalized confusion matrix for experiments with (top right) Transformer with scattering wavelets, (top left) Plain LSTM , (bottom right)Transformer with feature maps and (bottom left) Scaterring wavelet with LSTMs 136

List of Tables

4.1 Typical lead II ECG features and their normal values in the sinus rhythm at a heart rate of 60 bpm for a healthy male adult [52] . . . 45

5.1 A Summary of the Collected Data . . . 66
5.2 Total Number of Samples Used for Training in Both Iterations . . . 66
5.3 Confusion Matrix for the Selected Trained Model . . . 78
5.4 Summary of GoogLeNet fine-tuned and retrained for Fall Detection using ECG Signals with different Parameters . . . 78
5.5 Summary of AlexNet fine-tuned and retrained for Fall Detection using ECG Signals with different Parameters . . . 79
5.6 An Overview of K-fold Verification . . . 79

6.1 Attention Models . . . 96
6.2 Generic Runtime Complexity Analysis . . . 97
6.3 Runtime Complexity Analysis for Algorithms 3 and 4 . . . 98
6.4 Total Number of Samples in the ECG HAR Data Set [34] . . . 99
6.5 Confusion Matrix for Fall Detection ECG Data Set using CNN-LSTM (Algorithm 3) . . . 102
6.6 Confusion Matrix for Fall Detection ECG Data Set using Attention (Algorithm 4) . . . 102
6.7 Confusion Matrices for PTB Data Set . . . 103
6.8 Parameter Comparison between State of the Art and Our Work. Key for Training Hardware: 1 = 2 NVIDIA Titan Xp GPUs ,2 = 2 NVIDIA 2080Ti GPUs, 3= i5 core, NVIDIA graphics card, 4= NVIDIA A100-PCIE-40GB 103
6.9 Metrics for Leading CNN-LSTM and Attention . . . 104
6.10 Performance Indicators . . . 105
6.11 Five Fold Cross-Validations . . . 106
6.12 Layout . . . 108
6.13 ECG Data Set . . . 108
6.14 PTB Data Set . . . 108
6.15 McNemar's Contingency Tables . . . 108
6.16 Our Result Compared with other similar Studies in Literature which used PTB Database (Built upon [91]) . . . 110
6.17 Overview of the Experiments with Different Data Sets and the Acquired Performances . . . 110
6.18 5x2 cv Test Contingency Table for PTB Data Set . . . 112
6.19 5x2 cv Test Contingency Table for ECG Data Set . . . 112

8.1 An overview of the current literature for time series classification and the corresponding adaptation of dimensional embedding 131
8.2 An overview of the datasets used for the experiments. NA refers to not available 135
8.3 The results for different datasets using different configurations for dimensional embedding with transformers. Key: WoP: Without Positional Encoding, WP: With Positional Encoding, Trans: Transformer, WT: wavelet transform, SW: scattering wavelet, FM: Feature maps, DWT: discrete wavelet transform 135

Abbreviations

ADL	**A**ctivity of **D**aily **L**ife
BLE	**B**luetooth **L**ow **E**nergy
BW	**B**aseline **W**ander
CEP	**C**omplext **E**vent **P**rocessing
CNN	**C**onvolutional **N**eural **N**etwork
CSI	**C**hannel **S**tate **I**information
CSV	**C**omma - **S**eparated **V**alues
CWT	**C**ontinuous **W**avelet **T**ransformation
DCT	**D**iscrete **C**osine **T**ransform
DFT	**D**iscrete **F**ourier **T**ransform
DWT	**D**iscrete **W**avelet **T**ransformation
DE	**D**imensional **E**mbedding
DL	**D**eep **L**earning
ECG	**E**lectro**c**ardio**g**ram
FIR	**F**inite **I**mpulse **R**esponse
HAR	**H**uman **A**ctivity **R**ecognition
Hz	**H**ertz
IIR	**I**nfinite **I**mpulse **R**esponse
IOT	**I**nternet **O**f **T**hing
kNN	**k**- **N**earest **N**eighbour
LSTM	**L**ong **S**hort **T**erm **M**emory
MFCC	**M**el **F**requency **C**epstral **C**oefficients
ML	**M**achine **L**earning
NLP	**N**atural **L**anguage **P**rocessing
NN	**N**eural **N**etwork

PE	**P**ositional **E**ncoding
RBF	**R**adial **B**asis **F**unction
RQ	**R**esearch **Q**uestion
TS	**T**ime **S**eries
SVM	**S**upport **V**ector **M**achine
TSC	**T**ime **S**eries **C**lassification
W.R.T	**W**ith **R**espect **T**o
WT	**W**avelet **T**ransform
XAI	E**X**plainable **A**rtificial **I**ntelligence

Chapter 1

Introduction

'We are drowning in information but starved for knowledge.'– (John Naisbitt)

The modern world is comprised of technology such as cell phones, radio, video, and connected IoT devices. According to a report by Statistica for the data produced daily online in the year 2023 [70], approximately 328.77 million terabytes of data are created each day. In fact, it is estimated that 90% of the world's data was generated in the last two years alone and is increasing (See Fig.1.1). The data generated is an amalgam of videos, audio, text, and time series. This large amount of data would be of no use if relevant and meaningful information could not be extracted from it.

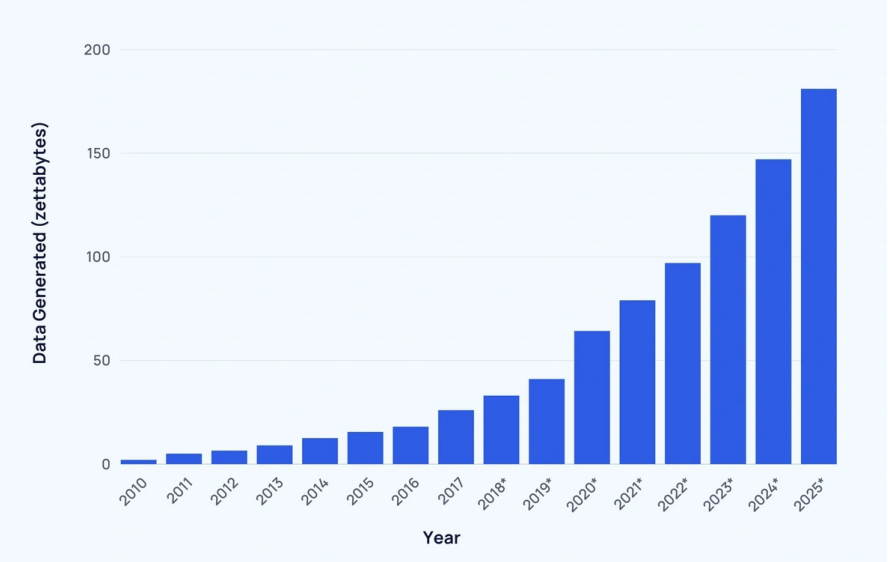


FIGURE 1.1: The comparison in an increase in the data generated annually [70]

As a result of the advancement in ubiquitous sensing, time series data exists from multiple application domains such as entertainment, biotechnology, pharmaceutics, telecommunications etc., to name a few. Since all time-ordered values can be categorized as time

series, they appear organically in many domains. This technology is available to us through primarily signals and signal processing [168].

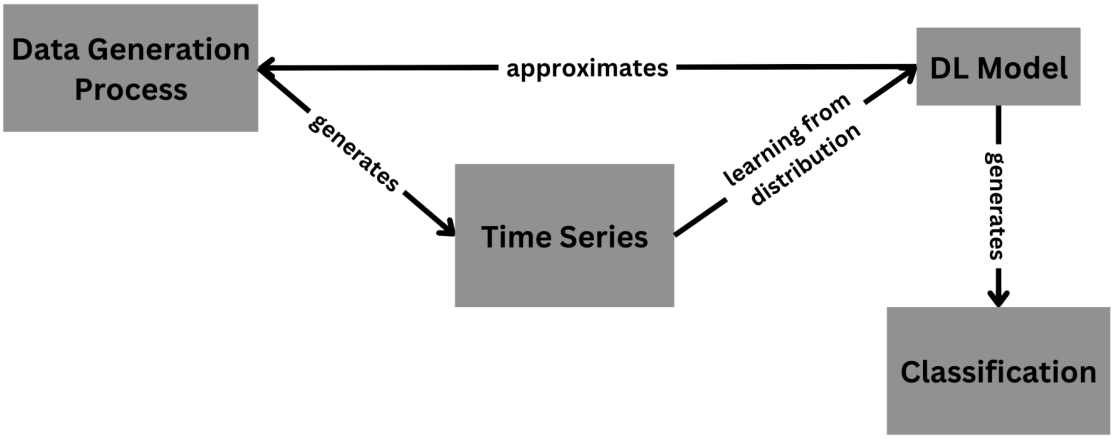


FIGURE 1.2: The conceptual flow of a model generation for time series [117]

Analysing and manipulating time series have been classical problems in the field of signal processing. It has been augmented and boosted by scientific progressions like digitization, acquisition, transfer and storage of the signals. One of the basic activities with time series is classification which is fundamentally a knowledge intensive task [51]. The pipeline from data generation of time series from sensors to classification by DL models is shown in Fig 1.2. The research on representation learning and classification has uncovered numerous potential applications.

The development in the computation devices and popularity of deep learning methods have contributed equally to the more automated models for time series manipulation.

A time series is defined as values ordered with respect to time. A time series can be categorized as univariate and multivariate. Formally a univariate time series X is defined as

$$X = [x_1, x_2, ..., x_T] \tag{1.1}$$

where T is the total number of real values in X. Similarly an N-dimensional multivariate time series,

$$X = [X^1, X^2, ..., X^N] \tag{1.2}$$

which consists of N different univariate time series with $X^i \in \mathbb{R}^T$ [111].

The primary focus of this dissertation is time series classification and automatic feature extraction for the classification task using data based learning and its applications. This chapter introduces the problem statements and the research questions that have motivated this study. It also highlights the motivation for pursuing the targeted research questions in the course of this thesis.

1.1 Motivation

One of the aims of statistical learning is to extract useful information from a corpus of data D through estimating good probabilistic models. There are two approaches to achieving this according to Breiman et. al [30]. In the first traditional approach, data is assumed to be generated from a given stochastic data model. The other approach assumes the data generation mechanism unknown and complex and uses algorithmic models. The focus in the second approach is to find a function $f(\mathbf{x})$, which is an algorithm that operates on $\mathbf{x}$ to predict the responses $\mathbf{y}$. However this is aimed to be achieved in a manner that is as automated as possible. Breiman argued that focus on data modeling has hindered the experts from exploring scientific and commercial fields where data is rendered to be suitable for analysis by data models. In this thesis, we have used both approaches depending on the specific research question.

According to [119], the classical machine learning models are usually created in a three step process: manually pre-process time series, create a model which is domain specific, or use an existing learner with more effective representation. All three steps have their challenges while implementing. Hence, deep learning and feature extraction have to go hand in hand for classifying underlying trends in a particular time series. The temporal dependencies and features which are unique to each time series are not exploited properly in an attribute-value representation which is applied to time series classifiers in classical machine learning methods.

Although deep learning has outperformed its predecessors in the field of vision and natural language processing (NLP), there is still a lack of a unified/general DL model that works best for most time series. Most of the effort in the literature is steered towards adopting existing DL models from NLP or vision towards TS. Another focus in the current literature is to find the appropriate techniques to represent and use time series for its use with deep learning methods. Since no best learning algorithm exists according to Wolpert's 'No Free Lunch' theorem [261], it is always important to base the conclusions conditionally on the experimental protocols and the given data sets.

Understanding and predicting patterns within time series data is crucial across various domains, from industrial production optimization to healthcare and climate science. By delving into time series analysis, the hidden structures and trends can be explored that can drive informed decision-making and provide a deeper understanding of underlying phenomena.

The ability to model and forecast time-dependent behavior empowers professionals to anticipate future trends, identify anomalies, and optimize strategies. The analysis is particularly relevant in scenarios where historical patterns play a pivotal role in shaping

future outcomes, such as stock market predictions, energy consumption forecasts, and epidemiological studies.

Whether it involves optimizing supply chain operations, improving patient outcomes, or aiding in strategic decision-making, time series analysis offers a powerful toolkit for transforming raw temporal data into actionable knowledge. In this study, the time series from different domains were analyzed. We analysed the production data set from eyeglass lens coating machines, Electrocardiogram (ECG) signals, and an air pollution dataset. All of these analyses served different usecases. The production data set from eye lens coating aimed to understand the mechanics of the process and to improve the production up time. The data from ECG is used to human activity recognition and diagnostics. Similarly the pollution dataset aimed to detect the different level of pollutants before they reach a certain level in an enclosed space. All of these usecases can be addressed using appropriate data analytic, machine learning, and deep learning techniques.

Many classical machine learning models exist to classify time series in multiple domains. For more details related to each model please refer to a Bishop [23].

According to [186], the time series classification approaches can be instance based (nearest neighbor (NN), similarity measures like Euclidean distance etc.), feature based (dimension reduction techniques, related feature selection etc.) and symbolic based, support vector based and model based (statistical approximation, generative stochastic methods and probabilistic networks etc).

The focus of this study has also been to find meaningful representations for different time series to be used for classification.

1.2 Research Questions and Goals

The aforementioned motivation leads to many interesting areas of research but following research questions (RQs) had been the primary focus of this dissertation.

- **RQ1:** Are existing deep learning methods for time series classification as effective as they are for other domains like natural language processing (NLP) and computer vision?
- **RQ2:** What are the hindrances in the actual realization of Industry 4.0 even with the availability of modern data analysis methods?
- **RQ3:** How can data science be effectively used in industrial set up to decrease production down time?

- **RQ4:** How features could be automatically extracted from different time series for classification problems?
- **RQ5:** Can human activity including fall be detected from ECG signals?
- **RQ6:** Can wavelet transforms act as feature extractors for time series to be used in deep learning models?
- **RQ7:** How can transformers be adopted to time series classification?
- **RQ8:** Does positional encoding plays any vital role for TS classification, if we already have wavelet transformation representation of our TS?

The corresponding answers and related discussions to these research questions are part of this dissertation and will be discussed during the course of this work. In order to provide a structured overview to the answers of the aforementioned research questions, a dedicated section will contain the research questions including the detailed responses.

The main objectives of this study has been as follows:

- **Objective 1:** Develop a method (a hybrid or ensemble model) which can capture the latent representation effectively from the TS datasets for our specific use case(s).
- **Objective 2:** To verify as a proof of concept that the state of fall from the state of rest and other activities can be detected from the electrocardiogram signals.
- **Objective 3:** To investigate the performance of multiple deep learning models and propose a solution which is computationally and parameterically effective for time series.
- **Objective 4:** To explore and compare different mathematical projection tools like wavelet transforms and polynomials as feature extractors for time series classification.
- **Objective 5:** To classify myocardial infraction from other heart diseases using deep learning tools and methods for an automatic detection.
- **Objective 6:** To deeply analyze dimensional embedding adaptation for time series in transformer model.

1.3 Contributions

The principle contributions of this thesis are listed below along with the corresponding chapter that contain the details about corresponding contributions.

1. The development of a deep learning model using transfer learning for detection of fall from other human activities from electrocardiogram signals (see Chapter 4).
2. The development of a CNN-LSTM algorithm to prove that a similar or better performance can be achieved without manually pre-processing for the ECG signals
3. Using wavelet transformations as embedding in transformers to remove the manual effort in effort to extract features automatically.
4. Applying, comparing and evaluating different multiple embeddings/feature extractors (CNNs, wavelets, polynomials) for the optimal performance of transformers for time series classification.
5. Applying, identifying and highlighting the short comings of applying a standard process model for applying data science exploratory project to industrial production data set.
6. Proposed and demonstrated the effect of different feature extraction embeddings for transformers for time series data for the task of classification.

1.4 Publications

The elaboration and the results obtained during this thesis have been published in several journals and scientific conferences contributing to the content of this dissertation.

The list of publications is presented below:

Peer reviewed Journal Publications

- F. S. Butt, L. La Blunda, M. F. Wagner, J. Schäfer, I. Medina-Bulo, and D. Gómez-Ullate. Fall Detection from Electrocardiogram (ECG) Signals and Classification by Deep Transfer Learning. *Information*, 12(2):63, 2021. doi:10.3390/info12020063 [34]
- F. S. Butt, M. F. Wagner, J. Schäfer, and D. G. Ullate. Toward automated feature extraction for deep learning classification of electrocardiogram signals. *IEEE Access*, 10:118601–118616, 2022. doi:10.1109/ACCESS.2022.3220670 [37]
- J. Rosa-Bilbao, F. S. Butt, D. Merkl, M. F. Wagner, J. Schäfer, and J. Boubeta-Puig. In *IoT-based Indoor Air Quality Management System for Intelligent Education Environments*, 2024. Submitted [199]

Scientific Conferences and Workshops

- F. S. Butt, J. Schäfer, M. F. Wagner, and D. G.-U. Oteiza. Time series analysis using machine learning techniques: Medical and industrial applications. In *proceedings of II Jornadas de Investigación Predoctoral en Ingeniería Informática (JIPII 2022)*, Cádiz (Spain), 2022. Department of Computer Science and Engineering, UCA. [35]
- F. S. Butt, J. Schäfer, M. F. Wagner, and D. G.-U. Oteiza. Towards Automated Feature Extraction For Deep Learning Classification of Electrocardiogram Signals. In *8th Spanish-German Symposium on Applied Computer Science (SGSOACS 2022)*, Toledo (Spain), 2022 [36]
- F. S. Butt, J. Schäfer, M. F. Wagner, and D. G.-U. Oteiza. Explainable AI for time series classification - An Overview and future directions. In *9th Spanish-German Symposium on Applied Computer Science (SGSOACS 2023)*, Tutzing (Germany), 2023 [38]
- F. S. Butt, J. Schäfer, M. F. Wagner, and D. G.-U. Oteiza. Automatic Feature extraction for time series analysis. In *Workshop: Statistics, Machine Learning and Applications*, Kaub (Germany), 2023 [39]
- F. S. Butt, J. Schäfer, M. F. Wagner, D. Stegelmeyer, and D. G.-U. Oteiza. Application of crisp-dm and dmme to a case study of condition monitoring of lens coating machines. In *Proceedings of the 2023 IEEE International Workshop on Metrology for Industry 4.0 & IoT (MetroInd4.0&IoT)*, Brescia (Italy), 2023. IEEE. doi:10.33965/ac2019_201912c027 [40]
- A. M. Binder de Serdio, D. Stegelmeyer, and F. S. Butt. Early Indicators of Project Abandonment in Industry-Academia Collaborations: Developing an Assessment Framework for Industrial Data Science Projects. In *10th Spanish-German Symposium on Applied Computer Science (SGSOACS 2023)*, Cádiz (Spain), 2024 [22]
- F. S. Butt, M. F. Wagner, J. Schäfer, and D. G.-U. Oteiza. In *Adopting Dimensional Embedding For Time Series Classification In Transformer Architecture*, 2024. Submitted [41]

1.5 Structure of the PhD Thesis

In this chapter the structure of the thesis is presented containing an overview of the sections which include a description of the content.

- Chapter 1 presents an overview of the problem in time series analysis using DL techniques. This chapter also highlights the research questions focused and objectives achieved during this thesis. The publications published during the course of PhD. have been categorized as Journal publications and scientific conferences and workshops publications and mentioned to highlight the attainment of the requirements of this doctoral program.

- Chapter 2 establishes the ground work by formally defining the terms such as time series and classification from a statistical point of view. It also presents an overview of the time series analysis problem.

- Chapter 3 emphasises on the application of TSA for an Industry 4.0 use-case. This chapter describes an industrial collaboration and steps taken to adapt time series analysis and data science for industrial setup. CRISP-DM and DMME were applied as the process model to acquire knowledge form the data in a systematic way. Research question **RQ2** and **RQ3** were discussed in detail as an effort towards finding the gaps in CRISP-DM and DMME was made. Not only were the weaknesses of these process models discussed but also recommendations have been made to improve for the better adaptation of data science in industrial framework. This chapter also lays out the foundation of time series analysis for other domains during the course of the thesis.

- Chapter 4 presents a comprehensive state of the art for the time series analysis at its various stages including classification and feature extraction specifically for the most recurrently occurring domain in this dissertation, i.e. electrocardiograms (ECGs). It gives an overview on the working of the human heart to elaborate the construct of the ECG signals and the focuses on the different patterns of ECG signals and their normal ranges. It also highlights in detail the existing work and different approaches used for feature extraction for deep learning and TSC.

- Chapter 5 is based on the study to verify the **RQ5**, that fall can be differentiated and detected from other human activities using ECG signals. The chapter foregrounds the steps to pre-process the raw ECG signals obtained like filtering, resampling, normalizing, and augmenting. Than the experimental details for the modeling and explaining the deep learning models are layed out. It describes the details of the transfer learning to two of the most prominent pre-trained CNNs available, AlexNet and GoogLeNet.

- Chapter 6 This chapter is built on the idea of extracting features from the time series specifically ECG's by using different approaches. To this end, two algorithms are proposed. Both consist of well-known mathematical operations of convolution

and wavelet transformation placed in front of LSTMs and transformer respectively. This chapter explains the experimental setup, the data sets used and the results obtained by the proposed algorithms. The results are statistically verified to be significant and are computationally and numerically better than state of the art results for majority of the datasets. This chapter addresses the research questions **RQ4** and **RQ6**.

- Chapter 7 This chapter is detailed on feature extraction mechanisms for deep and machine learning in general and for biomedical in particular. This includes explanation of wavelet basics and then this chapter also looks into details mathematically as in to why wavelets can act as better feature extractors with transformers. This chapter lays the foundation of more experiments in the coming Chapter 8 and to the research questions **RQ7**.

- Chapter 8 takes forward the experiments from Chapter 6 and investigates the role of different embeddings for TSC in Transformers. For this purpose, a new architecture of transformer is proposed to adapt better to time series classification. **RQ6**, **RQ7** and **RQ8** are answered in detail in this chapter. Additionally this chapter provides a review on latest literature on transformers for time series and how the dimensional embeddings are adapted across different studies. This also includes experiments and results from different wavelet transformations and feature maps as dimensional embeddings

- In Chapter 9 iterates through each research question and discusses the findings respective of each RQ.

- In Chapter 10 examines its accomplishments, and also outlines future research plans that are expected to enhance the existing feature extraction of time series solution.

Chapter 2

Time Series Analysis

> 'Although this may seem a paradox, all exact science is dominated by the idea of approximation.'– (Bertrand Russell)

This chapter describes the basic foundations of defining time series and time series analysis from statistical point of view. In this chapter, the formal definitions and a little background for the terms used frequently used in this thesis are given.

2.1 Probability Spaces and Time Series

The following definitions have mainly been taken from Fuller [80].

While investigating observing an experiment or a natural phenomenon, it is important to have a representation of all the possible outcomes.

The *elementary* events are referred to as individual outcomes denoted by ω. The set of all possible *elementary* events is called *sure* event represented by Ω. For example, in case of a dice, $\Omega = 1, 2, 3, 4, 5, 6$. Let A be a subset of Ω and let $\mathcal{F}$ be a collection of such subsets. If we observe the outcome ω and ω is in A, then A is said to have occurred. Similarly, *P(A)* is intuitively specified as the probability that A will occur. The function *P(A)* is required to satisfy following:

- **AXIOM 1**.*P(A)* ≥ 0 for every A in $\mathcal{F}$.
- **AXIOM 2**. $P(\Omega) = 1$.
- **AXIOM 3**. If A_1, A_2.... is a countable sequence from $\mathcal{F}$ and $A_i \bigcap A_j$ is the null set for all $i \neq j$, then $P(\bigcup_{i=1}^{\infty} A_i) = \sum_{i=1}^{\infty} P(A_i)$.

For technical mathematical reasons, defining $P(A)$ for all A in $\mathcal{P}(\Omega)$ and also to satisfy Axiom 3 is always not possible. To address this, the class of subset $\mathcal{F}$ of Ω on which P is defined is required to:

1. If A is in α, then the complement A^c is also in $\mathcal{F}$.
2. If $A_1, A_2, ...$ is a countable sequence from α, then $\bigcup_{i=1}^{\infty} A_i$ is in α.
3. The null set in is $\mathcal{F}$.

So, a non-empty collection $\mathcal{F}$ of subsets of Ω that satisfies conditions 1 to 3, is said to be a *sigma-algebra* or *sigma-field*.

A **probability space**, is represented by $(\Omega, \mathcal{F}, P)$, is the sure event Ω together with a sigma-algebra $\mathcal{F}$ of subsets of Ω and a function $P(A)$ defined on $\mathcal{F}$ that satisfies Axioms 1 to 3. For the practical implications, it is deemed enough to record the outcome of an experiment by some function that assumes values on the real line. This is done by assigning each outcome ω a real number $X(\omega)$ and, if ω is observed, we record $X(\omega)$.

Formally, a random variable X is a real valued function defined on Ω such that the set $\omega : X(\omega) \leq x$ is a member of $\mathcal{F}$ for every real number x. The function $F_x(x) = P(\omega : X(\omega) \leq x)$ is called the distribution function of the random variable X.

2.2 Time Series Analysis

Let $(\Omega, \mathcal{F}, P)$ be a probability space and let T be an index set. A stochastic process is a collection of time indexed random variable $X(\omega, t)$, where ω belongs to a sample space and t belongs to an index set. $X(\omega, t)$ is a random variable for a fixed t. For a given ω, $X(\omega, t)$, as a function of t, is called a sample function or realization. In time series analysis and stochastic processes, the population that consists of all possible realizations is called the ensemble. Some concepts and terminologies are explained below for a better appreciation of time series analysis. The index set is assumed to be the set of all integers unless mentioned otherwise. Consider a finite set of random variables $\{X_{t1}, X_{t2}, ..., X_{tn}\}$ from a stochastic process $X(\omega, t) : t = 0, \pm 1, \pm 2, ...$ The n-dimensional distribution function is defined by:

$$F_{X_{t1},...,X_{tn}}(x_1, ..., x_n) = P\{\omega : X(t_1, \omega) \leq x_{t_1}, ..., X(t_n, \omega) \leq x_{t_n}\} \tag{2.1}$$

where x_i, $i = 1, ..n$ are any real numbers. A process is a first-order stationary in distribution if its one dimensional distribution function is time invariant, i.e., if $F_{X_{t1}}(x_1) =$

$F_{X_{t1+k}}(x_1)$ for any integers t_1, k, and $t_1 + k$; second order stationary in distribution if $F_{X_{t1},X_{t2}}(x_1, x_2) = F_{X_{t1+k},X_{t2+k}}(x_1, x_2)$ for any integers $t_1, t_2, k, t_1 + k$; and nth-order stationary in distribution if

$$F_{X_{t1},...,X_{tn}}(x_1, ...x_n) = F_{X_{t1+k},....,X_{tn+k}}(x_1, ..., x_n) \tag{2.2}$$

for any n-tuple $(t_1, ...t_n)$ and k of integers. If equation 2.2 holds for any n,i.e., $n = 1, 2...$, the process is said to be strictly stationary [257]. After establishing that a stochastic process, $X(\omega, t)$, is a set of time indexed random variables defined on a sample space, the variable ω is suppressed and write $X(\omega, t)$ as $X(t)$ or X_t. For a given real-valued process $X_t : t = 0, \pm1, \pm2, ...$, we define the mean function of the process

$$\mu_t = E(X_t) \tag{2.3}$$

the variance function of the process

$$\sigma_t^2 = E(X_t - \mu_t)^2, \tag{2.4}$$

similarly the covariance function between X_{t_1} and X_{t_2}

$$\gamma(t_1, t_2) = E(X_{t_1} - \mu_{t_1})(X_{t_2} - \mu_{t_2}) \tag{2.5}$$

and the correlation function between X_{t_1} and X_{t_2}

$$\rho(t_1, t_2) = \frac{\gamma(t_1, t_2)}{\sqrt{\sigma_{t_1}^2}\sqrt{\sigma_{t_2}^2}} \tag{2.6}$$

2.3 Time Series Classification

TSA is an umbrella term used to describe many tasks including classification, segmentation, predicting, anomaly detection, motif discovery etc. According to [74] and [111], time series classification is one of the most vital and challenging task in time series analysis. This section formally defines TSC as it is also the main focus of this thesis.

Categorization of data into identifiable classes is called pattern classification. The term pattern refers to the observable information that is in form of quantitative description of the data of interest. Pattern classification problem can be reduced to a classification problem in most cases. Classification, in turn, can be both supervised and unsupervised. In statistical literature, the supervised learning is mostly referred to as discrimination, which would imply that classification rule is established from the given correctly classified data. In this thesis, since we deal with time series data, whose under laying pattern

generation is majorly a statistical process, the classification problem is often defined within the framework of statistical decision theory [192].

The classification or pattern recognition in humans is a closed process and we do not know the details about how precisely a decision is made. In contrast to this, the decision taken by machine pattern recognition algorithms follow a series of mapping from a high dimensional pattern space to a smaller decision space as shown in Fig. 2.1.

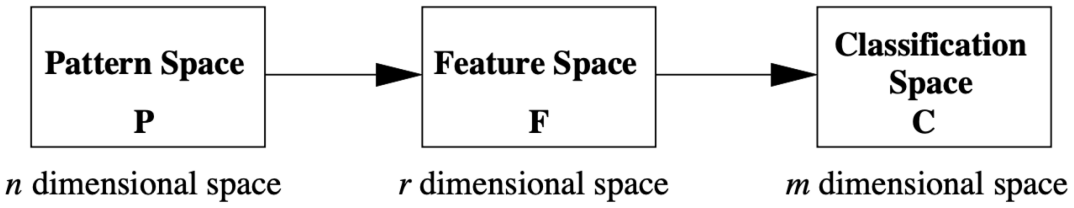


FIGURE 2.1: Standard Model for Statistical Pattern Classification [192]

While the intricacies of deep learning remain incompletely understood, the fundamental concept revolves around explicit mappings. These mappings require a recognition paradigm expressed in a rule-based form that is both explainable and implementable by machines, despite the overarching mystery surrounding the nature of deep learning. The pattern recognition mapping can be considered to be formed by two stages as shown in the Fig. 2.1: A mapping that maps the observable pattern space to a feature space. This stage is known as feature extraction. Another mapping from the feature space to the classification space which decides the class membership of the data point. One of the characteristic of feature extraction is isolating the useful attributes of the pattern. The feature space should reflect properties that enhance the in class similarities and between-class dissimilarities.

In the next step, the feature space is attempted to be separated into regions pertaining to each class (usually $r > m$). But these divisions between feature space and classification space are more practical than theoretical. A good feature extractor would make the task of a classifier trivial. Similarly, a good classifier would bypass the need for a feature extractor[192]. Some of the major concerns regarding the would-be classifiers are accuracy, speed, time to learn and interpretibility according to [166].

Many statistical approaches exist for the pattern recognition. Some of the classic classification approaches include Fisher's linear discriminant, decision tree and rule based methods, k- nearest neighbour and density estimates etc. Efron in [72] refers the collection of algorithms such as random forests, gradient boosting, support vector machines, neural nets (including deep learning) as "pure prediction algorithm". A prediction algorithm is a generic algorithms for inputting a dataset $\mathbf{d} = (x_i, y_i), i = 1, 2, ..., n$ and

outputting $f(x, \mathbf{d})$ that would for any predictor vector x, yields a prediction

$$\hat{y} = f(x, \mathbf{d}) \tag{2.7}$$

The true error rate of the rule for classification where $\hat{y}_i \neq y_i$, is given by

$$Err = Ef(X, \mathbf{d}) \neq Y \tag{2.8}$$

Where (X,Y) is a random draw from the same probability distribution gave the (x_i, y_i) pairs in $\mathbf{d}$.

2.4 Bayes Decision Theory

Bayes decision theory is one of the most prominent decision making theory in statistics. It is based on the assumption that the decision problem is fully expressed in probabilistic terms and that all relevant probability terms are known. This is not always true in real applications. The problem is described as followed: Given an n-dimensional measurement vector $\mathbf{x} = [x_1 x_2 .... x_n]^T$, determine it class membership or state of nature. There a M potential classes labelled by the set $\Omega = \{\mathcal{M}_1, \mathcal{M}_2, ..., \mathcal{M}_M\}$. Let the set $A = \{\alpha_1, \alpha_2, ..., \alpha_M, ... \alpha_r\}$ represents all possible actions; we allow $r \geq M$ since in certain situations it may be advantageous to enable the system to take a course of action that does not result in a forced decision on class membership. For example, a soft decision or 'doubt' option if more than one of the classes seems plausible and an 'outlier' option if none of the classes is acceptable. The action α_i, $\forall\, i = \{1, .., M\}$, corresponds to the decision that $\mathbf{x}$ belongs to $\mathcal{M}_i$.

From Bayes rule the probability of assigning $\mathbf{x}$ to class $\mathcal{M}_i$.

$$P(\mathcal{M}_i|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{M}_i)P(\mathcal{M}_i)}{p(\mathbf{x})} \tag{2.9}$$

where $P(\mathcal{M}_i)$ is the a priori probability that the class is $\mathcal{M}_i$; and $p(\mathbf{x}|\mathcal{M}_i)$ is the conditional probability density function (PDF) of $\mathbf{x}$ given that it belonged to class $\mathcal{M}_i$.

If $\lambda(\alpha_i|\mathcal{M}_j)$ is the cost or loss incurred by taking action α_i when the class is actually $\mathcal{M}_j$ then the expected loss or conditional risk associated with α_i is given by

$$R(\alpha_i|\mathbf{x}) = \sum_{j=1}^{M} \lambda(\alpha_i|\mathcal{M}_j)P(\mathcal{M}_j|\mathbf{x}) \tag{2.10}$$

The optimal Bayes decision process tries to select the action which eventually minimizes the conditional risk. In case of classification, a natural decision rule would minimize the average probability of error. To this end, the suitable cost function is the zero-one cost function,

$$\lambda(\alpha_i|\mathcal{M}_j) = \begin{cases} 0, & i = j \\ 1, & i \neq j \end{cases} \forall i, j = 1, ..., M \tag{2.11}$$

This assumes that a class membership decision is achieved through all courses of action and that the correctly classified vectors lead to a cost of zero and for all incorrect decisions is unity. The Bayes risk, or the expected loss under this decision rule, corresponds to the classification with the minimum error rate and represents the optimal performance achievable by a classifier.

Chapter 3

Time Series Analysis - An Industrial Application

> 'Machine intelligence is the last invention that humanity will ever need to make.'– (Nick Bostrom)

This chapter is based primarily on an industrial application of the time series analysis on usecases which were obtained as a part of industrial collaboration between my research group INDAS and industrial partners. The content below is mainly derived from the publication [40].

3.1 TSA for Industry 4.0

Digitalization is the transcendent future. The Covid-19 crisis has only emphasized the need to digitalize the world as quickly as possible. From classrooms in schools to the operation theatres in hospitals, automation has proven to be a necessity rather than luxury.

Decisions regarding essential industrial processes such as scheduling, maintenance management and quality improvement etc. are being influenced by the availability of data and its usage [235]. The definition by the Industry 4.0 Working Group initiated by the Federal Ministry of Education and Research (BMBF), Germany is as follows:

"Networks of manufacturing resources (manufacturing machinery, robots, conveyor and warehousing systems and production facilities) that are autonomous, capable of controlling themselves in response to different situations, self-configuring, knowledge-based,

sensor-equipped and spatially dispersed and that also incorporate the relevant planning and management systems".

"Industry 4.0 is a national strategic initiative from the German government through the Ministry of Education and Research (BMBF) and the Ministry for Economic Affairs and Energy (BMWI). It aims to drive digital manufacturing forward by increasing digitization and the interconnection of products, value chains and business models. It also aims to support research, the networking of industry partners and standardization" [53].

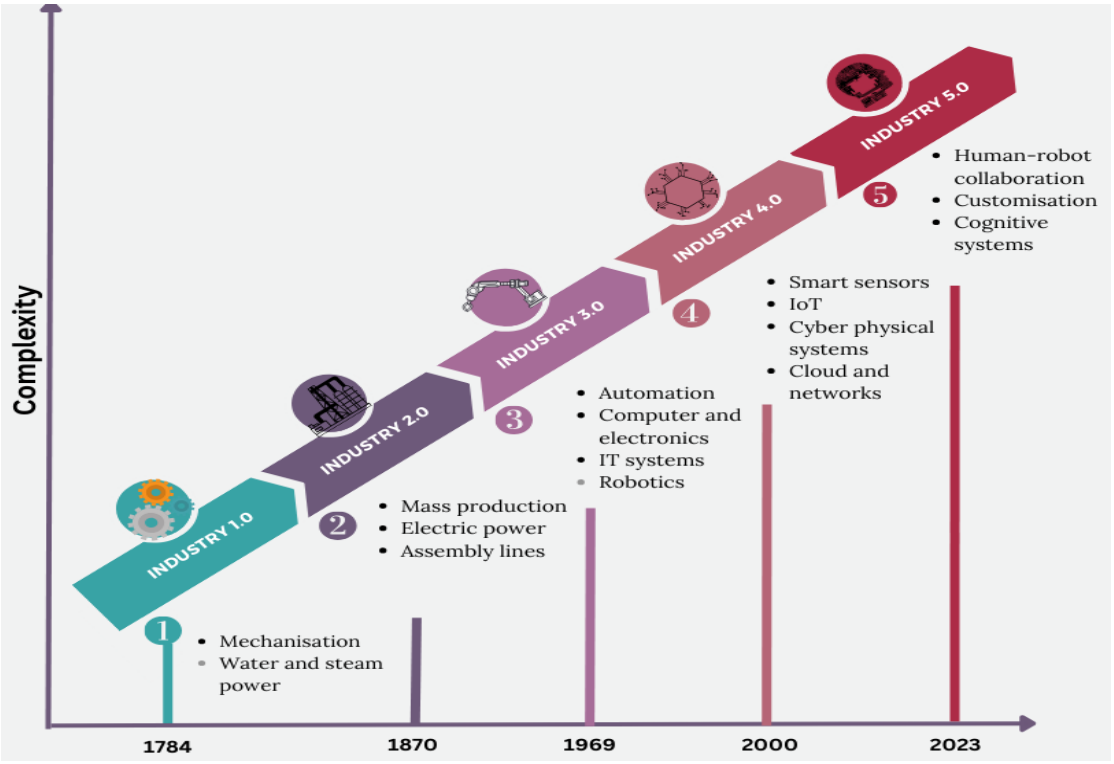


FIGURE 3.1: Schematic illustration of the four industrial revolutions and projection into Industry 5.0.

The 4.0 in Industry 4.0 refers to the 4th industrial revolution that mankind has ever recorded. The first revolution happened around 10,000 years ago by domesticating the animals. It combined human effort with that of animals to produce, transport and communicate (See fig.3.1). The second revolution called industrial revolution happened in the second half of the 18th century shifting from manual work to mechanical work through machines enabling mass production. The third revolution also known as digital revolution began in 1960s. The fourth industrial revolution actually started with the digital age and has come of age in the present world of mobile devices, powerful and cheaper sensors along with artificial intelligence (AI). Industry 4.0 was a term coined at Hannover Fair, Germany in 2011 describing the concept of 'smart factories' through revolutionizing the global value chains [213]. Maintenance is a vital industrial operation.

Every machine has a finite usability and lifetime. To prolong the lifetime, the machine needs to be maintained periodically. Predictive maintenance (PdM) helps us estimate when machine failure will occur so that we can maintain in advance, reduce downtime and maximize equipment lifetime. The maintenance's are of three main types:

- Reactive maintenance where we wait for the machine to go down or fail and then perform the repairs on it.
- Preventive maintenance where we check the performance periodically and repair it if required. This leads to increased down time hence resulting decreased productivity.
- Predictive maintenance which lets us estimate the time-to-failure of a machine. This helps us to reduce the downtime by scheduling the repair only when a time-to-failure is approaching. Since for industry 4.0 this decision has to be made using sensor data, we can use the data to even pin point the exact issue and that can help us improve the performance of the machine as well.

One of the objectives of this thesis was to apply time series analysis to the vital field of Industry 4.0 production using appropriate machine learning techniques for the purpose of predictive maintenance. Working on the predictive maintenance is also important to understand certain behaviors of machine and helps to outline the correlation of use cases with certain parameters which cannot be otherwise seen. This objective has helped to focus primarily on the maintenance phase of the Industry 4.0 by shifting it from preventive or reactive maintenance to predictive maintenance. Involving AI in different facets of industrial processes and products is vital for the growth of Industry 4.0. [14] provides a comprehensive overview on the current potential of artificial intelligence application in the manufacturing industry.

The successful deployment of AI in industrial enterprise is still absent in large despite the high presumptions of its usability. According to Accenture, 87% of manufacturers have not yet implemented an Industry 4.0 approach [95]. It creates a roadblock for the Industry 4.0 vision for data-driven digital transformation in practise. This absence is mainly contributed to the lack of many successful applications of AI in Industry 4.0. According to [183], most of the research is done in laboratory settings instead of successful deployment at later stages. Deploying a new automated system in an already successful traditional running system can disrupt the production supply chain and might not be so interesting for the conservative industries. Another reason mentioned in [183] is lack of documentation of the failed use-cases which could highlight the limitations of certain approaches for other researchers. All the case studies reviewed by [183] had common

limitations of data availability, quality and related issues (e.g. scarcity, contamination, drift).

Condition monitoring industrial machinery by using data mining in industrial projects has been extensively extended in the last few years. Applying data science to industrial processes should be straightforward in theory, but very few instances in the literature deal with the actual practical issues encountered while carrying out industrial data science projects. The case study discussed in this chapter was pursued in accordance with the steps outlined in the standard CRISP-DM (CRoss-Industry Standard Process for Data Mining) including its latest holistic approach for engineering applications called DMME (Data Mining Methodology for Engineering Applications). The industrial data was acquired as part of industrial cooperation from multiple anti-reflective lens coating machines. Various deep learning (DL) models like long short-term memory (LSTM), and machine learning (ML) models like Decision Trees and Support Vector Machines (SVM) were used as proof of concept for confirming the domain understanding of the process experts. Our main contribution is the description of deficiencies and gaps of the standard process framework CRISP-DM based on the issues faced in implementing each phase of the process in a real world case study. In addition, we propose future research ideas to close these gaps. This complements findings in the literature on gaps in CRISP-DM and DMME.

3.2 Background and Motivation

The application of data sciences in an industrial setup has expanded exponentially. The field has vast new advancements which are not captured in the process models for traditional data mining. Data mining now includes the latest data pre-processing techniques as well as machine and deep learning techniques. With the expansion of Industry 4.0 in the last decade, the amount of available sensor data has increased tremendously. Many industries are joining the Industry 4.0 family by rapidly transforming and incorporating digitization to improve the decision making process, existing processes and product development etc . Data science has helped Industry 4.0 navigate its processes and services in a meaningful manner from conception to its goal in terms of business model and technological advantage over its competitors. Applying data science for knowledge discovery from Industry 4.0 in a systematic manner is not explored in detail in the literature [44]. The need for a standardized process model for knowledge discovery in Industry 4.0 is evident. Though data science is applied to industrial problems very often in current times, no systematic approach is in sight. Although the need for AI assisted industrial operations has increased rapidly over the last decade, according to [110], 75 to 85 percent

of practical ML projects currently do not match their sponsors' expectations. The major reasons for the unmet expectations as listed are unrealistic expectations, high accuracy threshold for adoption, lack of focus on business goals, and lack of quality data. But there is not enough literature available to highlight the practical implementation challenges for these process models in real industrial set-ups.

Applying data science for time series in an industrial set up in an agile manner is challenging and has to follow some structure. In the following work, we apply a standard CRISP-DM procedure along with its extension DMME to a case study of eyeglasses anti-reflective coating quality assessment project. While we describe the case study per se which yields many interesting insights carefully, the main focus of this study is the description of the many benefits of applying the standard process model for data mining but also the many gaps and shortcomings which are highlighted and discussed in this chapter in detail. These findings will serve as a basis for future research to improve the methodologies CRISP-DM and DMME to close these gaps.

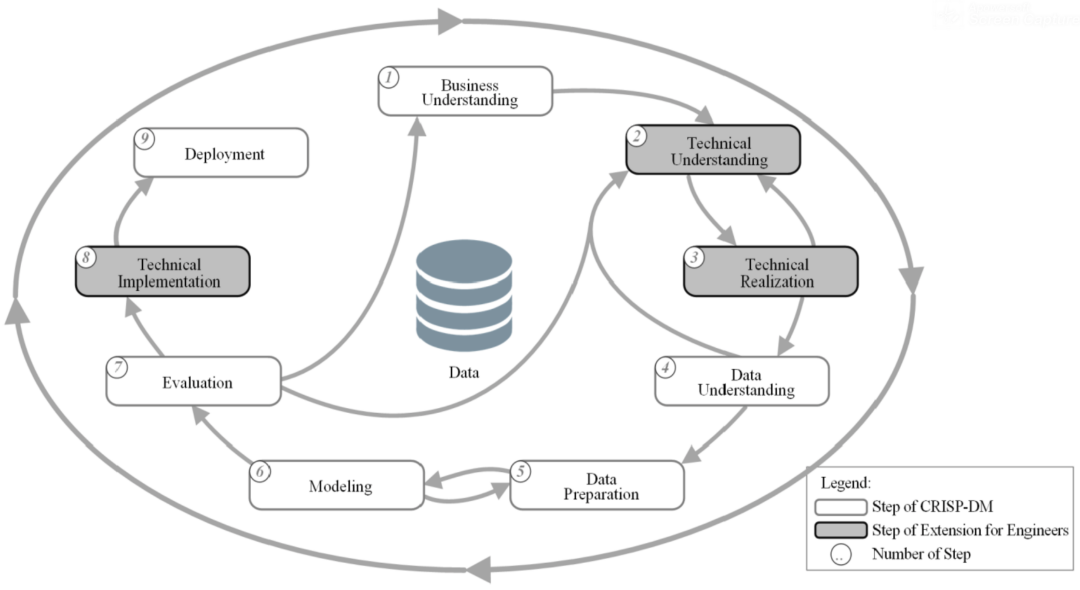


FIGURE 3.2: Graphical representation of the DMME process [108]

3.3 Related Work and Our Contribution

Some process models have been defined originally for the purpose of carrying out data mining on industrial data sets, such as KDD (Knowledge Discovery in Databases) and SEMMA (sample, explore, modify, model, and assess) developed by SAS institute in 2005[244]. [260] introduced a comprehensive process model for the application of data mining (DM) on Industry 4.0 called "CRoss Industry Standard Process for Data Mining"

or CRISP-DM. Since then, it has become a de facto standard for DM. This process model intuitively documents the commonly used steps in most of the data mining projects. Many projects involving industrial data mining still follow this process model such as [24],[187] and [222]. For a comprehensive survey on CRISP-DM usage in recent years, see [212]; for a recent survey on generic data mining, see [185]. However, according to [156] CRISP-DM is most suitable for projects which are goal oriented and process driven and might not fit projects with exploratory data science nature.

CRISP-DM serves as a base framework to layout the critical phases which occur in performing data mining related projects. It has been adapted for different specific tasks resulting in multiple variations e.g. CRISP-TDM for temporal data mining [45], CRISP-DM0 for null-hypothesis driven confirmatory data mining [98], CRISP-EM for evidence mining [250], and CRISP-MED-DM for data mining in healthcare [175]. Similarly, [233] uses CRISP-DM as a basis to develop an end-to-end process model for ML based applications and it covers the appropriate phases in the life-cycle of a ML application development. In 2019, another extension to the classic CRISP-DM model named DMME (Data mining methodology for engineering applications) was presented in [108] see Fig. 3.2. One of the draw backs of CRISP-DM highlighted in this paper is that it does not specify a data acquisition phase with production scenarios. The CRISP-DM model has six sequential phases. DMME has been specifically tailored for the engineering applications. It also includes a dedicated step for data acquisition.

In this study, DMME is applied to evaluate the sensor data from an anti-reflective (AR) coating machine for lenses keeping the original CRISP-DM model as the base process model. We aim to predict the quality of the coating based on the machine parameters. We have highlighted the principal real-life challenges in the application of the CRISP-DM and DMME model at every step of the process. Another goal was to mminimize the downtime of the production line through appropriate data analytics by predicting production and equipment failure beforehand.

The major contributions of this study are the application of CRISP-DM and DMME to a real industrial process and to highlight real-life challenges at each process step's implementation in a real industrial setup identifying deficiencies and gaps of CRISP-DM and DMME.

3.4 Process Models: CRISP-DM and DMME

The main difference between CRISP-DM and DMME is the introduction of two additional steps called technical understanding and conceptualization and technical realization and testing for engineering applications. We used technical understanding in addition to our traditional CRISP-DM process as shown in 3.3. Our collaboration was interdisciplinary and focused on an engineered approach to process maintenance and data handling.

3.4.1 Business Understanding

The interdisciplinary collaboration in this study consisted of three partners, who all had different business cases:

- Bühler Leybold Optics/Partner 1: The designer and manufacturer of the glass coating machines as shown in the fig 3.4.
- Optovision Moderne Brillenglastechnik GmbH/Partner 2: A customer of Bühler using the coating machine for the anti reflective coat to its eyeglasses. This partner would be the main provider of the data.
- INDAS, Frankfurt University of Applied Sciences/Partner 3: The research group applies data mining on the data provided by Partner 2.

The main aims of this step was to set the collaboration goals which would align with the Industry 4.0 vision for the all the partners. For Partner 1, the goal was to achieve more insight into the effect of each parameter on the final quality so that the basic structure of the machines can be improved using sensor data[1].

Partner 2 had two business cases, indeed. The first was to increase the productivity and quality of the processes by reducing the number of failed batches due to coating and the second one was to predict the next batch and/or optimizing the process parameters.

For the Partner 3, the goal was to have good quality data which can be used for appropriate machine learning and data mining. On the other hand, the industrial partners would generally have the vision to either increase their production directly or indirectly or to understand their processes better. One of the ways forward is to make sure at this phase that the business case defined is an actual and important use case for the industrial

[1]The lack of quantitative definition of this business case became apparent only during the course of the project, see the discussion below.

partners. Otherwise, the project in the long run would suffer from lack of managerial backup and interest.

As described in [86], an important goal for engineers in the data mining processes are to extract relevant parameters that influence the quality of the target predictors. In most glass coating industries the quality evaluation process is still manual as it seems to be cost and time effective. This presented a challenge in later phases because of absence of the objective quality criteria. The data scientists have to be involved at this point to evaluate the prediction criteria for the explained use case and make sure that all the data obtained are meaningful and quantitative. One of the main issues faced at this stage is that the industrial partners would prefer to start the data mining process without objectively defining the use case for the data scientist. This may be caused by lack of experience and expertise for implementing an industry 4.0 project and must be avoided at any cost.

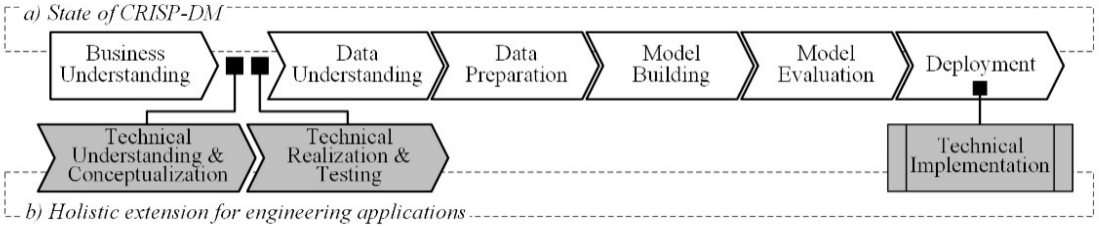


FIGURE 3.3: Phases of the reference model of DMME (data mining methodology for engineering applications). (a) State of CRISP-DM, (b) holistic extension for engineering application.[108]

3.4.2 Technical Understanding and Conceptualization

This DMME phase develops the understanding and concrete objectives for data mining and machine learning experiments.

3.4.2.1 Determine Technical Objectives

Understanding the technical tasks and determining the use cases to be solved. The technical task was to improve the coating process by determining the parameters affecting the coated quality of the lenses. Also, the specific use case was to increase the number of processes run on the machines before the downtime for cleaning the machine.

The technical goals were stated for all the parties separately. For Partner 1, the technical goal was to learn about cause-and-effect relationship of important parameters through the data in the coating machine to the quality of the lenses. This is aimed to improve the design of the machine in future.

For Partner 2, since they use the machines in their processes, the main goal was to increase their productivity by increasing the number of processes before stopping the machine to clean. An existing domain related assumption was that after every 24 processes the machine needs to be cleaned. That means machines had a downtime of some hours after every 24 processes. It is important to note, that while extending the cleansing interval to save costs is part of the use case of Partner 1, it was treated as a technical "given" constraint rather by some operational staff. Therefore, we conclude that questioning hidden assumptions like this one should be a paramount step in any data mining methodology.

FIGURE 3.4: A Coater Machine by Bühler Leybold Optics for Air Coating Lenses. (©All rights reserved by Bühler Leybold Optics)

3.4.2.2 Analysis of the Technical Situation

This activity analyzes the initial situation as part of the preparation of the development work. To this end, Partner 1 and 2 explained in detail the structure of the system and how the process is carried out. To gather more knowledge about the problem and process understanding, an onsite visit was paid by partner 3 to partner 1 and 2. Also, in order to have a better continuous support setup for online meeting twice a week was setup between the data scientist and the process engineers. One of the main challenges at this phase was to determine the relevant physical and technological interactions and their influence on the target variable as it was a part of the problem description. At this phase the

experts involved were the process engineers, manufacturing process planner, and quality assurance personnel to contribute in the understanding of the process and evaluating the current situation. Another real time issue determined at this phase was that the target variable, i.e., the quality of the reflex is determined manually by a quality assurance manager. Applying objective quality measurement solutions such as spectrometer is not cost effective in a mass eye lens producing industry.

3.4.2.3 Conceptualization

This activity mainly focuses on the development of the solution approach as a proof of concept. Hence it aims to determine the physical effects and their mapping to the technical objectives. This is achieved through a constant follow up between process engineers and data scientists.

As the first step, the principal effects that influence the quality are determined from domain knowledge. There are many kinds of air coating done inside the coater machine. Each one called a recipe is a different combination of substances and results in a different coating. The coating is achieved by heating and evaporating the substances which then attach themselves to the surface of the lenses.

The process of coating leaves residue inside the coater machine; and with every subsequent process the residue increases. With residue inside, the humidity is trapped inside the coater machines which in turn makes the initial pump down phase, where a vacuum is created, difficult or more time-consuming to achieve. The longer the machine takes to achieve vacuum, the lower the assumed probability of producing good quality.

3.4.2.4 Acquisition Concept

In the case of this project, the related sensor data was already available at Partner 2 site. The data set consists of a batch file that contains all the information about the process files. Each process file contains time series for multiple parameters over time such as temperature, pressure, ion source rate, etc.

A data mining approach was also developed as part of this activity. The DM idea was to start with correlation analysis to look for any relationships between the parameters and target parameters.

3.4.2.5 Experimental Planning

This step focuses on the experimental setup for the data acquisition and knowledge generation. In our case, knowledge generation was generated through regular meetings with the process engineers of both Partner 1 and 2 every week. This created a feedback loop where domain related knowledge misunderstanding or missing in data, could be clarified in a short span of time.

The data was from the process from all the machines manufactured by Partner 1 and deployed at Partner 2.

3.4.2.6 Specification and Project Plan

The project plan and specifications had been decided at a managerial level of the project. Partner 1 assumed the project management, a steering committee met regularly to oversee the project and a biweekly project meeting was decided. In the course of the project this was changed to a scrum approach to speed up the transfer of knowledge between industry and data scientists.

3.4.3 Technical Realization and Testing

In this phase the prototype for the realization of experimental plans along with proof of concept is achieved.

3.4.3.1 Prototype Realization

As the test bed for proof of concept, a linear correlation analysis was done on batch files to find any related correlation between lenses marked for quality check due to reflex issues and the number of processes ran on the corresponding machine. From the domain understanding the more the number of processes, the worse the quality of the lenses.

3.4.3.2 Test of the Concept

After the initial correlation analysis, no substantial correlation was found between the said parameters as shown in Fig. 3.5. This was not in conformance of the domain understanding. The explanation to this by the process engineers and data scientist was that the machines are still not carrying out the optimal number of processes before the downtime for cleaning. So we can increase the number of processes before cleaning.

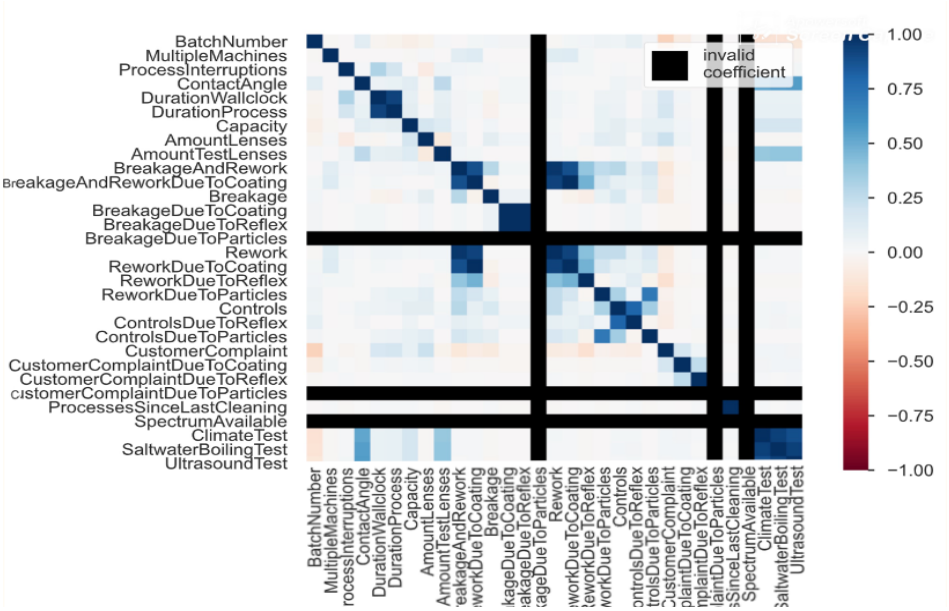


FIGURE 3.5: Correlation matrix displaying the correlation between important different quality parameters

This was already interesting for partner 2 but partner 1 needed more insight into the individual or combined effect of each parameter to the final quality of coating on the lenses. For understanding the parameters and their role in the quality determination, another set of experiments were designed. This included applying deep learning models for time series such as LSTM's to the process files.

3.4.3.3 Perform Experiments and Collect Data

From the results from previous step, the number of processes before cleaning were increased by two processes at a time until they reached 36. The caution in increasing the numbers was that a sudden decrease in quality of lenses at any machine must not be observed. The goal is finally to double the processes numbers before cleaning.

3.4.3.4 Documentation

The initial conclusion from the increase in processes was that in principle the number of processes could be increased without any noticeable affect on the end quality.

3.4.4 Data Understanding

The data understanding was achieved in a continuous process though visualization and domain experts input. This understanding also helped to further improve the data quality by eliminating files or parameters which did not align with the domain understanding.

The data was described initially through a data dictionary where each parameter was described with its purpose along with the units.

The data set was explored through visualization as shown in Fig. 3.6.

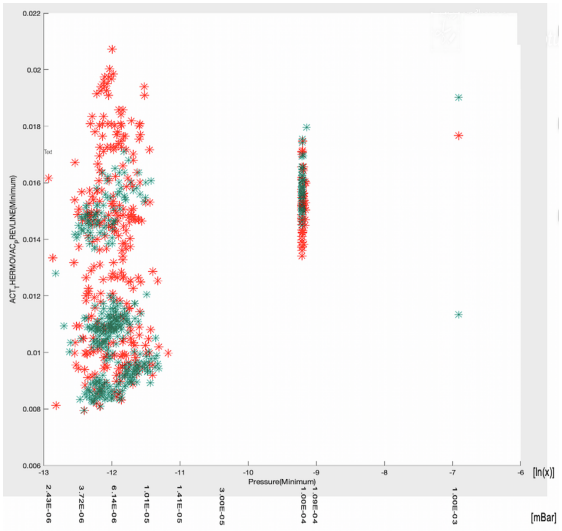


FIGURE 3.6: Diagram representing the distribution of (aggregated) pressure between good and bad quality processes. The x-axis represents (aggregated) pressure in milliBar (mbar) and the y-axis represents chamber pressure i.e. pre-vacuum line (mbar). Green stars represent good quality and red bad quality batches. Where 0: $<$ 10 % bad lenses per batch(in green) , 1: $>$70% bad lenses per batch(red)

3.4.5 Data Preparation

This step of the original CRISP-DM aims to prepare the data for related data mining and machine learning processes. However according to [69], the main focus of the methods like Crisp-DM and KDD is only focus on the vital tasks at every phase without going into details of actual adapted methods. [69] presents guidelines which are based on the DMME and they enable machines to acquire data that is suitable for data mining. In the first step for the data preparation all files with no data were eliminated. The next step was to eliminate files which were corrupt and/or when the process was interrupted in between. So only complete processes were taken into account. A total of 200 parameters were recorded in each process file. For initial elimination, Pandas profiler was used to create a statistical overview of the a parameter file. The parameters with low or no

variance were filtered out and parameters with substantial variance were removed. The

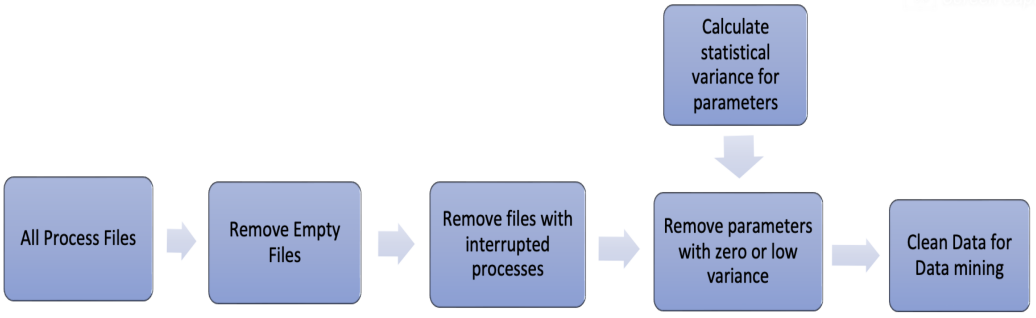


FIGURE 3.7: Flow diagram depicting the initial data cleaning process

initial data preparation process is shown in Fig. 3.7. After major filtration we worked mainly with maximum of 10 parameters. In order to gain a better understanding for Partner 1, experiments were designed to apply a deep learning and machine learning approach to predict the quality parameter from the parameters in the process file. For a better and stronger possibility of classification, the complete data set was divided into two classes: Class A, where quality of the lenses is less than or equal to 10%, and Class B, where quality is more than 70%. This threshold was chosen so that substantial difference in parameter ranges can be detected. Also, the complete time series for each process was taken initially. Since the time duration could vary from process to process slightly, all process files were padded with zeroes at the end to be of the same length. Each process had specific steps which were uniform across a specific coating recipe. In the second part of the experiments, the data was aggregated for each step. The aggregation functions applied were MIN, MAX, STANDARD DEVIATION and AVERAGE. But for this experiment, only similar recipe processes could be trained together.

3.4.6 Model Building

For the initial modeling, a simple LSTM was applied with rework due to reflex as a predictor from the parameters recorded during the processes. The pre-processed data set was used for the modeling. Similarly CNNs and other variations of LSTMs (BiLSTMs) were also implemented. The LSTMs could achieve accuracy beyond 55%.

3.4.7 Model Evaluation

The data was filtered progressively towards a more meaningful direction with application of domain knowledge. Data pre-processing techniques like data set balancing, filtration of processes using different machines and recipes were applied and respective models were trained with the filtered data. Different parameters were included to test the effect on the deep learning parameter prediction, but the maximum accuracy achieved was 65%, roughly the division between the classifying classes.

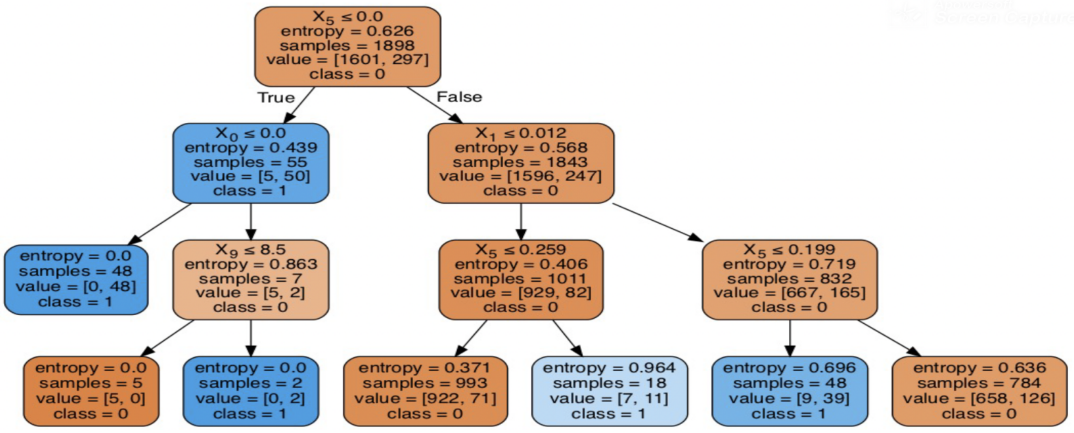


FIGURE 3.8: Decision tree to identify the threshold for each parameter for different classes

Initial results were seen once the shift towards the more classical Support Vector Machines and Decision Trees (Fig. 3.8) was made to classify the processes based on the process information. The decision trees specifically were helpful in not only identifying the critical parameters for the classification of quality parameters but also in determining some of the data quality issues. The decision trees would often pick up noise in the data and present it as a rule for classification. During the verification process, the domain experts would pick these rules as noise and then the data would be filtered further for those values or parameters. Various experiments were carried out to determine parameters that appear to affect the outcome to the model.

No substantial results for a conclusion could be drawn from the deep and machine learning models. However, upon the application of permutation feature importance [9], the common predictor found in both support vectors and decision trees was mainly the pump down vacuum and the pressure.

3.4.8 Results

As discussed earlier, different deep learning models including CNNs and LSTMs were trained but the accuracy's were never adequate for implementation. Fig. 3.9 depicts different CNN models which were trained with different parameters. The result from the

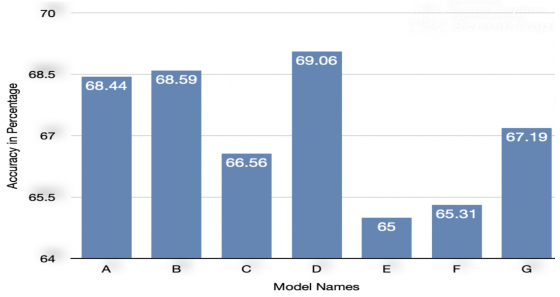


FIGURE 3.9: Accuracies of different convolutional neural network models where each model was trained with different input parameters

permutation feature importance was in accordance with the domain knowledge. The results for permutation feature importance and correlation analysis confirmed the process engineers' observations. So this project acted as a proof of concept for both industrial partners. At this phase since no ML model was ready for execution, a technical implementation phase of DMME was not required.

3.5 Case Study Insights

This section reviews the usefulness and relevance of both process models in industry 4.0 predictive maintenance setup.

As stated earlier, this project has been beneficial for all the partners involved. For Partner 1, this project helped them verify the proof of concept. Similarly, for Partner 2, the number of processes has increased before the down time, which serves already a direct monetary benefit to them. No conclusive link was found between the process parameters and the quality parameter through the deep learning models. This might be contributed to the qualitative nature of the predictor.

During the execution of the case study the following short-comings of the CRISP-DM and DMME have been identified:

1. Lack of business case
2. Lack of data quality description

3. Lack of role descriptions

These are in-line with the findings in [156] that CRISP-DM and DMME are most suitable for goal oriented and process driven projects but not exploratory data science projects.

3.5.1 Business Case

As mentioned in [222], a concrete business case is often missing in real industrial collaborations. Usually, the problem definition is high level and abstract. This has to be defined as precisely as possible.

The CRISP-DM is not very useful for business understanding and developing a business case that is viable for all co-operation partners. To this end, one must search outside the paradigms of this process model and proceed in an agile manner.

This is in contrast to the findings in [212], where a lack of deployment was considered as major short-coming. However, in our experience the lack of business case is a much more fundamental problem. For instance, Partner 1 was lacking a clear business case for its business, but was rather interested in "learning" about the relationship of complex process parameters to quality. Although these were identified, it was never clear, which level of prediction accuracy was necessary to achieve a real business benefit. On the other had, had Partner 2 identified the reduction of the cleansing intervals as a clear business case from the beginning, the extension from 24 to 36 would have been regarded as a huge success. Furthermore, while Partner 2 had a business case for investigating the relationship of complex process parameters to quality, it was not quantifiable.

While these facts seem to be not uncommon and rather typical for exploratory data analysis projects, CRISP-DM neither systematically addresses any of these issues nor helps to manage them.

3.5.2 Data Quality Description

The data must be evaluated at this initial phase to be able to go forward. The CRISP-DM and DMME do not lay out concrete measures to determine the quality of the data in the initial phases. Both briefly mention the importance of evaluating and creating a data quality report. But no guidelines on how to achieve that are presented. Data quality can be one of the largest obstacles in any data mining project which deals with real data sets. One of the main challenges was the manual determination of the quality parameter.

3.5.3 Role Description

Similarly, CRISP-DM does not provide any insight on how to obtain the domain-related knowledge from the industrial experts to be used for data mining procedures such as ML and DL. Although this problem is addressed partially by DMME in their technical understanding phase, it is still unclear which resources can bridge the gap between the knowledge acquired through experience by the process engineers and translate that knowledge into the actual modeling phase. Another related issue found is that the roles participating in each phase are not clearly highlighted in the DMME or original CRISP-DM. In [99], an overview of each role is presented.

3.6 Discussion and Future Work

Since all the latest machinery has sensor data available, most industries incorrectly assume that they are ready to move towards Industry 4.0 with the sensor data. The availability of data alone is deemed enough to start a data mining project.

As part of future direction, in order to address the lack of business case, we would like to merge existing business modelling frameworks with CRISP-DM and DMME to close this particular gap.

To address the data quality issue we will rely on more quantitative measures. For this particular project, we will use a spectrometer reading for instance that can be used since qualitative measurements have a lot of noise present to clearly indicate a relationship between parameters and the end parameter.

Furthermore, as future work, based on [99] we would like to objectify more concretely the role of each contributor in the project so that it is quantified and useful to engage similar roles in future projects using CRISP-DM and DMME.

3.7 From Industrial Time Series to Bio-medical Time Series

This study started with a focus on time series analysis for industrial domain. To this end, the research group INdustrial DAta Science (INDAS) started actively started acquisition of potential industrial collaborations. The acquisition of industrial partners has been the most time and effort taking step. The acquisition steps include any intermediate phases including but not limited to the initial contact, development of a non disclosure agreement, sharing of the data, and availability of the data to the analysts etc. Immediately

after the start of this doctoral studies, Covid-19 emerged. Due to this, the acquisition became more challenging because the focus of the industrial partners shifted from research to survival. Even after Covid-19, industrial partners are still recovering from the aftermaths of the pandemic. Some of the collaborations after the acquisition stage had to be terminated. The main grounds for ending the collaboration were the differences in objectives or finding later on that the raised hypothesis is not suitable for data science application.

After the Bühler/Optovision co-operation, many co operations are still under negotiation, e.g. with Deutsche Bahn, Boeing, Samson etc. But the duration of acquisition and meaningful application of data science to the industrial use-cases had exceeded the time frame of a normal doctoral study, that is why the shift towards different time series was essential. Another crucial aspect of working with industrial data set is that quality of data is almost never fit for immediate analysis or application of different ML models. This takes away most of the effort from algorithm development or research for time series analysis towards data cleaning. However, moving towards other time series than industrial production data set is not a diversion of objectives rather an expansion of them because the principle problem remains the same i.e. time series analysis with machine and deep learning techniques.

3.8 Conclusion

This case study was successful in implementing the combination of CRISP-DM and DMME process on an industrial data set. The implementation included most of the DMME phases except for deployment and technical implementations. The data mining techniques used in this study ranging from linear correlation analysis to ML models like SVMs and Decision Trees also helped to indicate which parameters can affect the outcome of the AR-coating process. The parameters found were in line with the domain understanding of the process experts. Though DL or ML models did not produce sufficient results (in terms of model evaluation criteria such as accuracy, evaluation matrices etc.) to be deployed as an end product, still the results obtained were in line with the domain understanding. In the areas where CRISP-DM or DMME did not provide any concrete guidelines, the study had to take an agile approach and rely on the experience of senior management to navigate the project.

Chapter 4

ECG Signals - Construct and Analysis

'Things are never simple when it comes to the human heart.'– (Gillian Jacobs)

4.1 Time Series Classification

Time series classification is a classical problem and many ways exist in literature to handle it. Since major part of this thesis is based on the ECG signals classification, this chapter overviews the origin and acquisition process of the signals. Later in the chapter, an overview of the state of the art for ECG signal analysis and classification is also presented.

4.2 ECG Signals

The ECG signal records the electric activity of the heart. It was first invented by a Dutch physician William Einthoven in 1902. In 1910, arrhytmia and the patterns associated with angina were discovered with the help of ECG and Einthoven was awarded Nobel prize in Medicine in 1924 for his revolutionary work[209].

In order to understand the construct of the ECG signals, an overview of the heart structure and its working is presented briefly in the section below. The following section has been extracted mainly from my Master thesis [33].

4.2.1 Structure And Physiology Of Human Heart

The heart is a muscular organ which pumps and supplies the blood to different parts of the human body. It is situated between the thoracic cavity and the lungs space called mediastinum. It is often referred to as a pump due to the fact that a contraction mechanism is used to develop a pressure which in turn expels the blood into major channels called aorta and pulmonary trunk [15][21].

A human heart resembles a pinecone. It is usually 12 cm in length, 8cm wide and 6 cm in thickness, almost the size of our fist. The weight and size differ in males and females. The heart of a well-trained athlete can be considerably larger than a normal heart [21].

The heart is composed of a right heart pump and a left heart pump. An atrium and a ventricle further make up two chamber pumps of each of these hearts. So in total, a heart has four chambers. Each atrium helps to move the blood into the ventricle and is the weaker primer pump.

Now the ventricles, supplying the main pumping force, thrust the blood either through the pulmonary circulation by the right ventricle or through the peripheral circulation by the left ventricle[15]. Fig 4.1 represents a human heart with its different parts and the passage of blood in them.

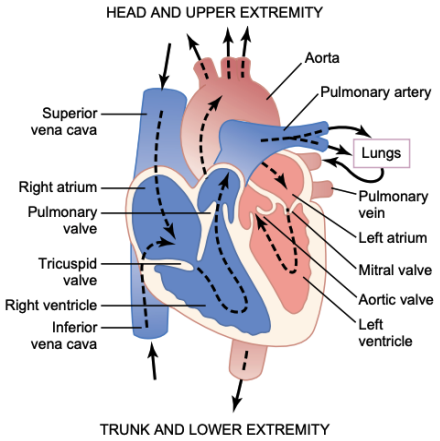


FIGURE 4.1: Structure of the heart, and course of blood flow through the heart chambers and heart valves.[15]

4.2.2 Cardiac Muscles And Electrical Activity

The heart is made up of three primary types of cardiac muscles: atrial muscles, ventricular muscle, and conductive muscle fibers. Atrial and ventricular muscles share few characteristics with skeletal and smooth muscles. That is one of the reasons why muscle

noise is embedded within the ECG signals because of the similarity in the structure. The conductive muscles of the heart have exceptional properties to initiate an electrical potential at a fixed rate that triggers the contractile mechanism by spreading from cell to cell. This property is known as autorhythmicity [21]. Though the cardiac muscles have autorhythmicity, heart rate is regulated by the endocrine and nervous system.

The heart is made up of two major types of muscle cells:

- 99% of the cells in atria and ventricles are made up of myocardial contractile cells. The impulses are conducted by these cells, and they are responsible for contractions that pump blood through the body.
- The remaining 1% of the cells are myocardial conducting cells which form the conduction system of the heart. They are functionally similar to neurons. The electrical impulse or action potential initiated by these cells propagate and travels throughout the heart and triggers the contractions to push the blood.

A fully developed human adult heart is capable of generating its own electrical impulse, which is activated by the cells pacing the fastest, as part of a system called the cardiac conduction system. The cardiac conduction system consists of the sinoatrial node, the atrioventricular node, the atrioventricular bundle, the atrioventricular bundle branches, and the Purkinje cells. The activation of heart muscle cells starts with the change in its initial electrical potential via a membrane-bound ion in- and outward flux. This event is called depolarization which is followed by repolarization. Repolarization is the successful recovery of the initial ion concentration state between intra-cellular and extra-cellular space [15][21].

4.2.3 Sinoatrial (SA) Node

The sinoatrial node is located close to the opening of the superior vena cava, in the walls of the right atrium. It is a specialized clump of myocardial conducting calls. Due to its highest built-in rate of depolarization, it creates the normal cardiac rhythm and hence, is also called the pacemaker of the heart. The electrical pattern known as sinus rhythm starts from this SA node and is followed by contraction of the heart. After initialization, the impulse spreads throughout the atria and the atrioventricular (A-V) node. A time duration of approximately 50 ms is taken by the impulse to travel between two nodes. The muscular contraction is triggered by the current of depolarization that begins in the right atrium and spreads across the upper part of both atria and then down to ventricles. The blood is then pumped into ventricles by contracting from superior to inferior portions of the atria.

4.2.4 Atrioventricular (AV) Node

There is another clump of specialized myocardial conductive cells called the atrioventricular (AV) node which is situated in the lower section of the right atrium. Before the depolarization of AV node and transmission of the impulse to the AV bundle, a delay takes place, which is of approximately 100 ms. This delay is very critical to heart function. The heart cells complete their contraction that pumps the blood into ventricles before the transmission of the impulse to ventricles cells itself during this delay. The impulses can be transmitted maximally at 220 per minute by the AV node with extreme simulations by SA node. This duration corresponds to the typical maximum heart rate of a young adult [21].

"Another cluster of cells called the Purkinje fibers spread the impulse to the myocardial contractile cells in the ventricles. They are located throughout the myocardium extending from the apex of the heart towards the atrioventricular septum and the base of the heart. These fibers have a fast conduction rate and conduct the impulse to all of the ventricular muscle cells in about 75 ms. The contraction begins at the apex with the electrical stimulus and travels towards the base of the heart in a similar manner as squeezing toothpaste from its bottom. This results in the blood pumping out of the ventricles and into the aorta and pulmonary trunk. The total time from the initialization the impulse in the SA node until depolarization of the ventricles in approximately 225 ms" [15][21].

4.3 The Cardiac Cycle

The cardiac cycle is defined as an activity of the heart that takes place from the onset of one heartbeat to the next one. Each cardiac cycle begins with the generation of an action potential in the SA node like explained earlier. The cycle consists of diastole (a period of relaxation) followed by systole (a period of contraction).

4.3.1 Phases Of The Cardiac Cycle

The blood flows according to the pressure gradient. The cycle begins with diastole, during which both atria and ventricles are relaxed. Blood flows in the right atrium from the superior and inferior venae cavae. Similarly, the four pulmonary veins allow the blood to flow in the coronary sinus and the left atrium. The tricuspid and mitral valves, both atrioventricular valves open allowing the blood to flow from atria to ventricles.

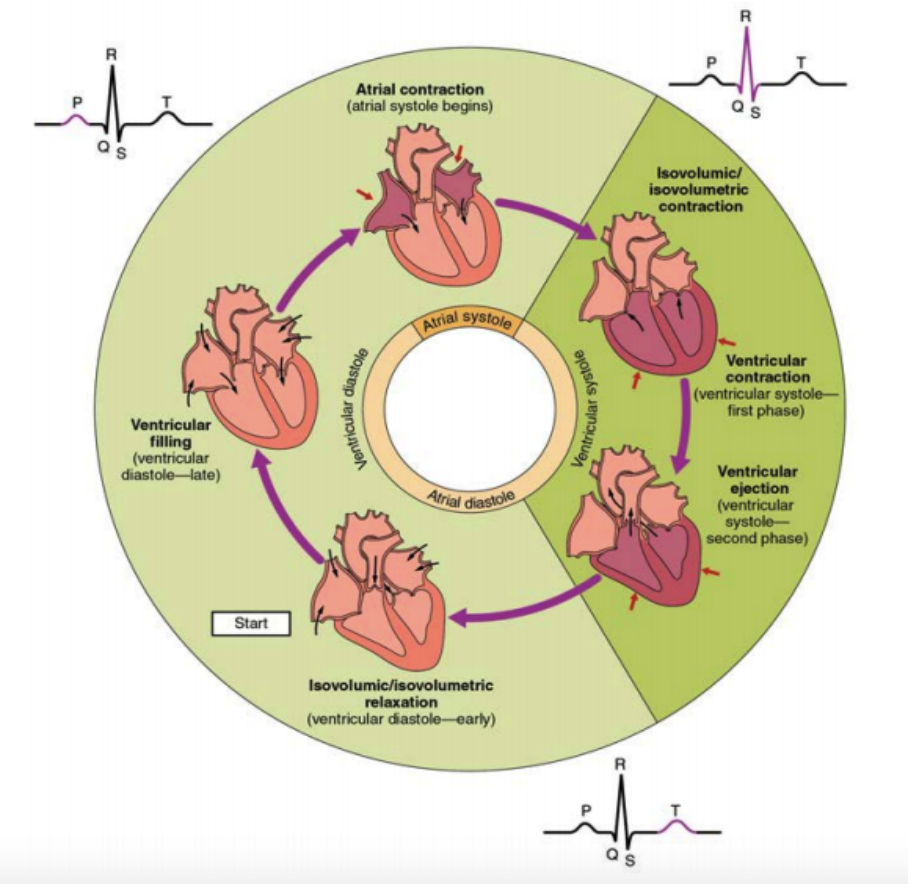


FIGURE 4.2: The cardiac cycle starting with atrial systole and progressing to ventricular systole,atrial diastole, and ventricular diastole. Corresponding ECG correlation is highlighted[21]

Approximately, 70-80% of the ventricular filling occurs in this way. The two valves, the pulmonary and aortic valves, are closed to prevent the backflow of blood into the ventricles from the pulmonary trunk on the right and aorta on the left [21].

4.3.2 Atrial Systole And Diastole

Due to the contraction of atrial muscles, the pressure builds up within the atria resulting in the blood being pumped into the ventricles through the open valves. In the start of atrial systole, ventricles are filled up to approximately 70-80% of their capacity due to the inflow of blood during diastole. The contraction of atria prior to ventricular contraction takes place to pump blood into the ventricles before the strong ventricular contraction begins. Thus the role of atria here is mainly that of the basal pumps for the ventricles whereas ventricles provide the major source of power for the blood flow through the body's vascular system.

The duration of atrial systole is approximately 100 ms, and it concludes before the start of ventricular systole bringing atrial muscles to rest(diastole) [21].

4.3.3 Ventricular Systole

Ventricular systole is caused by the depolarization of the ventricles. It is further divided into two phases. Initially, with the contraction of ventricles, the blood pressure within the chamber rises but not high enough to open the pulmonary and aortic valves and to be ejected from the heart. This results in an increase in blood pressure above the atria (which are now in diastole).

This causes the blood to flow back towards the atria closing tricuspid and mitral valves. In the later phase, the pressure within the ventricles is higher than the pressures in the pulmonary trunk and the aorta due to the ventricular muscle contraction. Blood is pumped from the heart by push opening the pulmonary and aortic semilunar valves. Since the existing pressure in the aorta is much higher, the pressure generated by the left ventricle is considerably higher than the pressure generated by the right ventricle. Despite this fact, both ventricles pump the same amount of blood known as stroke volume. The ventricular systole lasts a total of 270ms [15][21].

4.3.4 Ventricular Diastole

This phase follows the repolarization of the ventricles. In its initial stage, the ventricular muscle relaxes resulting in the decrease in pressure on the remaining blood within the ventricles. When the pressure within ventricles is lower than the pressure in both the pulmonary trunk and aorta, the blood flows back toward the heart, producing a dicrotic notch (small dip) which can be seen in blood pressure tracing. Now the semilunar valves close preventing the backflow of blood into the heart. There is also no change in the volume of the blood because the atrioventricular valves remain closed at this point. This early phase of ventricular diastole is called isovolumic ventricular relaxation period.

In the second phase of ventricular diastole, pressure on the blood within the ventricles drops even further due to relaxation of the ventricular muscle. When the pressure drops below the pressure in atria, the blood flows from the atria into the ventricles by push opening the tricuspid and mitral valves. The blood flows from major veins into the relaxed atria and from there into the ventricles.

The ventricular diastole lasts approximately 430 ms. Both chambers are now in diastole, the atrioventricular valves are open, and the semilunar valves remain closed. The cardiac cycle is complete.

4.4 The Normal Electrocardiogram

The cardiac impulse passes through the heart causing the electrical current to spread from the heart to adjacent tissues. A small current extends all the way to the surface of the body. The electrodes placed on the skin can effectively detect the current on opposite sides of the heart, and record the electrical potentials generated by the current [15].

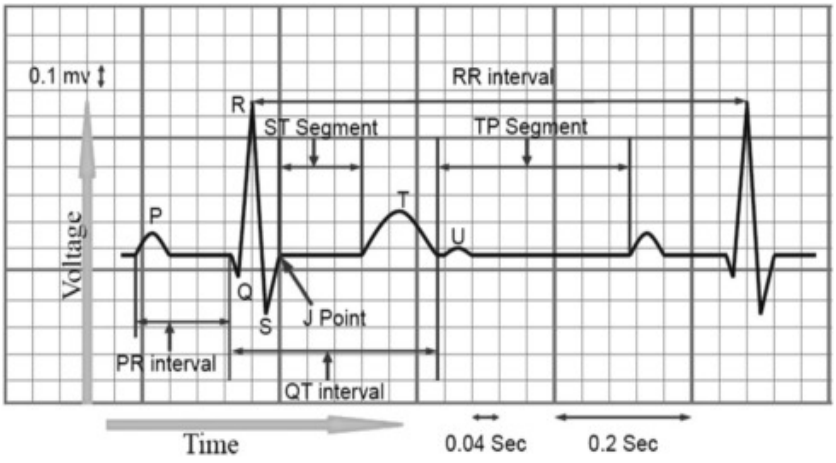


FIGURE 4.3: A normal ECG [140]

The normal ECG is composed of a P wave, QRS complex and a T wave (Fig 4.3).

The P wave is generated due to the depolarization of the atria. The atrial depolarization is followed by atrial contraction (systole) which continues until QRS complex. The atria re-polarize about 0.15 to 0.20 seconds after termination of the P wave which is about the same time when the QRS is being recorded. The QRS complex represents depolarization of ventricles before contraction and the electric potential generated due to this depolarization. Both the P wave and components of the QRS complex are known as depolarization waves. The normal duration for the peaks is presented in Table 4.1

The repolarization of the ventricles creates the T wave and marks the beginning of ventricular relaxation. No potential is shown on ECG when the ventricular muscle is completely polarized or depolarized. The current can only travel from one part of ventricular to another when the ventricular muscle is either partially polarized or partially depolarized hence shown on the ECG.

Fig 4.3 charts out the normal ECG ranges for each interval. Ten electrodes are placed on the patient's limbs and the chest in a conventional 12 lead ECG. That provides 12 different angles to measure the magnitude of the heart's electrical potential. So at each moment, the magnitude and direction of the heart's electrical depolarization are measured during the cardiac cycle. A graph is plotted between corresponding voltage versus time.

Feature	Normal Value	Normal Limit
P width	110 ms	±20 ms
PR interval	160 ms	±40 ms
QRS width	100 ms	±20 ms
QTc (corrected) interval	400 ms	±40 ms
P amplitude	0.15 mV	±0.05 mV
QRS height	1.5 mV	±0.5 mV
ST level	0 mV	±0.1 mV
T amplitude	0.3 mV	±0.2 mV

TABLE 4.1: Typical lead II ECG features and their normal values in the sinus rhythm at a heart rate of 60 bpm for a healthy male adult [52]

4.4.1 The ECG Electrodes And Leads

Two electrodes are placed on different sides of the heart to record an electrocardiogram. A lead is composed of two wires and their electrodes to make a complete circuit between the body and the electrocardiograph [21].

The term lead means an imaginary line between two ECG electrodes. During ECG the electrical activity of different leads is measured and recorded. The electrodes are wires attached to the body of the patient to record the ECG, and they allow respective leads to be calculated [68].

The electrical activity of different parts of the heart muscles is calculated by placing electrodes on different sides of the heart. The ECG displays the voltage between the pairs of these electrodes and the muscle activity measured by those electrodes.

In a 12 lead ECG, three different types of leads called bipolar limb leads, Unipolar limb leads, and Unipolar precordial (chest) leads are used, each looking at the different angle of the heart. The bipolar limb leads, also known as standard leads, are called leads I, II and III. They use a single positive electrode and a single negative electrode between which electrical potentials are measured. The bipolar leads view the frontal plane of the heart from these two points [20].

4.4.2 Einthoven's Triangle

Einthoven's Triangle as shown in fig 4.4 is drawn around the area of the heart. The top of a triangle, surrounding the heart, is formed by two arms and left leg.

The upper two corners of the triangle represent the points where fluids around the heart are connected electrically with the two arms and the lower corner of the triangle represent the point at which the left leg connects with the fluids [21].

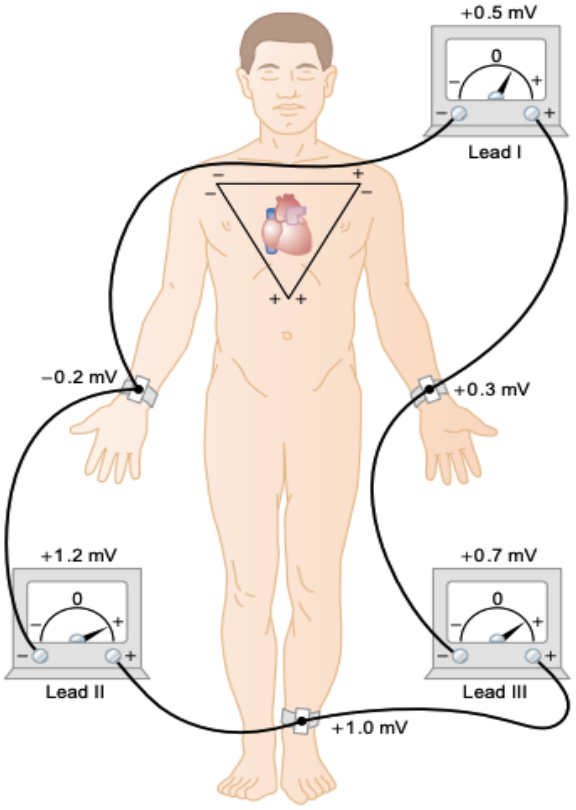


FIGURE 4.4: Conventional arrangement of electrodes for recording the standard electrocardiographic leads superimposing the Einthoven's triangle[21]

At any given point in time during the cardiac cycle, the heart current flows in a particular direction in the heart. A vector representation is used to measure the electrical potential generated by the current flow. An arrow pointing in the direction of the electrical potential with the arrowhead in the positive direction is used. The arrow length is drawn proportional to the voltage of the potential [21]. The direction of the vector is different at different points of the cardiac cycle and is captured by different leads.The electrical heart axis is the direction and size of the vector of the electric heart field. The changes or fluctuations in the electric heart field can be detected only during depolarization or repolarization.

4.5 Systematic Methodology of ECG Analysis – An Overview

ECG is one of the most important parameter indicating one's physiological well being. It is extensively used for evaluating the cardiac situation of the patients [216]. The periodicity in the ECG signals over time has made it effective for several non invasive

biomedical applications like heart monitoring, blood glucose monitoring [243], bio-metric identification [173], emotion recognition[97], fall detection [34] and prevention [238].

Studies like [120], [165] and [52] presents a comprehensive overview of the different ECG analysis techniques. The analysis of ECG, like most of the time series analysis, involves pre-processing, feature extraction, feature selection, feature transformation, classification and interpretation as shown in Fig.4.5 .

ECG signals requires the preprocessing techniques for noise reduction since the signals are contaminated with different types of noise and artifacts during the acquisition process. Major types of noises include power line interference, baseline wander, electrode contact noise, electrode motion artifacts, muscle contractions, electrosurgical noise, and instrumentation noise [120]. More details on ECG noise and filtration can be found in Chapter 5, section 5.5.2. The prime function of applying preprocessing methods is to make the peaks and joints of the ECG signals, like QRS(onset and offset), RR interval, P-onset, T-onset clearer and reliable to further processing. The major preprocessing techniques include filtering, resampling, digitization, artifact removal and normalization.

This section outlines the systematic methodology to time series analysis with focus on the ECG signals.

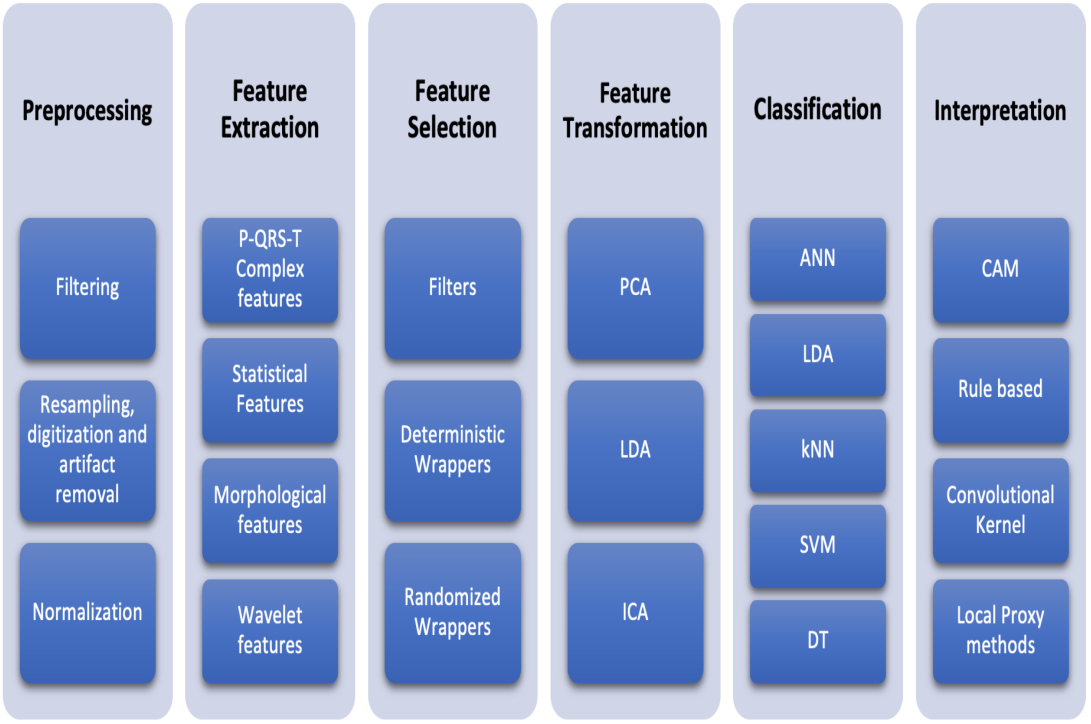


FIGURE 4.5: Systematic Methodology for Analyzing a Time Series in general and ECG in particular

4.5.1 Feature Extraction

One of the eventual steps after preprocessing is to extract relevant features for the analysis. Feature extraction is a vital step specially for bio-medical signals because of their unique nature. Physiological signals are non-stationary, non-linear, non-Gaussian and non-short form which poses its own challenges in analysis.

Feature extraction aims to reduce dimensionality and compress data, allowing for a more concise representation using a subset of features. This streamlined data representation is advantageous for optimizing machine learning and artificial intelligence models, particularly in tasks like classification and diagnosis. Additionally, feature extraction filters out redundant data from the dataset, retaining only the essential information of interest.

The features can be extracted both manually and automatically depending on the application. Several techniques exist to extract features based on the domain knowledge. Most common feature extraction techniques for ECG analysis include P-QRS-T complex features, statistical features, morphological features, and wavelet features. Singh

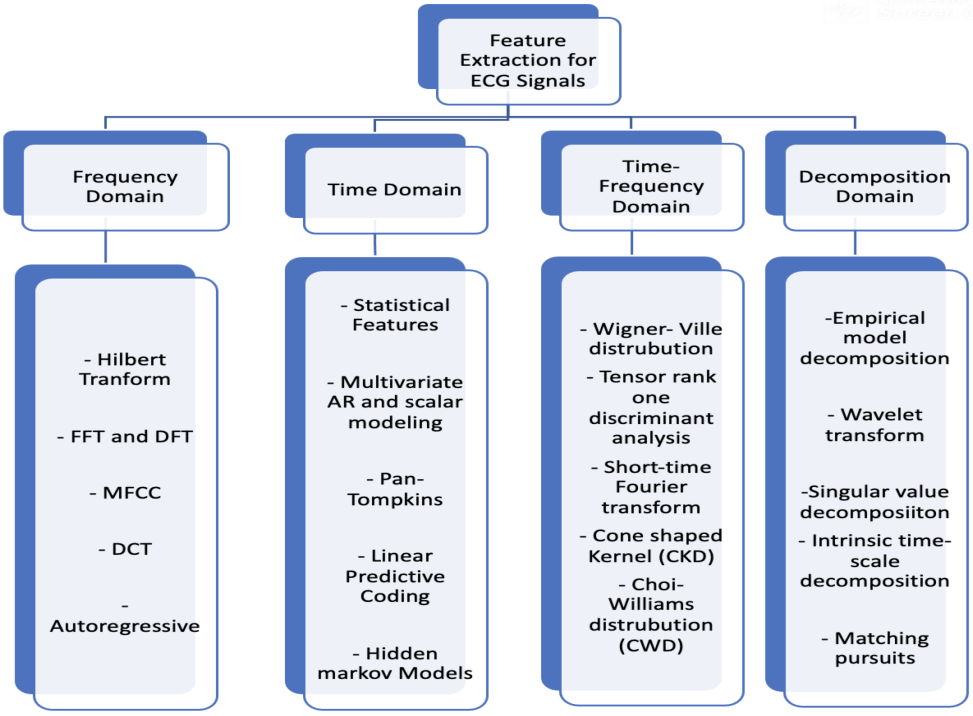


FIGURE 4.6: An overview of the Feature extraction techniques fro ECG signals

et al. [225] categorizes the many existing feature extraction technique for ECG signals into four domains called time domain, frequency domain, time–frequency domain and decomposition domain as shown in Fig 4.6. For details of each algorithm, please refer to

the original reference at [225]. In this dissertation, mainly wavelet transformation from decomposition domain were used as feature extraction techniques.

4.5.2 Feature Selection

Feature extraction can still result in high dimensional features which makes the ECG signals difficult to model since many pattern recognition techniques are originally not designed to cope with higher dimensional feature space. It is not only computationally inefficient but also effects the performance of the classifiers in later steps. Hence, some methods should be employed to select the relevant features after the extraction process. The selection can be automatic or based on the domain knowledge.

Saeys et al. [203] categorizes the feature selection methods into three kinds: filter, wrapper and embedded as shown in Fig.4.7.

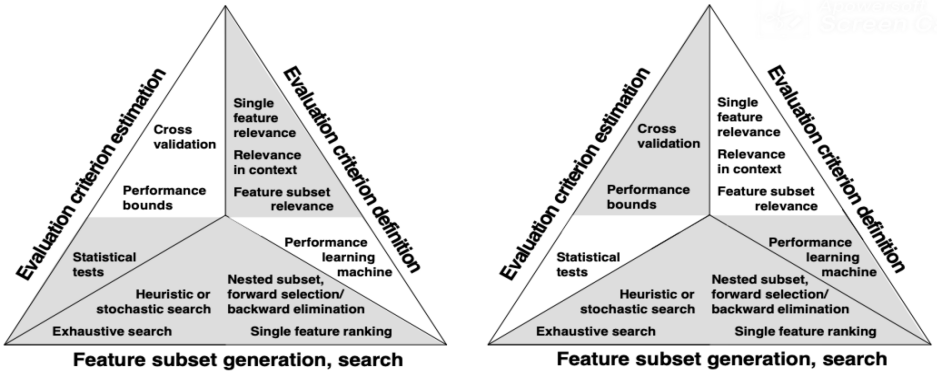
The filter based selection uses scoring mechanism to score relevant features and filter out the ones with lower scores. The scoring is independent of the classifier or the learning model. Some of the filter based feature selection for ECG signals includes correlation criteria, Fisher score, information gain-based selection, mutual information techniques, fuzzy clustering and rough sets based selection. One of the drawbacks of the filter based method is that the interactions between different features with each other and the classifiers are not taken into account.

The Wrapper methods tackle this problem by integrating the model hypothesis search into the feature subset search. This makes them computationally heavier than the Filter methods. Some of commonly used wrapper methods for ECG are sequential feature selection method, sequential floating feature selection algorithm, kNN based sequential forward selection algorithm and forward selection and backward selection SVM with Gaussian RBF kernel.

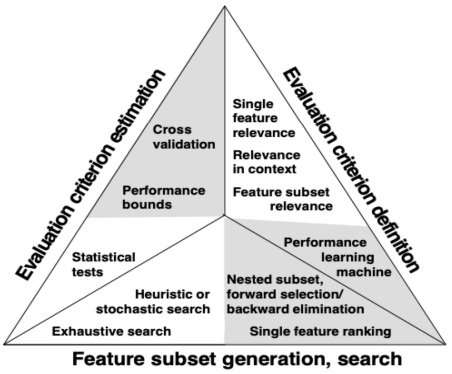
On the other hand, in the embedded method the feature selection is entwined in the training process of the classifier. Therefore the embedded methods are specialized for the respective trained classifiers. Other commonly used feature selection method for ECG analysis is genetic algorithms (GA) based selection approach.

4.5.3 Feature Transformation

Feature transformation reduces the feature subspace by transforming the original space into a lower dimensional subspace. This is why both the terms, feature transformation



(a) Filters (b) Wrappers



(c) Embedded methods

FIGURE 4.7: The three principal approaches of feature selection. The shades show the components used by the three approaches: filters, wrappers and embedded methods [93]

and feature reduction, are often used interchangeably. However in feature selection, the feature subspace is reduced by selecting a discriminate subset from original feature space.

Principal component analysis or PCA is one of the most common methods to reduce signal dimensional by achieving a linear mapping of high dimensional input vector into low dimensional vector whose components are uncorrelated. It was presented initially by Pearson [180], as more fitting for modelling response data than the analysis of variance. It was later developed to its present form by Hotelling [102].

The principal components (PC) are derived from the signal $\mathbf{x}$ which is assumed to be a zero-mean random process and is characterized by the correlation $\mathbf{R_x} = E[\mathbf{xx}]^T$ [43]. The principal components are derived by applying an orthonormal linear transformation $\mathbf{\Psi} = [\Psi_1, \Psi_2, ...\Psi_N]$ to $\mathbf{x}$,

$$\mathbf{w} = \Psi^T \mathbf{x}, \tag{4.1}$$

so that the elements of the principal component vector $\mathbf{w} = [w_1, w_2, ...w_N]^T$ become mutually uncorrelated. The first PC is obtained as scalar product $w_1 = \Psi_1^T \mathbf{x}$, where the

vector Ψ_1 is chosen so that the variance of w_1, given by:

$$E[w_1^2] = E[\Psi^T \mathbf{x}\mathbf{x}^T \Psi] = \Psi_1^T \mathbf{R}_x \Psi_1, \tag{4.2}$$

is maximized subject to the constraint that $\Psi_1^T \Psi_1 = 1$. The maximum variance is obtained when Ψ_1 is chosen as the normalized eigenvector corresponding to the largest eigenvalue of $\mathbf{R}_x$, as denoted λ_1; the resulting variance is

$$E[w_1^2] = \Psi_1^T \mathbf{R}_x \Psi_1 = \lambda_1 \Psi_1^T \Psi_1 = \lambda, \tag{4.3}$$

Subject to the constraint that w_1 and the second principal component w_2 should be uncorrelated, w_2 is obtained by choosing Ψ_2 as the eigenvector corresponding to the second largest eigenvalue of $\mathbf{R}_x$, and so on until the variance of $\mathbf{x}$ is completely represented by $\mathbf{w}$ [43]. Some of the recent studies which implement PCA for ECG analysis are [155], [73], and [221].

Similarly, another feature transformation method found in literature is **Linear discriminant analysis** (LDA). Feature space is compressed by projecting the high dimensional features into optimal discriminant vector space thus extracting the classification information.

Let $T\mathbf{X} = \{x_1, x_2, ..., x_m\}$ be the n-dimensional data in R^n space, m is the number of training samples, and n is the dimensionality of training samples. The category label is given as $c_i \in \{1, 2, ..., k_c\}$,where k_c is the number of categories of the sample. The goal is to learn an optimal projection matrix W such that x_j is projected to y_j of the d-dimensional data ($d < n$),

$$y_j = W^T x_j \tag{4.4}$$

If $\mathbf{S_w}$ is in-class scatter matrix and $\mathbf{S_b}$ is a inter-class scatter matrix of samples, and m is the mean vector of all samples, $\mathbf{S_w}$ and $\mathbf{S_b}$ can be calculated as:

$$\mathbf{S_w} = \sum_{i=1}^{k_C} \sum_{j \in c_i} (x_j - m_i)(x_j - m_i)^T \tag{4.5}$$

$$\mathbf{S_b} = \sum_{i=1}^{k_C} (m_i - m)(m_i - m)^T \tag{4.6}$$

Independent component analysis (ICA) is another feature transformation technique which identifies the underlying independent factors statistically. It can identify and statistically separate out the individual sources from the mixtures without any prior information about the sources and the mixing parameters [104].

If observed variables from a process are denoted by $x_i(t)$, i=1,...T, where i is the observed data variable index and t is the time index. An observed signal $x_i(t)$ is assumed to be a combination of hidden (latent) variables $s_j(t), j = 1, ..., m$, and some unknown coefficients a_{ij}

$$\mathbf{x_i(t)} = \sum_{j=1}^{m} a_{ij} s_j(t) \quad \text{for all} \quad i = 1, ...n \tag{4.7}$$

The $s_i(t)$ are the independent component whereas the coefficient a_{ij} are called the mixing coefficients. Here only the variable $x_i(t)$ is observed and both a_{ij} and $s_i(t)$ have to inferred. The ICA algorithm aims to extract the data by a linear generative model such that the stochastic sources s_i are as mutually independent as possible[109].

Some of the recent studies using ICA for ECG analysis are [207][114](for classification) and [134] (for de-noising).

4.5.4 Classification

Classification is one of the key tasks in time series analysis. It has many applications areas like diagnostics [179] [19] [118], geriatrics [163], biometric identification [148] [246][67] and many more. The aim in classification is to assign each input vector to one of the finite number of discrete categories [23]. According to [229], formally a classifier C is a function mapping a feature space H into a set of predefined class label L

$$C : H \rightarrow L \tag{4.8}$$

Many classifiers exist in literature for ECG classification. They can be widely divided into categories including machine learning techniques named support vector machines, decision trees, k nearest neighbour(kNN), LDA and deep learning methods including artificial neural networks.

Support Vector Machine or SVM were introduced by Vapnik in 1992 [54]. Given an input example $X = (x_1, ...x_d)$ of d dimension into two classes. A decision function of SVM separates the two classes by $f(X) > 0$ or $f(X) < 0$. The size of the training set N is (y_i, X_i), $i = 1, ..., N$. Where $X_i \in R^n$ is the input pattern for the ith example, and $y_i \in \{-1, 1\}$ is the class label. Support Vector classifiers implicitly map X_i from input space to a higher dimensional feature space which depend on a non-linear function $\phi(X)$. A separating hyperplane is optimized by maximization of the margin. Then SVM is solved as the following quadratic programming problem,

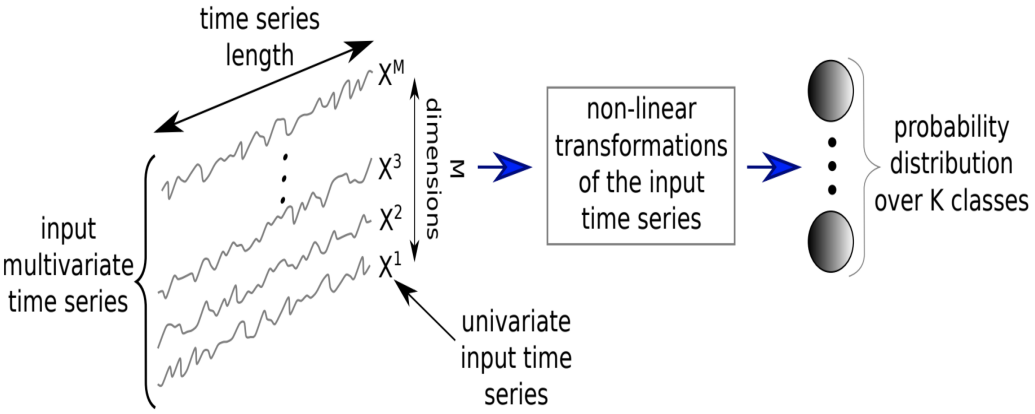


FIGURE 4.8: A unified deep learning framework for time series classification [111]

$$Maximize: \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K(K_i, X_j) \tag{4.9}$$

$$Subject\ to: 0 \leq \alpha_i \leq C(i = 1, ...n), \sum_{i=1}^{n} \alpha_i y_i = 0 \tag{4.10}$$

Where $\alpha \geq 0$ are Lagrange multipliers. Many α will be equal to 0 when the optimization has solved and the others would be Support Vectors. C is chosen according to the problem and is a positive constant. This parameter expresses degree of loosing constraint. A large C can classify training examples more correctly. $K\left(X, X'\right)$ is the kernel function which is inner-product defined by $K\left(X, X'\right) = \phi\left(X\right) \cdot \phi\left(X'\right)$. Then the SVM decision function is

$$f(X) = \sum_{X_i \in SV} \alpha_i y_i K(X_i, X) + b. \tag{4.11}$$

A common kernel is the Gaussian radial basis function (RBF) given by [276],

$$K(X, X') = e^{-\|X - X'\|^2 / 2\sigma^2} \tag{4.12}$$

An important characteristic of SVM machine is that model parameter determination is corresponded to a convex optimization problem, which means that any local solution is also a global optimum [23]. Many studies have used SVMs for classification of ECG signals. Some of the such recent studies include [46],[276], [193],[169], and [115]. Most of the SVM based classifiers would require manual feature extraction in the initial stages of data pre-processing. One of the main strength of SVM is its white box nature which makes it easier to be interpretable for explaining the decisions.

Another common classification white box model is **decision tree DT**. Its strength also lies in its internal pretable nature. The recent studies using DT for ECG classification are [274],[170],[138],and [205].

Artificial Neural Networks (ANN) are interconnected artificial neurons, which are interconnected with adjustable weights. The neurons are basically non linear discriminant functions, which are based on linear combinations of fixed non linear basis function $\phi_j(\mathbf{x})$. If $f(.)$ is a non linear activation function , the linear model in case of classification take the form

$$y(\mathbf{x}, \mathbf{w}) = f\left(\sum_{j=1}^{M} w_j \phi_j(\mathbf{x})\right) \tag{4.13}$$

This model is transformed to make the basis function $\phi_j(\mathbf{x})$ depend on parameters which can be adjusted along with coefficients w_j during training. For the input variables $x_1, ...x_D$, M linear combinations are constructed in the form:

$$a_j = \sum_{i=1}^{D} w_{ji}^{(1)} x_i + w_{j0}^{(1)} \tag{4.14}$$

Where a_j are known as activations and the parameters $w_{ji}^{(1)}$, $w_{j0}^{(1)}$ are *weights* and *biases* respectively. Also $j = 1, ...M$ and the superscript (1) indicates that the parameters belong to the first layer of the network. Each of them are transformed using a differentiable, non linear activation function $h(\cdot)$ to give

$$z_j = h(a_j) \tag{4.15}$$

These quantities correspond to the outputs of the basis functions and are called *hidden units*. The nonlinear functions $h(\cdot)$ can be logistic sigmoid, 'tanh', or ReLU (Rectified Linear Unit) [23]. The activation functions can be chosen based on the nature of the data and the assumed distribution of target variables.

There have been many models used for classification in literature. The most commonly used neural networks used recently for ECG classification are Convolutional Neural Network (CNN) [19], [107], [129], [267], [3], Recurrent Neural Network (RNN), Long Short Term Memory (LSTMs) [201], [103], [130], [268], transformers [164], [105], [48], [171], [106], and also ensemble models. The working mechanism of CNNs, transformers and LSTMs are discussed in detail in later chapters.

4.5.5 Explanation

Although the classification algorithms have leveraged the diagnostic process, they are mainly data driven and rely on the underlying representation of the data which makes it difficult to interpret the classification results. This, in turn, makes it more difficult to implement these diagnostic tools in practise.

Since the terms, explainability and interpretabilty like most of the other terms in the Explainable AI (XAI), lack a formal definition [141], [31]. However, interpretability for data science can be vaguely defined as the capability to provide an explanation in a human understandable terminology. [141] proposes to gain and define proper context and the the motives of the interpretability to have concrete outcome of explainability and interpretability laid out.

The explanation should act as an interface between algorithms and humans so that the decision is comprehensible and at the same time an accurate representation of the algorithm mechanism [90]. The absence of interpretability can raise issues in areas like technical (understanding and implementing a certain algorithm), ethical (e.g. for critical decisions in automated military operations) and legal (e.g. e General Data Protection Regulation (GDPR) implementation) [90], [85].

It is not only important but a prerequisite for AI models and analysis to be explainable and interpretable for the techniques to be implemented in safety critical systems such as the medical domain.

The models can be interpretable in two ways:

- The first way is to build an interpretable model from scratch. Traditional machine learning models like decision trees and support vectors are white boxes by design and the results are human interpretable. Bodini et. el [26], Neves et el [174] etc. discusses open box interpretibility techniques for ECG signals.
- The second approach is used when the model is complex. Here an effort is made to approximate the relationship between input and output in human understandable manner.

Many taxonomies exist in literature to catogarize XAI approaches like one shown in Fig.4.9. According to [88] and [240], many of the present methods for XAI can be categorized in the following way:

- **Ante-Hoc vs. Post-Hoc**

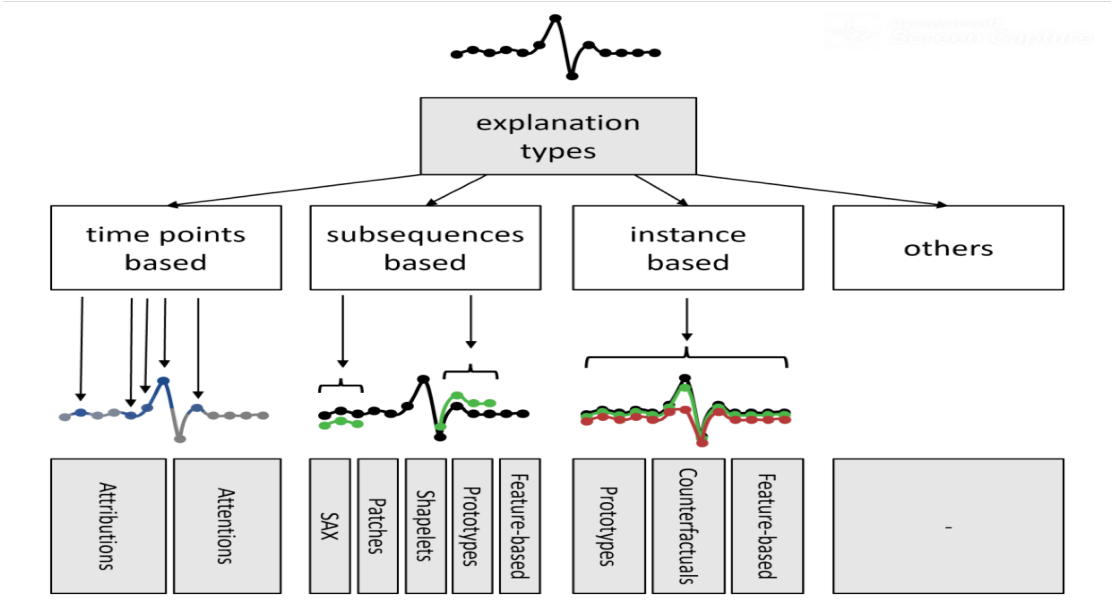


FIGURE 4.9: The proposed taxonomy categorizes the reviewed XAI approaches in different explanation types based on their explanations [240].

The post-hoc approach provides explanation in a manner that is separated from the model. The explainabilty is not built in the model. Thus, a post-hoc explainer would consist of a function g that takes as input the classifier f as well as a dataset D. It provides an insight into what the model has learned without changing its structure. e.g. Lime[197], Layer-wise Relevance Propagation (LRP)[17] and Black Box Explanations through Transparent Approximations (BETA) [137].

Ante-Hoc methods have the explainability built into the the model from the beginning. To train such model, the goals of explainability are clear right from the start. Recently more effort is being put into designing models which are ante hoc in nature. Some of the proposed methods are REverse Time AttentIoN model (RETAIN)[50], Bayesian deep learning (BDL)[1], Self-Explaining Neural Networks [10], Concept Bottleneck Models [126], Concept-based Model Extraction[131], and Concept Whitening[49].

- **Global vs. Local**

  This categorization refers to the kind of explanations obtained by applying explainablitly techniques. If an explanation approach describes the overall logic for the particular input instance, it is referred to a global explanation. Function g returns a generalized explanation for the decisions that are valid for the whole set X.

  On the other hand, if the approach can only explain certain instances or family of instances, it is a local explanation, i.e., g unveils the reasons for the classification only for a specific instance x. Naturally, obtaining a global explanation is

more challenging task than obtaining a local one. [240] uses another taxonomical structure to categorize the explanations for time series as shown in Fig. 4.9

- **Model-Agnostic vs. Model-Specific**

  Some of the interpretation techniques are unique for certain models and are only applicable with those particular models. e.g. using activation maps of intermediate layers in CNNs to understand the learning of the weights is model specific to CNNs. On the other hand, model-agnostic tools are meant to provide explanations regardless of the underlying structure or model and are used only after the model has been trained. These agnostic methods typically analyze pairs of feature inputs and outputs and provide the explanation on instance basis. It's crucial to emphasize that, by definition, such methods lack access to internal model details, such as weights or structural information [240].

Chapter 5

Fall Detection Using ECG Signals

> 'Knowing is not enough; we must apply. Willing is not enough; we must do.'–
> (Johann Wolfgang von Goethe)

This chapter is mainly based on the worked published in [34]. However it is not written in a 'verbum pro verbo' from the publication in this chapter. The work was aimed to detect human fall and detect different human activities using ECG signals detection.

In this study we aim to detect human activities, in particular fall events using wearable devices based on ECG sensor data. To this end we classify the activities with a pre-trained deep neural network, i.e. we apply a transfer learning approach to reduce training time and data.

The chapter is structured as follows: A brief description of the technologies used in our study is given in Section 5.2. Section 5.3 presents an overview of the different form of fall detecting techniques and human activity recognition (HAR) in literature along with an emphasis on our contribution to the field. Experimental setup and data collection process is illustrated in Section 5.4. Section 5.5 presents our proposed methodology and explains the algorithms used. It also describes the phases of our work and its implementation. Section 5.6 reviews the collected results and their implications. This leads to the discussion related to our research question in Section 5.7. Finally, Section 5.8 outlines the conclusion and gives some future perspective that can be used to further probe the field of HAR using ECG signals.

5.1 Introduction

The World Health Organization (WHO) defines fall as "unintentionally coming to the ground or some lower level and other than as a consequence of sustaining a violent blow, loss of consciousness, sudden onset of paralysis as in stroke or an epileptic seizure" [262].

According to a study, falls are the dominant cause of unintentional injury-related deaths and non-fatal injuries in people aged 65 and above. It poses a severe challenge for senior adults and people with movement disabilities [7]. The likelihood of falls increases with age and diminished health quality of an individual. The frequency of falls is more significant in seniors living in nursing homes than those who are living in the community. According to [242], approximately 30-50% of people residing in long-term care institutions fall per annum, and 40% of them experience repetitive falls. About two-thirds of people who suffered a fall are susceptible to recurrent falls. About 50% of the patients who lay on the floor for more than an hour after a fall died within six months of the fall [259].

A critical step in providing timely response to falls is detecting them as early as possible. Several studies and surveys have been conducted to categorize falls and detect them.

5.2 Background

For the reader's convenience, we briefly explain the technologies that were employed in this section.

Several fall detection systems exist in literature. A recent review of the fall detection systems in [256] categorizes them as follows: (i) wearable device, (ii) ambient system, (iii) image processing system, and (iv) combined systems. Wearable devices provide a cheaper and more practical solution in terms of freedom of movement and energy consumption. The aim of most fall detection systems is not only to detect a fall but also to inform concerned authorities in case of an urgent medical emergency. Most of the latest algorithms for fall detection use machine learning [82] and deep learning algorithms [146].

Deep learning became prominent in computer vision (CV) when a deep learning convolution neural network, AlexNet, outperformed its competitors in the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) in 2012 during an image classification task. Before the wide spread recognition of deep learning, features from the data had to be hand crafted and then fed into a machine learning model. Deep learning in general and CNNs in particular, have revolutionized the field by not only detecting the features themselves

but the features selected by them have been proven to be more significant than those crafted by hand [147].

According to [237], "transfer learning is a machine learning method where a learning model developed for a first learning task is reused as the starting point for a learning model in a second learning task" .

It can also be defined as the ability of a system to recognize and apply knowledge and skills learned in previous domains/tasks to novel domains/tasks, which share some commonality [178]. Transfer learning has been formally defined and categorized in [178]. The problem statement, notation and definition used here are also taken from [178]. A domain $\mathcal{D}$ is comprised of two components: a feature space $\mathcal{X}$ and a marginal probability $P(X)$, where $X = \{x_1, x_2, ...x_n\} \in \mathcal{X}$. Similarly, given a specific domain, $\mathcal{D} = \{X, P(X)\}$, a task consists of two components: a label space $\mathcal{Y}$ and an objective predictive function $f(\cdot)$ (denoted by $\mathcal{T} = \{\mathcal{Y}, f(\cdot)\}$), which is not observed but can be learned from the training data, which consist of pairs $\{x_i, y_i\}$, where $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$. The function $f(\cdot)$ can be used to predict the corresponding label, $f(x)$, of a new instance x. From a probabilistic viewpoint, f(x) can be written as P(Y|X).

Formally transfer learning is defined as, "given a source domain $\mathcal{D}_\mathcal{S}$ and learning task $\mathcal{T}_\mathcal{S}$, a target domain $\mathcal{D}_\mathcal{T}$ and learning task $\mathcal{T}_\mathcal{T}$, transfer learning aims to help improve the learning of the target predictive function $f_\mathcal{T}(.)$ in $\mathcal{D}_\mathcal{T}$ using the knowledge in $\mathcal{D}_\mathcal{S}$ and $\mathcal{T}_\mathcal{S}$, where $\mathcal{D}_\mathcal{S} \neq \mathcal{D}_\mathcal{T}$, or $\mathcal{T}_\mathcal{S} \neq \mathcal{T}_\mathcal{T}$".

Transferring knowledge from one domain to another using transfer learning is not only of theoretical interest, but also of great practical importance, as it can spare a lot of training time and resources and by reducing the data necessary in a domain, it can make learning feasible at all. In our work we have transferred the knowledge gained from training neural networks with natural images to the domain of synthetic, artificial medical images in order to classify images, that were created as a result of pre-processing medical data such as ECG data.

5.3 Related Work and Our Contribution

This section reviews the work that has been previously done to detect falls using different methods and highlights our contribution to the field. The ECG signals predict the overall health of the human body. A considerable amount of effort has already been contributed to the area of fall detection using accelerometers and gyroscopes. Some studies, like that in [245], have used different machine learning techniques on these sensor readings to predict and detect the fall. Few studies use ECG signals to predict the falls like in [161],

but our study uses the frequency-time domain of the ECG signals by computing their continuous wavelet transform (CWT) coefficients and converting them into scalograms. The overall idea is mainly inspired by [159] where a convolutional neural network (CNN) has been trained on scalograms to differentiate between different heart diseases using ECG signals. Similar approach was used in our study to differentiate between falls and no-falls ECG signals.

Detecting fall comes under the vast umbrella of human activity recognition (HAR). HAR is usually considered a computer vision (CV) problem where CV algorithms are used on images or videos to distinguish one activity from another. However, other device free solutions based on e.g. radio signals such as received signal strength (RSSI) or channel state information (CSI) exist, see e.g. [255], [172], and [58]. Alternatively, wearable sensors provide a resource friendly and practical solution to real time activity recognition, e.g. in particular the rising popularity of gadgets such as smartwatches is an indicator for this trend. Many studies like [116] and [64] infuse readings from accelerometers and ECG sensors at decision level to determine an activity. Though the wavelet transform has been previously used for analyzing ECG signals for detecting different cardiac conditions as reviewed in [4], it has never been used separately to differentiate a fall from non-fall. Similarly [123] analyzes the impact of body movements on ambulatory ECG frequency spectrum. It also uses artificial neural networks to classify different body movements.

The CNN we utilise has also been used in [100] to extract features for bio-metric purposes using gait features from sensor data. However, in our work, for the first time we classify the following activities using ECG signals: Fall, Resting, and general Daily Activities, i.e. we distinguish not only Fall from Resting but also Fall from Daily Activities, where Daily Activities refer to generic daily activities performed by the subjects. The fall risk using heart rate variability in combination with data mining techniques is assessed in [162].

In this chapter, we have tried to explore the research question: "Can a fall be detected by using only ECG signals?", by collecting the ECG signals of people falling and then applying our proposed algorithm on the signals. The purpose of this study is to use the biomedical electrical signals of the heart via electrocardiogram to train a deep neural network to analyze and observe the patterns of ECG before and during the fall. Although a lot of work has already been done to detect falls using accelerometers and gyroscopes along with heart rate variability, the focus of this study has been to use ECG as the only factor to determine the presence of fall. Here we are not referring to the cases where cause of the fall is due to a specific heart condition.

This study has explored the less studied field of ECG signals for fall detection by applying machine learning techniques on it. It is based partially on [33] but extends the results obtained therein substantially.

In the field of medical imaging, finding appropriate amount of data has always been a challenge. It is mainly due to two reasons: Firstly, because of strict data privacy issues and secondly finding a large group of volunteers for conducting experiments can be challenging. A data augmentation technique for time series called slicing was used to enhance the limited dataset.

Henceforth, for the machine learning domain, we have used transfer learning and compared two of the popular pre-trained networks for image classification – AlexNet and GoogLeNet. We have demonstrated that it could be really beneficial to use a pre-trained network in medical imaging domain as it does not require a lot of data. However, we would like to emphasize that it is an initial study with focus on the proof of concept in a laboratory set up. It is not conclusive enough to be deployed but it does provide a strong baseline to work further on it.

5.4 Experimental Setup and Data Collection

There are some online ECG databases available like ECGVIEW [127] and physionet [84] which provide ECGs for different research purposes. The ECG databases for people falling could not be found so the ECG data was collected as part of the experiment. The

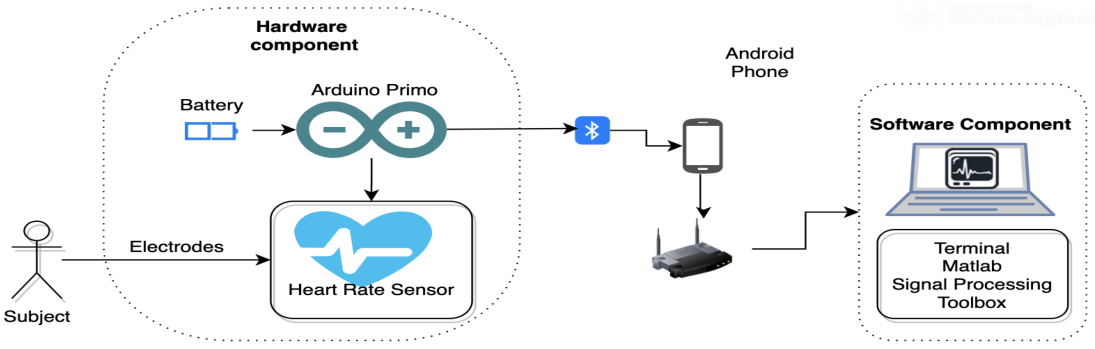


FIGURE 5.1: The System Architecture Diagram for a Data Acquisition System – Hardware and Software Components [135]

data acquisition system consisted of two components: A hardware component which was obtained from [135] and a software component as shown in Fig 5.1.

The hardware component consisted of a wearable belt with ECG sensor attached to a Arduino board. The wearable prototype was designed after taking several features in

account as explained in [25]. The device should have been able to record the signals continuously and should be adaptable to the experiment protocol which requires a lot of movement including falls. For this reason, a 3-lead ECG was opted as it is not only a standard for emergency purposes (e.g. in ambulance) but it also requires less wiring and hence provides patient compliance. The readings of a 3-lead ECG are comparable to a 12-lead ECG in our experimental set up. The design of the wearable device is discussed in detail in [25].

Out of many different types of falls, rolling-out-bed as a main and significant one was chosen for this study. Rolling out is defined as from lying, rolling out of bed and going to the floor [245]. The experimental setup consisted of a bed or a table. The sensor and Arduino gadget were worn by the subject. The ECG electrodes were tapped on the chest of the subject. Then the Bluetooth low energy (BLE) connection between the peripheral and central device was established.

5.4.1 Inclusion Criteria

The study was administered on 8 healthy subjects out of which 3 were females and 5 were males. The mean age of the subjects was 34.5. The following inclusion criteria were designed to enlist the volunteers for the HAR experiment:

- Age between 18 and 55 years
- no history of heart disease or hypertension
- no drugs or alcohol consumption before the experiment
- no physical constraints for falling (arthritis etc)

Our experiment was evaluated by the security officer of our university and a risk assessment was established with assistance from a medical doctor classifying our study as a HAR experiment in accordance with the ethical principles of the Declaration of Helsinki [16].

The heart of a human goes through various changes from birth to adulthood. ECG parameters vary for different age groups in general below 18. Human heart goes through a lot of physiological and anatomical changes from birth to adolescence. This causes some ECG features to differ significantly in adults as compared to children of different ages [63]. In order to correctly interpret the results with various age groups in children, one should have detailed knowledge of age dependant changes which is out of scope for our study. So that is the reason the age group chosen was between 18 and 55 for the experiment.

5.4.2 The HAR Experiment

The ECG was recorded in three positions: laying down, falling by rolling over from the bed and performing daily activities (such as walking and running). Fig. 5.2 shows the fall in three steps. The first on the extreme left is resting position. Next comes the fall initiation and it ends with the person laying on the ground without moving.

FIGURE 5.2: The Rollover Fall Process

Each experiment lasted about a total of 40-45 minutes consisting of multiple sub-readings of 30 seconds each. A total of three falls was recorded in a single reading. In each fall, the subject was in resting position as shown in the Fig. 5.2 for first 10 seconds. In the next 10 seconds, the volunteers would fall by rolling off the side of the table/bed and would lay on the ground until 30 seconds are completed. Then the subject would go up and repeat. Each reading consisted of 90 seconds containing a fall every 30 seconds. On average, 10 readings from each subject were collected.

For the resting ECG, the subject had to lay down on a flat surface without talking and moving to avoid the muscle noise. The log file consisting of 90 seconds for fall and 30 seconds for RESTING was sent to the central device via BLE. For daily activities, subjects performed some tasks like walking in a fast pace, sitting and standing etc for an average time of 10 to 15 seconds following no specific pre-determined protocol.

5.4.3 The Collected Data

The data collected consists of a total of 8 volunteers. Readings from two of the volunteers were too distorted to be used so they were not included in the next steps. It is important to note that since it is a preliminary study, more experiments are planned and recommended for future work including elder people and people with some pre-existing heart issues. An overall summary of the collected data is presented in Table 5.1. This table presents the data statistics as it was collected and saved in files. Due to limited

time with each volunteer, some of the subjects performed only fall experiments, others could do only resting readings and some could perform all three including daily activities. Table 5.2 presents the overview of multiple data instances that were obtained from those files.

TABLE 5.1: A Summary of the Collected Data

No.	Participant Id	Gender	Age	Experiment Duration	Total FALL Readings	Total RESTING Readings	Total DA Readings	Duration of each FALL	Duration of each RESTING	Duration of each DA
1	Sub1	M	35	120 min	5	13	5	90 s	90 s	30 s
2	Sub2	M	52	50 min	3	10	0	90 s	90 s	0
3	Sub3	F	30	50 min	6	0	0	90 s	0 s	0
4	Sub5	F	32	120 min	25	36	5	90 s	30 s	30 s
5	Sub6	M	40	45 min	8	0	0	90 s	0 s	0
6	Sub7	M	18	45 min	10	0	0	90 s	0 s	0

TABLE 5.2: Total Number of Samples Used for Training in Both Iterations

Reading Type	**Total Count (Phase I)**	**Total Count (Phase II)**
FALL	153	500
RESTING	104	474
DAILY ACTIVITIES	-	296
Total	257	1270

5.5 Our Methodology and Implementation Protocol

The following section outlines our proposed algorithm and its initial implementation.

5.5.1 Proposed Algorithm

The basic algorithm designed involves the following steps:

1. Pre-process
2. Filter
3. Calculate continuous wavelet transform and create scalograms
4. Fine tune and retrain a pre-trained CNN
5. CNN Classification

Algorithm 1: Detecting FALL with Raw ECG Signals

Input: A time series ECG raw data Ts
Output: The classified activity label l
$Ts \leftarrow ECG\_RAW\_VALUE\_EXTRACTION(Ts)$
$Ts \leftarrow FILTER(Ts)$
$WTS \leftarrow SCALOGRAM\_CREATION(Ts)$
$l \leftarrow CNN\_CLASSIFICATION(WTS)$
return l

$ECG\_RAW\_VALUE\_EXTRACTION(Ts)$ includes pre-processing raw ECG signals by performing an Analog to Digital Conversion (ADC), i.e. extracting the voltage from the raw data, observing the frequency domain and if required, interpolating the data to upsample to ensure uniform sampling frequency. $FILTER(Ts)$ applies elliptical filter to remove noisy artifacts from raw ECG signals Ts. After the signals have been filtered, wavelet transform is calculated for each signal. The absolute value of wavelet transform is used to create corresponding scalograms or scale-frequency images. The set of images is fed into a fine tuned pre-trained convolutional neural network. Algorithm 2 describes algorithm for using AlexNet for classification. The trained model is then verified using k-fold verification.

Algorithm 2: CNN_CLASSIFICATION

Input: A set of Frequency-scale Scalograms WTS
Output: The classified activity label l
$AlexNet \leftarrow RETRAIN\_FINE\_TUNED\_ALEXNET$
$l \leftarrow AlexNet(WTS)$
return l

5.5.2 ECG Signal Filtering

ECG is a low amplitude bio-signal that can be easily contaminated by several different types of artifacts and other bio-signals. A lot of sensor-related information or overhead has to be removed from the log file at the initial stages of filtration.

The sensor value we get is value from an analog to digital 14 bit converter (ADC). The following formula was used to find the corresponding voltage to the ADC value [217]:

$$Voltage = \frac{3.3V}{16385} \times ADCReading \tag{5.1}$$

Noises in ECG signal are composed of components at high–frequency and/or at low–frequency. Noises with high–frequency components include power line interface, muscle noise, and white Gaussian noise. Baseline wandering is a noise with low–frequency components. Minimizing baselines wander and power line interference is the first step in all electrocardiograph (ECG) signal processing [153]. Major noise artifacts lie in the following regions [59] :

- Muscle noise – 5-50 Hz
- External electrical noises – 50 or 60 Hz
- Respiration Noise – 0.12-0.5 Hz,

The frequency spectrum (the graph between frequency and amplitude of the signal) of the signals was initially analyzed to detect noises. A harmonic of about 1.5 Hz was observed in the spectrum. These harmonics are most probably due to the power line interface (PLI) because the frequency multipliers (harmonics) are PLI characteristics. A baseline wander is also seen in some of the readings (Fig. 5.3).

The goal of filtering is to remove maximum noise while still preserving the characteristics and properties of the signal. With a wide range of filtering techniques available choosing the right filter presents a challenge in itself. The next step after the identification of a type of artifact in our signal is to choose an appropriate filter. ECG signal filtering is a vital step for ECG signal processing and analysis.

Several different techniques exist in literature to remove noise from ECG signals. [66], [12] and [139] use wavelet transform to remove baseline drift and reducing overall noise. Similarly, [139] uses an ECG simulation to compare different techniques to remove baseline wander in order to preserve changes in ECG wave and found wavelet-based baseline cancellation the best method. However, it also recommended the Butterworth high-pass filter because of its computationally fast nature.

5.5.2.1 IIR versus FIR

The classification of transfer functions in time domain based on the length of its impulse response sequence leads to transfer functions known as infinite impulse response

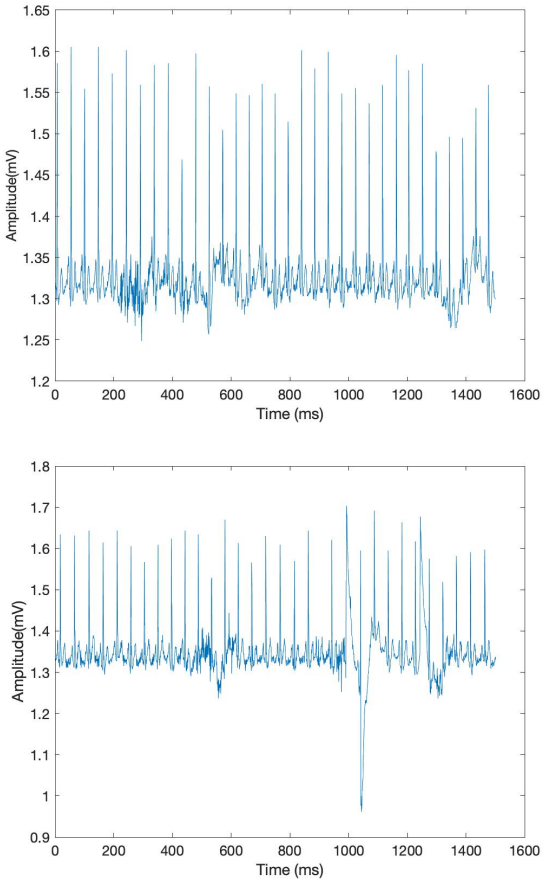


FIGURE 5.3: Examples of Baseline Wander.

and finite impulse response. The infinite impulse response or IIR filters have an impulse response that does not become exactly zero after a certain point, but continues indefinitely. Whereas in contrast, with the finite impulse response (FIR) the impulse response is finite in length [206].

One chooses between FIR and IIR filters depending majorly on the relative advantages of the two filter types. Since a challenge in our experimental set up is a lot of subject movement and implementation of ECG de-noising in combination with a smart mobile device (hence a limited resource environment), a careful balance between efficiency and accuracy has to be found. In [167] comparison between windows based FIR and IIR Butter-worth filters was observed and the IIR filter was shown to outperform FIR by acquiring a better computational efficiency with a minimal signal distortion. Similarly, in [216], the best trade-off between spectral density and average power was shown by IIR filter (Chebyshev Type II) as compared to its counterpart FIR filters. IIR filters have less computational complexity and hence require less computation power as compared to FIR filters. The memory requirement is also increased in the case of FIR filters hence IIR filters can be the better choice for removing baseline noises [216]. Wavelet analysis is

also used in recent works to denoise ECG data. [66] uses discrete wavelet transformation (DWT) to correct baseline wander and reduce noise. They estimate the baseline wander through coarse approximation in DWT and propose recommendations for the selection of wavelets and the maximum depth for decomposition level.

5.5.2.2 Elliptical Filter

To acquire a given level of performance, a much higher filter order is required in case of FIR as compared to IIR. This causes a greater delay in these filters for an equal performing IIR filter.

First of all, the minimum order of an elliptic filter which is required to meet a set of filter design specifications was determined and then same order was used to filter all the signals.

The elliptic filter was better fitted to the data but it not only removed the PLI in the signal, but baseline wander was considerably improved as well.

5.5.3 Implementation Protocol: Hardware And Software

The following section gives a detailed account of the implementation process for phase I. Now we have filtered ECG signals from the previous step and we want to use a pre-trained CNN, AlexNet, to differentiate between falling and resting ECG signals.

After filtration, each signal of 90 seconds duration was individually examined and divided in 3 readings in such a way that precisely one fall or at least more than 70% of the fall laid in each reading. This was done to increase the dataset as well as homogenize the data at one–time duration i.e. 30 seconds. At this point the collected signals for Daily Activities were considered a part of the class No-Fall.

Two models were trained on different hardware resources for phase I. One with a GeForce RTX 2080 Ti GPU with computing capability of 7.5 and another with a PC containing Intel(R) Core(TM) i5-7260U 2.20 GHz CPU.

5.5.4 Time-Frequency Representations

The scale-frequency representations were created using the algorithm 1, titled SCALOGRAM_CREATION(Ts). Each ECG signal had to be of the same length in order to compute a continuous wavelet transform (CWT) filter bank for all of them. It had to be done carefully so that no valuable signal, specifically fall signals, are lost.

In the next step, CWT filterbank for one of the signals was computed and using those coefficients, time-frequency representations called scalograms for rest of the data were created. Morse wavelet was used to calculate the wavelet transform. The scalograms were saved for later processing. Each representation was of 227*227*3 sized RGB image as it is the expected input format for AlexNet.

Fig. 5.4–5.6 provide a comparison between falling and non-falling scalograms. It can be seen that scalogram with a fall in them have some areas with high energy concentration in them in yellow color. On the contrary, the energy seems to be distributed evenly in non-fall ECG scalogram representing the cardiac cycle. Then those images were divided randomly into training and validation data with 80% training images and 20% testing images.

5.5.5 Phases of Our Implementation

The whole study was conducted (see Table 5.2) in two major phases as explained in following section. In phase I, the ECG signals obtained by our own experimental set up were used to create wavelet transforms and scalograms. The model trained in this phase had two classification classes: No-Fall which included all readings which did not have fall, and Fall which had signals with fall in them. Eventually, we fine tuned and retrained AlexNet to obtain a trained CNN model which gave 98.02% validation accuracy. (Phase II is described in section 5.5.6 below.)

5.5.5.1 Training The Network : Phase I

In phase I, a very small number of daily activities along with resting in no-fall scenarios were included. The daily activities included fast walking, sitting up from laying, sudden standing from sitting, picking up something from the ground while standing and stumbling while fast walking. The daily activities were performed in a domestic setting. Each activity duration ranged between 20 to 30 seconds.

These were also processed in the same manner as the previous readings in the earlier iteration. Two training sessions were done in this iteration, one with Intel i5, 1.3 GHz and another with the GPU.

5.5.5.2 Preparing and Training the Model

In transfer learning, a large dataset is used to train a primary network and the features learned from the training are either re-purposed or transferred to an another designated

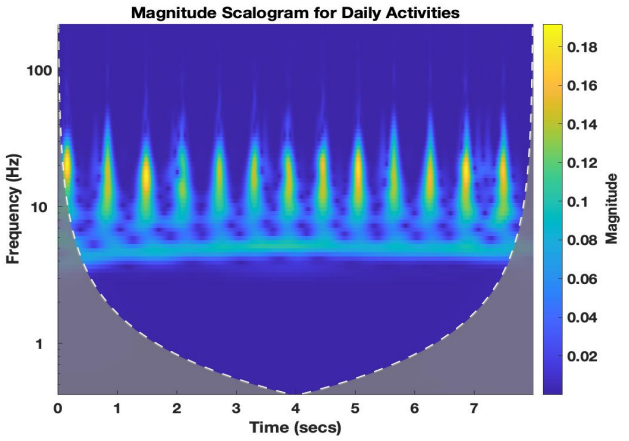


FIGURE 5.4: DAILY ACTIVITIES

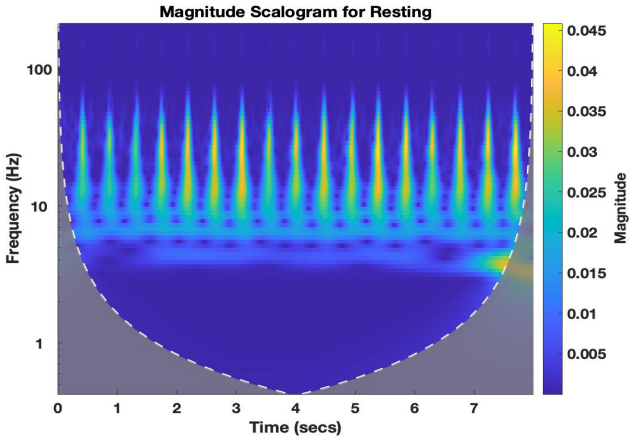


FIGURE 5.5: RESTING

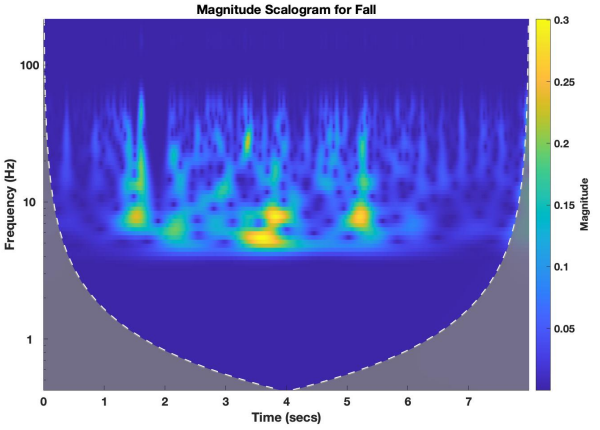


FIGURE 5.6: FALL

FIGURE 5.7: Scalograms with different classes of ECG signal in them

network to be trained on a specific dataset and task. It is usually very difficult to acquire a dataset of sufficient size to train an entire CNN from scratch so practically in most scenarios a CNN is pre-trained on a very large dataset and then used either as initialization or as a feature extractor for the required task. This process is called transfer training. This process works if the features are more generic and suitable to both primary and target tasks and not specific to the basic task [270].

In our case, since the dataset was small, the chances of over-fitting are a concern for using transfer learning. A pre-trained network, AlexNet, was used to train the model and the deeper layers of the network were fine-tuned. Fine-tuning is the closest solution to our problem because universal features like edges or curves are captured by a pre-trained network which is trained on a large and diverse dataset like ImageNet [62]. According to [224], despite the differences between natural images and medical images, the CNNs trained on the well annotated ImageNet can still be transferred to make medical image recognition tasks more effective. It is conjectured that by fine–tuning the transfer learning strategy yields the best results for performance in medical imaging [224] and our experiments confirm this conjecture.

5.5.5.3 Tuning The AlexNet

As part of any transfer-learning approach the AlexNet had to be fine-tuned to our dataset. The last three layers of AlexNet are configured for 1000 categories and since we have two classification classes, we fine-tuned these layers according to our classification problem. Layer 23 was set to be a fully connected layer of size equal to the number of our classification classes, 2. Layer 24 applies the Softmax and does not need to change. Layer 25 holds the name of the loss function used for training the network and the class labels [160]. So, layer 25 was set to be the classification output layer.

The solver algorithm used is 'Stochastic Gradient Descent with Momentum' (SGDM). The initial learning rate used for training was set to $10^{-}4$. The maximum number of epochs were 5 and the size of mini-batch to be used at each epoch was 15. The training was carried out in both a CPU and a GPU.

5.5.5.4 Explainability: Activation's of Different Layers In CNN

Each layer of the CNN produces a specific response to the original image called activations which can be viewed to investigate and learn further how the network learns and what features are adapted by the model to recognize falls. The activations of different layers can be viewed and one can discover which features are learned by the network by

comparing areas of activation with the original image. This can also help if we want to manually extract the features by looking at different layers and the features they highlighted. In Fig. 5.9, several tiles can be seen on the grid. Each one is the output of

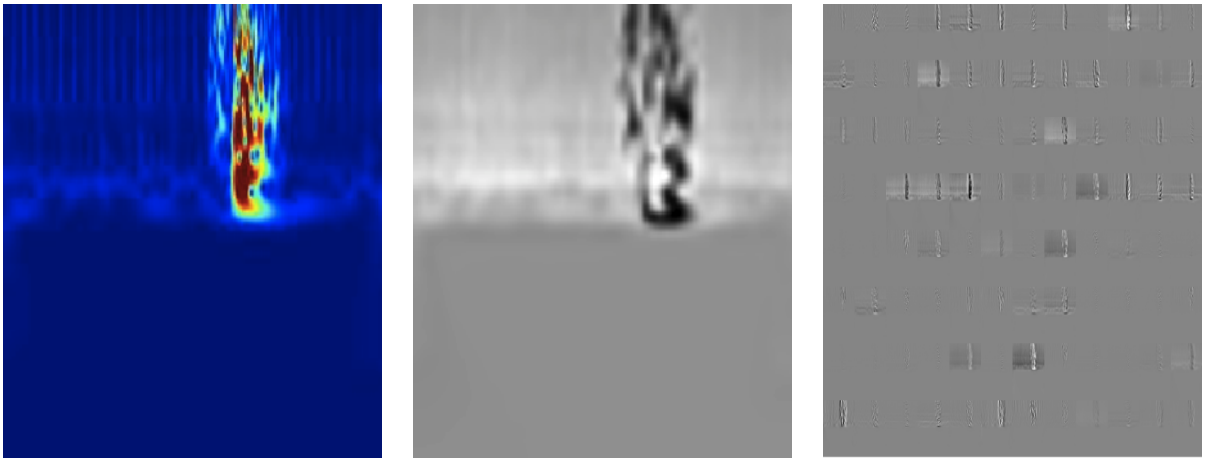


FIGURE 5.8: Scalogram of FALL ECG and its corresponding activation (Left to right): Figure (a): Scalogram with a distinct FALL. Figure (b): Strongest Activation Of the Image in Layer 5 (`conv5`). Figure (c): Activations in The `conv1` Layer

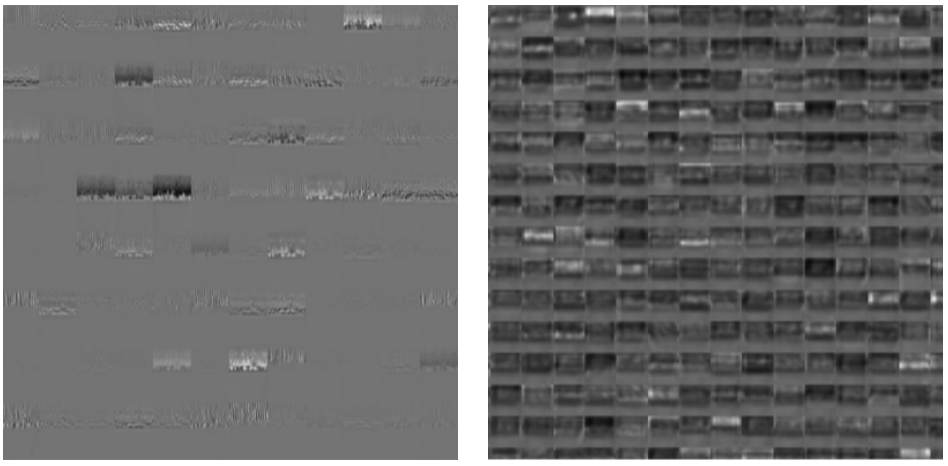


FIGURE 5.9: Scalogram Of RESTING ECG: Activation of earlier and deeper Layers : (Left to right): Figure (a): Layer 1. Figure (b): Layer 2

a channel in convolution layer 1 (`conv1`). Strong positive activation is represented by white pixels and similarly strong negative activations are shown by black pixels. A grey pixel represents moderate activation. The position of the pixels in the activation of a channel corresponds to the same position in the original image [160]. It can be noticed in Fig. 5.8, the network has started to learn about specific fall areas in its activation's on the first convolution layer. Though it was never explicitly told to learn about this specific fall pattern in scalogram, it has automatically outlined those features in a fall.

The feature highlighted by the strongest strongest activation in the convolution layer 5 (`conv5`) in Fig. 5.8 shows that the network correctly highlights the fall feature in a scalogram. Similar patterns can be observed in other scalograms containing falls.

Similarly, by looking at Fig. 5.9, it can be observed that activations of the deeper layer are more specific in features as compared to activations of the earlier layers which are more generic.

5.5.6 Extension of the Algorithm: Phase II

In the next phase, this study was enhanced by increasing datasets using time series augmentation technique called slicing and adding two different publicly available datasets, [124] and [84]. In this phase another pre-trained CNN called GoogLeNet [236] was trained along with AlexNet [133]. The training for this phase was carried out on a Macbook with 2,6 GHz 6-Core Intel(R) Core(TM) i7. This phase was also concluded with an accuracy of 98.44%.

5.5.6.1 Data Augmentation and its Challenges

In order to verify the effectiveness of the proposed algorithm, an additional class was added to our model called Daily Activities. This class encompasses the daily activities which we do in daily life like moving around, walking, sitting and jumping etc. Two publicly available datasets, [84] and [124], were added in the newly defined class. The dataset [84] contains ECG readings from 10 subjects (with mean age of 27, 1 female and 9 males). The signals were recorded performing four body movements – left and right arm up/down, sitting down and standing up and waist twist. Similarly the other database, [124], has ECG signals recorded from a healthy 25-year-old male performing different physical activities.

One of the main issues incorporating a new dataset to our existing dataset was the difference in sampling frequencies between the collected data and external data sources. We collected our data at a rate of 62 Hz whereas both external databases have data sampled at a rate of 500 Hz. In order to bring all the data to a common sampling frequency, our collected ECG dataset was interpolated at a rate of 5. Interpolation works better as compared to simple upsampling or resampling. In MATLAB function `upsample()`, zeroes are inserted between the corresponding values. Similarly in `resample()`, an anti-aliasing filter is applied after interpolation, which causes a drastic decrease in frequency amplitude. Hence, the function interpolation was used instead of `upsample()` and `resample()`.

In the previous phase, the dataset collected by us was limited and convolutional networks tend to over-fit with limited datasets. Hence, it was vital to augment the available data. Very few techniques exist in the field of time series augmentation specifically for deep neural networks [112]. We have used the method called windows slicing for augmenting the ECG signals as described in [55].

For a time series T, $T = \{T_1, ..., T_n\}$, window slicing is described as slicing the T into small snippets such that each snippet $S_{i:j} = \{t_i, t_{i+1}..., t_j\}$, where $1 \leq i \leq j \leq n$.

The slicing operation is described as follows [55]:

$$Slicing(T, s) = \{S_{1:s}, S_2 : s + 1, ..., S_{n-s+1:n}\} \tag{5.2}$$

In our case, the length of each slice was 4000 and the TS was sliced at every 1000^{th} interval. So each time series signal of length n, gave us $\{n - 4000/1000\}$ signals. All the time series generated by Slicing(T,s) will have the same label as that of T.

5.5.6.2 Tuning the GoogLeNet

Along with AlexNet, GoogLeNet was trained in phase II to obtain a comparative analysis of transfer learning. GoogLeNet, like AlexNet, is also a pre-trained convolutional neural networks. Introduced in [236], GoogLeNet was a winner of ILSVRC 2014 classification and detection challenges. Since the pre-trained models are already trained on a large image set, fine tuning them would provide a simpler and efficient solution to the issue of data shortage for deep learning problem specifically in medical domain. Now our model would have three classification classes namely Fall, Resting and Daily Activities.

GoogLeNet has 144 layers as compared to 25 layers in AlexNet. It expects an input image of size 224*224*3. GoogLeNet introduced a novel feature called inception. It replaces the fully connected architecture with sparse architecture inside the network. The size of convolution filter is fixed in earlier CNN models like AlexNet, and VGGNet. However, now multiple convolution filters of varying size and a maxpooling is done altogether for the previous layer, and the result is stacked together again at output. This not only leads to extraction of different features but it is also computationally efficient [236].

Fine tuning GoogLeNet requires redefining some layers of the network. First of all, final drop out layer is replaced with a probability of 0.6 instead of 0.5. The last two layers, 'loss3-classifier' and 'output' are replaced with layers which are in accordance with our classification scheme. The layer 'loss3-classifier' is replaced to classify three classes instead of 1000. Finally, the last layer 'classification' is replaced with a new layer

with out any classification labels. The output classes would be set accordingly during the training time.

5.5.6.3 Transfer Learning to the Rescue: GoogLeNet and AlexNet

Even after applying data augmentation and adding external database, our database reaches to a total of 1273, see Table. 5.2, which would be not suffice to train any CNN from scratch for the risk of over-fitting. This kind of situation is far too common in medical field due to strict data protection and privacy laws and unavailability of domain specific data. Transfer learning can be the answer to various questions in this regard. A pre-trained model is already trained on a huge dataset; in case of most CNNs, IMAGENET [62]. They have their inner layers already generalized enough to extract the relevant features from a new domain. So training on a new domain can be achieved even on a small or mid size dataset. We just have to re-train outer most layers to readjust their weights. Hence, it is also resource friendly.

A number of models, both AlexNet and GoogLeNet, were trained. The different parameters, such as initial learning rate (ILR), training algorithm, percentage of validation-training-testing data were varied in order to verify the generalization of the model and proposed algorithm. Training algorithms called Stochastic gradient descent with momentum (SGDM) and Root Mean Square Propagation (RMSprop) were used with similar configurations to compare the outcome. SGDM is a stochastic approximation method initially proposed in [198] with a momentum added to it [189]. RMSprop was initially proposed and explained by Geoff Hinton in an online course [83] and is an unpublished optimization algorithm. The summary of the trained models and the results are shown in Table 5.4 and Table 5.5. Out of many trained models, Model 2 from Table 5.5 was selected as the most suitable trained model. Model 5 and 6 were not chosen despite of a perfect 100% validation accuracy because they might be over-fitted. Similarly, Model 12 was not chosen because the percentage of validation data is greater in Model 2 (0.8-0.1-0.1) than in Model 12 (0.6-0.2-0.2). A confusion matrix for testing data of Model 2 is shown in Table 5.3. The last row and column in the confusion matrix indicate the total number of correct predictions expressed in percentage. For example, if we look at the first column, 55 instances of Daily Activities are correctly predicted as Daily Activities and 5 of them were incorrectly predicted as Fall by the model. This amounts to a total correct prediction percentage of 91.7% for daily activities. Similarly, out of a data of 255 instances, 246 were correctly predicted which amounts for a total accuracy of 96.5%.

A graphical summary of the training result is shown in Fig. 5.10.

TABLE 5.3: Confusion Matrix for the Selected Trained Model

		Actual			
		Daily Activities	Fall	Resting	Total (in percentage)
Predicted	Daily Activities	55	1	0	98.2%
	Fall	5	96	0	95.0%
	Resting	0	3	95	96.9%
	Total (in percentage)	91.7%	96.0%	100%	96.5%

TABLE 5.4: Summary of GoogLeNet fine-tuned and retrained for Fall Detection using ECG Signals with different Parameters

No.	Algorithm	Train.-Val.-Testing Ratio	ILR	Val. Accuracy	Testing Accuracy
1	SGDM	0.8-0.1-0.1	1e-4	98.44%	95.3%
2	SGDM	0.8-0.1-0.1	1e-5	95.31%	93.7%
3	SGDM	0.8-0.1-0.1	1e-6	85.94%	86.6%
4	RMSprop	0.8-0.1-0.1	1e-2	39.06%	39.4%
5	RMSprop	0.8-0.1-0-1	1e-4	98.44%	98.4%
6	RMSprop	0.8-0.1-0-1	1e-5	100%	97.6%
7	RMSprop	0.8-0.1-0-1	1e-6	96.09%	92.1%
8	RMSprop	0.8-0.1-0-1	1e-9	37.50%	37.0%

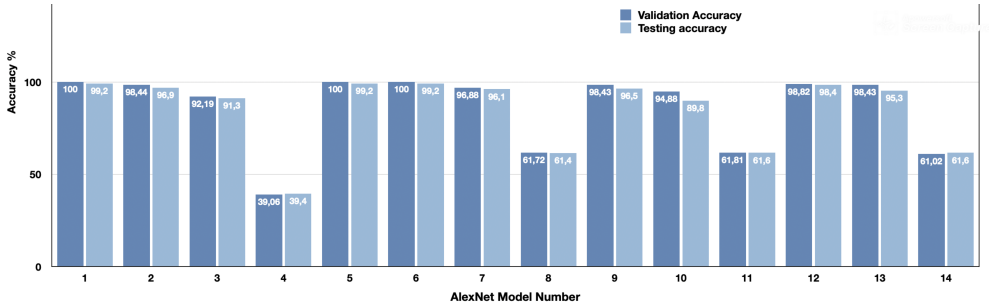


FIGURE 5.10: A Graphical Summary of the Training Results for AlexNet Model

5.5.6.4 k-fold Verification

A k-fold verification was applied on model number 2 from Table 5.5, as a measure to estimate the generalization of the model. Keeping the size of dataset in mind, k was set to 3. The dataset was divided in three equal parts. At each split, one set was used as the training data and the trained model was tested on both validation and testing data to obtain validation and testing accuracy respectively. An average validation accuracy of 97.69% and a testing average accuracy of 97.37% was achieved. An overview of the process is depicted in Table 5.6.

TABLE 5.5: Summary of AlexNet fine-tuned and retrained for Fall Detection using ECG Signals with different Parameters

No.	Algorithm	Train-Val.-Test Ratio	ILR	Val. Accuracy	Testing Accuracy
1	SGDM	0.8-0.1-0.1	1e-4	100%	99.2%
2	SGDM	0.8-0.1-0.1	1e-5	98.44%	96.9%
3	SGDM	0.8-0.1-0.1	1e-6	92.19%	91.3%
4	RMSprop	0.8-0.1-0.1	1e-2	39.06%	39.4%
5	RMSprop	0.8-0.1-0-1	1e-4	100%	99.2%
6	RMSprop	0.8-0.1-0-1	1e-5	100%	99.2%
7	RMSprop	0.8-0.1-0-1	1e-6	96.88%	96.1%
8	RMSprop	0.8-0.1-0-1	1e-9	61.72%	61.4%
9	SGDM	0.6-0.2-0.2	1e-5	98.43%	96.5%
10	SGDM	0.6-0.2-0.2	1e-6	94.88%	89.8%
11	SGDM	0.6-0.2-0.1	1e-9	61.81%	61.6%
12	RMSprop	0.6-0.2-0-2	1e-5	98.82%	98.4%
13	RMSprop	0.6-0.2-0-2	1e-6	98.43%	95.3%
14	RMSprop	0.6-0.2-0-2	1e-9	61.02%	61.6%

TABLE 5.6: An Overview of K-fold Verification

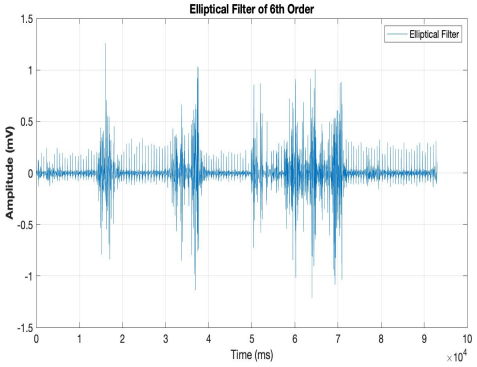
	Fold 1	Fold 2	Fold 3	Val Accuracy	Test Accuracy
Split 1	**Fold 1**	Fold 2	Fold 3	97.65%	98.2%
Split 2	Fold 1	**Fold 2**	Fold 3	98.04%	97.4%
Split 3	Fold 1	Fold 2	**Fold 3**	97.39%	96.5%
Training data, Testing data			Average Accuracy	97.69%	97.36 %

5.6 Analysis of Results

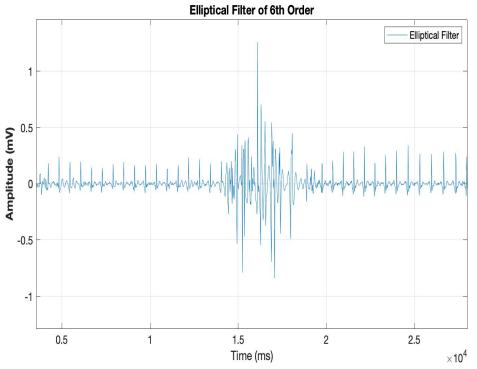
In this section analysis of different results is done to obtain a better understanding of the process.

5.6.1 Analysis of FALL Vs NO-FALL ECG Signals

Fig. 5.11(a) shows a fall ECG with three falls in it. It can be observed that the amplitude has increased significantly in three areas. This can be contributed to either noise due to hit during the fall or due to an actual increase in cardiac activity. The increase lasts for about 1.6 seconds and then immediately goes back to normal. This time duration coincides with the time taken for the body to touch the ground. This increase in amplitude is of special interest to us. When zoomed in, as shown in Fig. 5.11(b) the increase goes to an amplitude of 1 mV maximum and is followed by normal cardiac activity as we can see the normal QRS complex after the fall peaks. The other smaller peaks in heart rate in other areas are due to the movements of the subjects from the ground back to the table. The same pattern is repeated with small differences in all other recorded falls.



(a) An ECG Signal With Three Falls



(b) The Fall Pattern Immediately Followed By Normal Heart Activity

FIGURE 5.11: Falls in ECG signals in Time Domain

Fig. 5.12 is a closer look at another fall from the ECG signal in Fig. 5.11(a). This fall is also of the same duration (1.6 seconds) as of in Fig. 5.11(b).

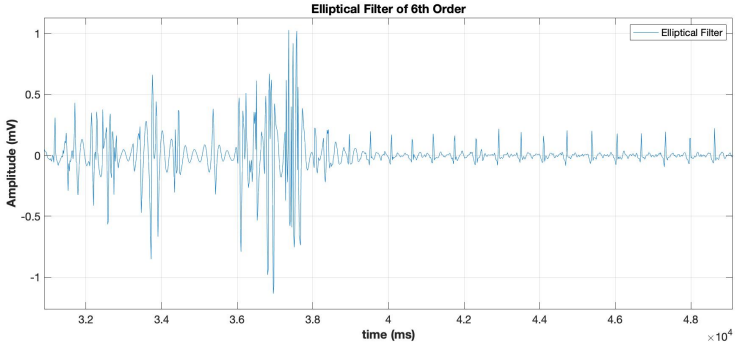
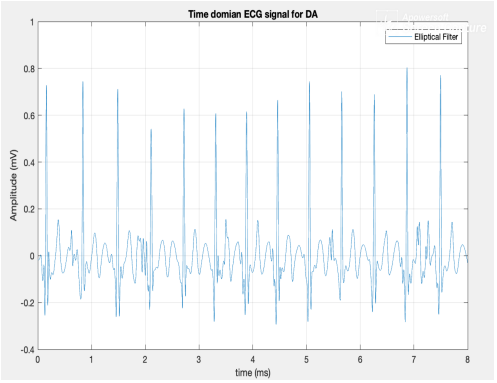
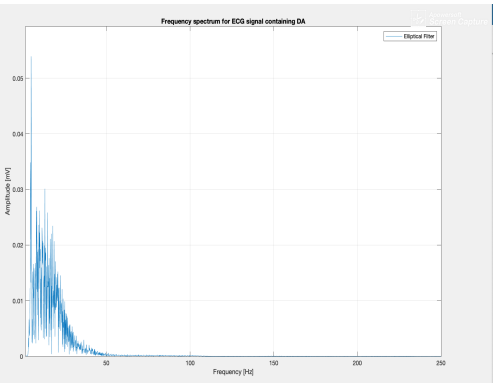


FIGURE 5.12: A Closer Look at The FALL Activity in an ECG Signal

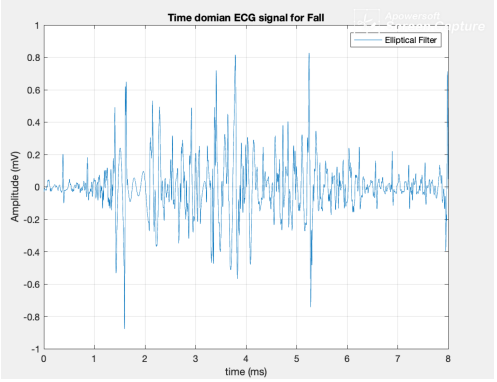
Fig. 5.13 gives a side-by-side view of the time domain signals and the corresponding frequency domain of the signals for all three classes of the model. A distinct frequency spectrum can be seen in Fig. 5.13 for FALL, RESTING and DAILY ACTIVITIES. The fall is visible in time domain and is also characterized by a spike in frequency domain. The



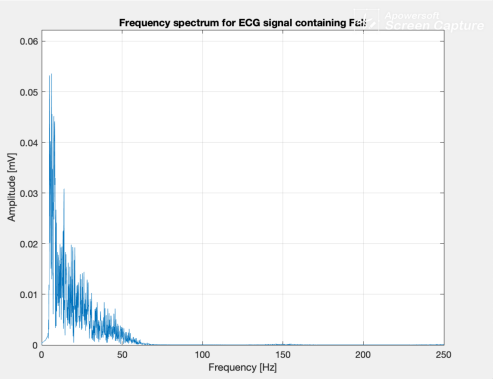
(a) Time Domain ECG Signal for Daily Activities



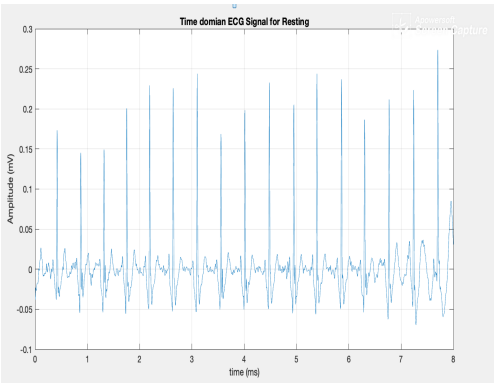
(b) Frequency Domain ECG Signal for Daily Activities



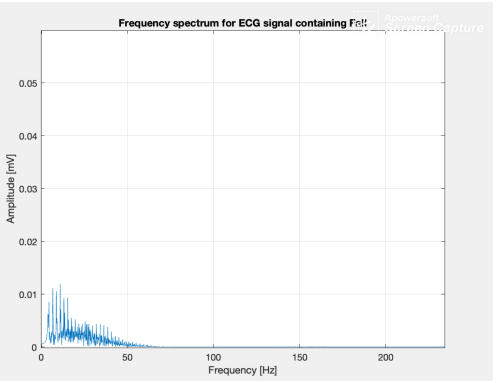
(c) Time Domain ECG Signal for Fall



(d) Frequency Domain ECG Signal for Fall



(e) Time domain ECG Signal for Resting



(f) Time Domain ECG Signal for Resting

Figure 5.13: A Comparison of different ECG Time Domain Signals with their Corresponding Frequency Spectrum

daily activities ECG graph shows considerable noise even after filtration. This is due to the constant movement of the subjects resulting in baseline wander.

5.6.2 Analysis of Scalograms

The scalogram is obtained by plotting the absolute value of CWT of a signal as a function of time and frequency. They are helpful in visualization of varying events in a signal. Scaling and shifting are used on a prototype wavelet to highlight the transient changes in the signal. We used 'Morse' wavelet as a prototype wavelet because its form is very similar to the ECG shape. The fall scalograms shown in Fig. 5.7 have distinct localized high energy areas whereas the resting scalograms have high energy distributed all over the diagram. Now the comparison between DAILY ACTIVITIES scalogram with fall scalogram is interesting. The scalogram of daily activity is shown in Fig. 5.4. The high energy is distributed almost uniformly all over the representation. If we compare the DAILY ACTIVITIES to the fall scalogram (Fig. 5.6), it can be clearly seen that noise or high energy due to movements can be seen throughout the daily activities in contrast to a localized high energy area in fall scalogram. Also, the magnitude of the the energy is different for RESTING and DAILY ACTIVITIES scalograms.

5.7 Discussion of the Research Question

It is evident from the extension of phase I, that even after adding an additional class, the model is trained well and is generalized enough after varying the parameters. With the addition of an external database, the model has learned well the classification task.

This answers our basic research question which was: "Can a fall be detected using an ECG signal?". It is shown that not only can we detect fall in ECG signals but we were able to do it with an accuracy of greater than 90%. As for changing the parameters, it is noticeable from Table 5.4 and Table 5.5, how initial learning rate plays the most significant role in determining the validation accuracy. Varying the parameters like the algorithm and the dataset ratio does not make significant change in the validation or testing accuracy. It can be also seen from the tables that by keeping all other parameters constant, the choice of neural network does not make any significant difference in the results.

Hence, a roll over fall has some characteristics imprinted in electrocardiogram signals, which are detectable in time-scale domain (scalograms). That is why the models have learned to distinguish a fall activity from a no fall activity A visible difference with

respect to energy distribution can be seen in different activities. These characteristics are distinguishable from an ECG of a resting person or a person performing daily activities. More investigation still needs to be done in the frequency spectrum of ECG signal to learn more about the pattern specific to the fall and other activities.

Our algorithm has obtained an accuracy of more than 97%, greater than that achieved in [123]. Although in [123], different movements were classified using traditional deep learning techniques and fall was not among those activities.

However, our experimental setup is not free of limitations. The focus of this paper has been on determining proof of concept in laboratory and by no means this is a deployment ready prototype. The initial experimental population is considerably small. That was compensated in the second phase by adding two different external datasets. Both datasets not only increased the instances numerically but also brought variation with respect to number of subjects, age and type of noise and signals. Both datasets had obtained signals using different devices. All these factors strengthens our proof of concept. We are also bounded by the obvious limitations of the controlled experimental set up. For example, the subjects have to wear the safety gadgets (helmet, knee caps etc) in order to comply with the safety regulations. This might cause an onset of anxiety and nervousness in volunteers, which in turn may effect the heart rate signals. But these circumstances are unavoidable and we have to take them as experimental conditions.

5.8 Conclusion and Future Work

It is shown clearly that ECG signals can be utilized alone for not only distinguishing different activities but also fall from no-fall activities by using a combination of wavelet transform and transfer learning. The transfer learning models have learned as efficiently and accurately as any other convolutional neural networks trained from scratch. In future, we plan to further analyze the transfer learning approach for time series classification by directly training the ECG signals using long short-term memory (LSTM) networks. Also, a detailed frequency spectrum analysis of signals can be performed to extract the features specifically present in different activities. Further experiments are planned to be carried out to include different kind of falls, activities and circumstances. As discussed in the previous section, because of the limited dataset initially acquired, we plan to carry out our experiments on a larger and more diverse (in terms of age and gender) group of people to improve this study. Another extension we would like to do is to test our proposed algorithm with state of the art pre-trained networks like EfficientNet [239] etc.

Chapter 6

Towards Automated Feature Extraction

> "If you want to find the secrets of the universe, think in terms of energy, frequency and vibration." – Nikola Tesla

This chapter is mainly based on the worked published in [37]. It is also not written in a 'verbum pro verbo' from the publication and has been adapted to the flow of the dissertation. The work focuses on extracting the ECG features automatically to replace the main pre-processing step in the previous work.

Many recent studies, which focused on the automatic classification of electrocardiogram (ECG) signals using deep learning (DL) methods, rely on existing complex DL methods, such as transfer learning or providing the models with carefully designed extracted features based on domain knowledge. A common assumption is that the deeper and more complex the DL model is, the better it learns. In this chapter, we propose two different DL models for automatic feature extraction from ECG signals for classification tasks: A CNN-LSTM hybrid model and an attention/transformer-based model with wavelet transform for the dimensional embedding. Both of the models extract the features from time series at the initial layers of the neural networks and can obtain performance at least equal to, if not greater than, many contemporary deep neural networks. To validate our hypothesis, we used three publicly available data-sets to evaluate the proposed models. Our model achieved a benchmark accuracy of 99.92% for fall detection and 99.93% for the PTB database for myocardial infarction versus normal heartbeat classification.

According to [47], in the United States of America alone, the leading cause of death for men and women irrespective of the racial and ethnic groups is heart disease. Hence, a

timely and accurate diagnosis of the heart conditions is of vital importance. An Electrocardiogram (ECG) is a well-grounded method used for measuring and evaluating the performance of the cardiovascular system. Several techniques exist in both literature and practice to evaluate the ECG signals in different manners. It is one of the most important parameters that indicate a person's physiological well-being and is extensively used to evaluate the cardiac situation of the patients. It has been widely used for different purposes such as to get an overview of the health of a human heart, for bio-metric purposes, and for fall detection and prevention as described in [238]. ECG is a non-invasive method for evaluating the health of the human cardiovascular system. It can detect many heart diseases such as atrial fibrillation, myocardial infarction, AV block, and ventricular tachycardia, etc. It provides an insight into the central nervous system, particularly the autonomic nervous system. Many of the automatic classification techniques using deep learning for ECG use either very deep neural networks or a pre-trained neural network that require either the weights set up to a configuration after being trained on an immense amount of similar data sets. Another approach is to pre-process the data sets by applying some filtration or feature extraction which is based on data domain knowledge and then fed into a neural network to train this. All of the above–mentioned steps involve an explicit understanding of the domain and the pre-process itself. [121] overviews the many ECG feature extraction techniques present in the literature.

Our work is motivated by the desire to design novel and simple models that avoid *any* feature selection and complex data pre-processing which necessitates domain knowledge, while on the other hand requiring *less* computation power but achieving at least state-of-the-art accuracy. In this paper we propose two architectures to achieve these goals including out-performance of benchmarks and analysis of the statistical evidence of our claims.

6.1 Motivation

As discussed in 4.4, a normal ECG consists of five major deflections called P, Q, R, S, and T waves, which constitute a single cardiac rhythm as shown in Fig. 4.3. The P wave lasts about 0.08 s and is the smallest, followed by the large QRS complex which lasts between 0.08 s and 0.10 s. The end of the cardiac cycle is marked by a T wave that lasts approximately 0.16 s (See Table 4.1. A single waveform varies depending on the size of the heart and the conductive properties of the body which in turn gives the waveform a unique pattern per person [113].

The disruption of blood flow to the muscle layer of the heart causes a cardiovascular condition called a myocardial infarction (MI). This disruption is mostly due to the build-up

of the plaques in the arteries which result in reduced blood flow to that part of the heart muscle. MI is called a silent heart attack because the patient is not aware of the condition unless they suffer from a heart attack. An early diagnosis of MI is therefore of vital importance as it would help the patients to get timely treatment hence preventing the high percentage of mortality associated with it. Due to the small amplitude (millivolts), the manual interpretation of ECG signals is time-consuming and prone to errors. This limitation can be mitigated by an automatic diagnosis of heart conditions based on the signals. Our study aims to work towards automation of the cardiovascular disease diagnosis from ECG signals.

In this study, we propose two methods to automatically extract features from a time series and then feed those features into another deep learning model for classification. First, a hybrid model for multiple ECG classification tasks is proposed as an alternative to many complex models that require many pre-processing steps before the actual training. We experimented with a robust hybrid deep learning model for the ECG classification tasks, which proved to outperform many state-of-the-art complex models and achieve similar or even better accuracy with no pre-processing steps. The CNN placed in front of a LSTM also known as CNN-LSTM, has recently been used for multiple classification tasks; however, its use for ECG classification has not been systematically explored. The CNN model first searches for the features in high-dimensional input data and then after converting it into one-dimensional data, it is fed as an input to the LSTM model. The role of a CNN in this context is to act as an automatic feature extractor. Secondly, a novel attention/transformer model using wavelets for dimensional embedding is introduced to improve the efficiency of the classification process. As it has less trainable parameters than CNN-LSTM it has advantages in terms of (training) performance as shown in Table 6.8. As a bonus, we also evaluated both models also on the data for fall detection.

6.2 Related Work and Our Contribution

Several recent studies have focused on automatic ECG classification. Among the several different techniques present for ECG classification, deep learning has gained popularity in recent times. This is mainly owing to its automatic feature learning and the availability of large public data sets. Many deep learning techniques use feature extraction as an essential pre-processing step before feeding the data to the neural network. The most common feature extraction techniques for ECG classification are continuous wavelet transform (CWT), discrete cosine transform (DCT) [125], Pan-Tompkins algorithm [208] and discrete wavelet transform (DWT) [57]. One of the major disadvantages of using wavelet transform as a feature extractor is that the complexity of the process increases

with the increase in decomposition level. All feature extraction processes require some domain knowledge in order to efficiently extract relevant features from the data. Therefore, we aim to explore the research question of whether a similar state-of-the art result can be achieved with no pre-processing and with a simpler model architecture in an efficient manner in terms of resources and computation.

The contributions to this study are twofold: First, we introduce a CNN-LSTM architecture that surpasses many complex and pre-trained models that have been optimized for single data sets on multiple data sets at the same time. Second, to further optimize the automatic feature extraction, we introduce a novel embedding technique for an attention/transformer encoder architecture that uses discrete wavelet transform to extract features from the ECG time series and feeds them to the attention mechanism. In addition, we provide statistical evidence for the significance of the performance figures reported by the two models proposed by us.

In the following sub-sections, we present the state of the art in the related work and highlight our contributions.

6.2.1 CNN-LSTM Architectures

Jambukia et al. (2015) [113] presented an overview of the ECG classification of different types of arrhythmias. Another current review on deep learning methods for ECG arrhythmia classification [71] deduced that among the many deep learning models, CNNs and LSTMs were among the most effective for learning arrhythmia in ECG classification tasks. The use of the CNN-LSTM architecture for classification is not entirely novel. Socher et al. [226] proposed a model for 3d object classification that combines a CNN with an RNN. They concluded that the CNN provides the translation variance for lower-level features whereas RNNs can learn the interactions and compositional features in the data. Zheng et al [277], transformed the data acquired by a three-axis accelerometer into an image format and then used a CNN with three convolution layers to classify human activities. XIA et al. (2020), [263] used CNN after a LSTM layer to classify human activity recognition (HAR) with an accuracy of 95.85%. Ordóñez et al. (2016),[177] proposed an activity recognition classifier that combines a deep CNN and dense layers. In [269] the authors proposed a 1-D CNN for the classification of cardiac arrhythmia, and in [96], a 34-layer convolutional neural network is used for classification of cardiac arrhythmia exceeding the performance of board-certified cardiologists. However, few studies have focused on hybrid CNN-LSTM models for ECG classification. Studies like [118],[231],[252], and [254] have implemented CNNs and their variants for ECG classifications. [202] used RNNs to classify the ECG signals. The use of LSTM-based approaches

is also beneficial for other cardiac signal analyses. [81] construct a bidirectional LSTM for the analysis of blood flow dynamics from static CT angiographic images. In [158] a restricted Boltzmann machine and deep belief networks were used for detection of ventricular and supraventricular heartbeats using single-lead ECGs. For a general overview of deep-learning techniques in cardiovascular image analysis, see the survey [142].

In our study, we not only performed multiple classifications with CNN-LSTM model for ECG but also worked with three different ECG data sets including data for fall detection to present a proof of concept that CNN placed in front of LSTM surpasses many complex and pre-trained models.

6.2.2 Attention and Transformer Architectures

The seminal paper by Vaswani et al. “Attention is All you Need" [249] has triggered an enormous number of successful applications of attention mechanisms and transformer architectures in deep learning.

The main idea behind attention-based transformer architectures is to replace the recurrence mechanisms used in LSTMs and the convolutions used in convolution networks to extract features entirely using an alternative so-called self-attention mechanism. This mechanism is shown in eq. (6.1) and computes the correlation between the input values among each other and can be interpreted as an associative memory using ideas from statistical physics, see [194]. Replacing the (serial) recurrence mechanism with the standard matrix algebra of the (self-) attention mechanism has a number of advantages for parallelization capabilities and the performance of classification tasks.

However, the vast majority of research has been and still is focused on the natural language processing (NLP) domain. Little research has been carried out on the application of attention-based architectures in other domains, such as time series analysis. One of the first papers in this regard is LSTNet by Lai et al. [136], where the authors introduced long- and short-term time-series networks (LSTNet) using the convolutional neural networks and recurrent neural networks to extract short-term local dependency patterns and to discover long-term patterns for time series trends. Shih et al. [223] applied an attention mechanism to multivariate time series data in three medical domains. Song et al.[227] have applied attention models to clinical time series analysis. A systematic and comprehensive analysis and study of utilizing attention mechanisms, however, in the time-series domain is still required. The application of transformers in the domain of ECG classification can be found in Chapter 8, Section 8.3.1.

One of the shortcomings of the self-attention mechanism preventing its application for e.g., time-series is the $\mathcal{O}(n^2)$ complexity with regards to the length of the input vector, i.e., the length of the time series in our case. To address this problem LinFormer has been introduced by Wang et al. in [253]. Linformer is the first theoretically proven linear-time transformer architecture and henceforth might be suitable also for long time series. The linear scaling is achieved by discovering that self-attention is low rank, and henceforth projecting information on a low rank constant sub-dimensions achieves to decouple from the $\mathcal{O}(n^2)$ scaling. Recently Rabe and Staats [190] have proposed an algorithmic solution to at least reduce the memory (but not the time) complexity from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$.

In this paper, we propose a novel attention architecture using projection on discrete wavelet components as a means to address the $\mathcal{O}(n^2)$ problem and for dimensional embedding. Moreover, the results show that using this technique, *attention-only* architecture is on par with or even outperforms more complex models and has several additional advantages such as e.g., better run-time performance.

6.3 Algorithms

This section provides a brief overview of the algorithms and the technologies that were used during the course of this study and also presents the state of the art in the respective technologies.

6.3.1 CNN-LSTM Model

We define some basic terms related to the convolutional neural network and LSTM for clarity in the following section.

6.3.1.1 CNNs and LSTMs

Convolutional neural networks, introduced as LeNet in 1989 by LeCunn, have revolutionized the field of image recognition and are among the most prominently used deep neural networks. They were named after the linear matrix operation called convolution. Since convolution is a linear operation, the convolution layer is often followed by a non-linear layer. Although introduced earlier, it gained popularity after its application as the first deep neural network applied for object recognition in the ImageNet Large Scale Visual Recognition Competition (ILSVRC) in 2012. AlexNet was proven to excel on the largest computer vision data set as compared to contemporary methods. Recently,

[122] presented a state-of-the-art review of the recent deep CNNs architectures. The individual CNN components were explained in [6] in a structured way. The most common architectures of CNNs include an input layer, a convolution layer followed by a pooling layer, a drop-out layer, and a fully connected layer followed by an output layer. The number of layers and their layout can change depending on different problem sets. The convolution operation[1] itself is given by:

$$\begin{aligned} V_{i,j} = X * W_{i,j} + b &= \sum_{L} X^{L} * W^{L}_{i,j} + b \\ &= \sum_{L} \sum_{k,l} X^{L}_{kl} * W^{L}_{i+k,j+l} + b \end{aligned}$$

where X or X^L resp. denote the L-th input matrix. W is the convolution kernel matrix, b is the bias, and $V_{i,j}$ is the output matrix after convolution.

CNN's are known for their excellent feature extraction capability. One of the most salient features of CNN is its translation invariance. Therefore, it can extract features irrespective of the spatial context. Though it has proven to be beneficial in image recognition, its application and usefulness in time series are yet to be fully exploited. Cases, where the historical context is relevant for classification, would not work well with CNN alone, as it does not carry any information about the history of the time series. The CNNs initially extract the local features in the sub-regions of the time series and then the information is merged in later stages to detect the higher order features. We applied 1D convolution to the time series using both univariate and multivariate data sets. The ECG Human activity recognition (HAR) data set and PTB diagnostic data set contained one feature each, so the 2D convolutional operation would not be suitable as it will incorrectly convolve across multiple time series. Long short-term memory (LSTM) networks–a variation of recurrent neural networks (RNNs)–were introduced by Hochreiter [101] in 1997. They tend to present a solution to the common problem associated with RNNs called vanishing and exploding gradients. In principle, classical RNNs can keep track of long-term dependencies in the sequences. However, in practice, during the backpropagation phase of training, these long-term gradients either vanish or explode owing to the successive multiplicative operations. An LSTM consists of a chained loop structure. Each LSTM unit is made up of an input gate, an output gate and a forget gate. The LSTMs keep the long-term memory by maintaining a cell state that sustains a part of the information from earlier states by forgetting and/or applying increment operations on the previous states. Adding a CNN in front of an LSTM helps to feed the LSTM the features from CNN which were extracted from the time series.

[1] in the two dimensional case–the one-dimensional case is analogous

6.3.1.2 CNN-LSTM Architecture and Algorithm

Fig. 6.1 and Fig. 6.2 provide a more graphical overview of our model. Initially (1*N) time series with N time stamps are convolved with k filters each of size M*1. Subsequently, the k feature maps each of size (N-M)+1 time stamps are generated which are passed through a dropout layer followed by the max pooling layer and later fed into the LSTM layer where the encoded extracted features are fed into it from the CNN. The LSTM unit is followed by a fully connected or dense layer that applies softmax as an output function to classify the input time series into one of the output classes.

```
==========================================================================
Layer (type:depth-idx)                                  Param #
==========================================================================
CNN_LSTM                                                --
├─CNN: 1-1                                              --
│    └─ReLU: 2-1                                        --
│    └─Sequential: 2-2                                  --
│    │    └─Conv1d: 3-1                                 561
│    │    └─BatchNorm1d: 3-2                            374
│    │    └─ReLU: 3-3                                   --
│    └─Sequential: 2-3                                  --
│    │    └─Conv1d: 3-4                                 24,000
│    │    └─BatchNorm1d: 3-5                            128
│    │    └─ReLU: 3-6                                   --
│    │    └─MaxPool1d: 3-7                              --
│    └─Sequential: 2-4                                  --
│    │    └─Conv1d: 3-8                                 8,256
│    │    └─BatchNorm1d: 3-9                            128
│    │    └─ReLU: 3-10                                  --
│    │    └─MaxPool1d: 3-11                             --
│    └─Sequential: 2-5                                  --
│    │    └─Conv1d: 3-12                                8,256
│    │    └─BatchNorm1d: 3-13                           128
│    │    └─ReLU: 3-14                                  --
│    │    └─MaxPool1d: 3-15                             --
│    └─Sequential: 2-6                                  --
│    │    └─Conv1d: 3-16                                8,256
│    │    └─BatchNorm1d: 3-17                           128
│    │    └─ReLU: 3-18                                  --
│    │    └─MaxPool1d: 3-19                             --
│    └─Sequential: 2-7                                  --
│    │    └─Conv1d: 3-20                                8,256
│    │    └─BatchNorm1d: 3-21                           128
│    │    └─ReLU: 3-22                                  --
│    │    └─MaxPool1d: 3-23                             --
│    └─Sequential: 2-8                                  --
│    │    └─Conv1d: 3-24                                16,512
│    │    └─BatchNorm1d: 3-25                           256
│    │    └─ReLU: 3-26                                  --
│    │    └─Dropout: 3-27                               --
│    │    └─MaxPool1d: 3-28                             --
├─LSTM: 1-2                                             264,192
├─Linear: 1-3                                           2,562
==========================================================================
Total params: 342,121
Trainable params: 342,121
Non-trainable params: 0
==========================================================================
```

FIGURE 6.1: CNN-LSTM Model for PTB DB

Algorithm 3: Classification of ECG signals with Raw Signals using CNN-LSTM

Input: A time series ECG raw data ts

Output: The classified label l

1: $ts \leftarrow$ RAW_VALUE_EXTRACTION(ts)
2: $features \leftarrow$ CNN(ts)
3: $l \leftarrow$ LSTM_CLASSIFICATION($features$)
4: **return** l

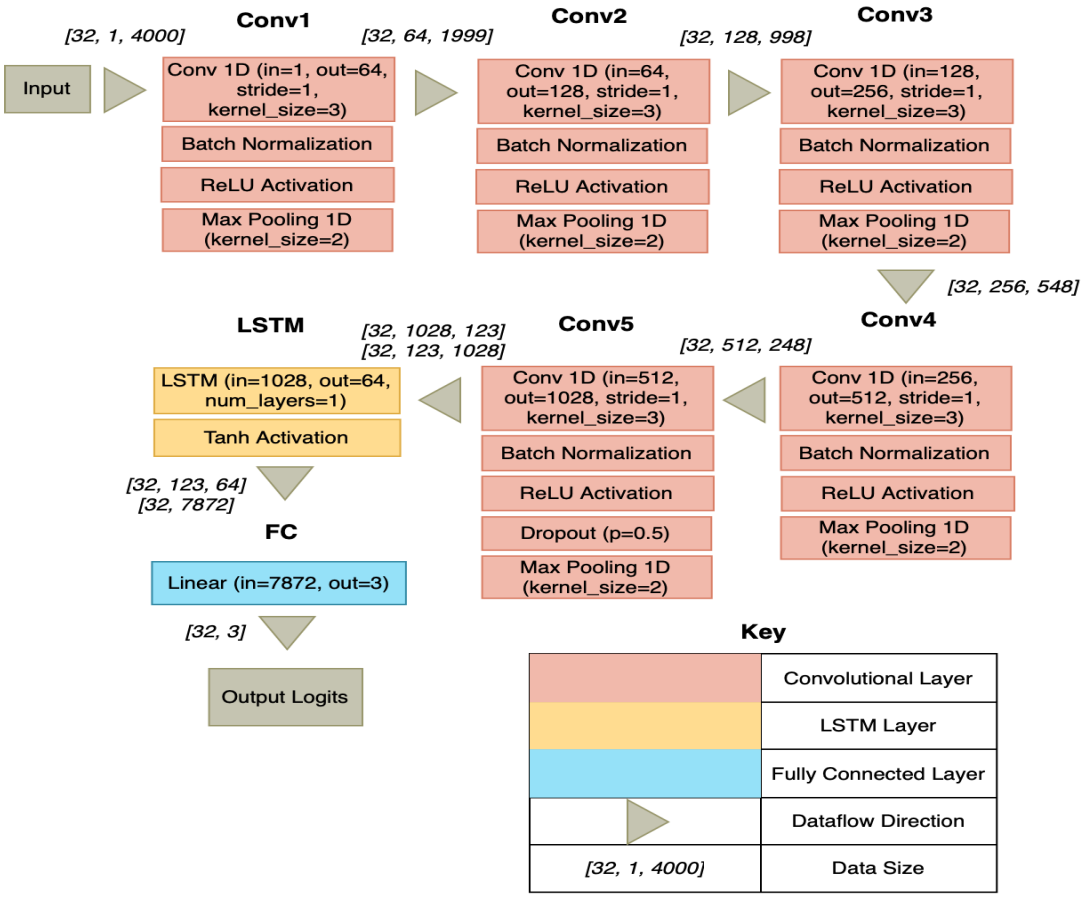


FIGURE 6.2: Final CNN-LSTM Architecture for Fall and HAR using ECG signals

Algorithm 3 outlines the algorithms for extracting features from the ECG signal and classifying them using a CNN-LSTM model. The number of CNNs and LSTMs can be varied but we used a maximum of five 1-d convolution layers in front of the three LSTM layers.

6.3.2 Attention Model

For reader's convenience, we recall the basic definitions of the attention mechanism following [249] and the notation therein. However, attention is discussed again in detail in Chapter 8, Section 8.2.1.

6.3.2.1 Attention

Attention is defined as

$$Attention(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{6.1}$$

where Q, K and $V \in \mathbb{R}^{n \times d_k}$ are input embedding matrices, n is the length of the (time) series, and d_k is the embedding dimension, resp.

The transformer uses Multi-Head Self-Attention (MHA) allowing the model to jointly attend to information at different positions of the time-series or different semantics of the domain. MHA is defined as

$$\text{MultiHead}(Q, K, V) = \text{Concat}\left(\text{head}_1, \text{head}_2, \ldots, \text{head}_h\right) W^O, \tag{6.2}$$

where h is the number of heads. Each head is defined as

$$\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \;=\; \text{softmax}\left[\frac{QW_i^Q (KW_i^K)^T}{\sqrt{d_k}}\right] VW_i^V, \tag{6.3}$$

where W_i^Q, $W_i^K \in \mathbb{R}^{d_m \times d_k}$, $W_i^V \in \mathbb{R}^{d_m \times d_v}$, and $W^O \in \mathbb{R}^{d_v \times d_m}$ (projection onto the output) are learned matrices and d_k, d_v are hidden dimensions of projection subspaces. For simplicity in the sequel, we drop the differentiation between d_m, d_k and d_v and refer to them by d.

The matrices Q, K and V are usually referred to as query, key and value matrices to remind of the associative memory architecture of a transform, compare e.g., also the analysis in [194].

6.3.2.2 Attention and Dimensional Embedding

For applying the attention mechanism to time-series one has to decide on the proper dimensional embedding, i.e., on the dimension of the embedding subspace and on the embedding transformation. We recall that in the domain of ECG the "natural" dimension is small. For instance, the signals are one-dimensional if a one-dimensional channel (single lead) is used (as is the case in this paper for the attention/transformer model, i.e., Algorithm 4). Even if multi-channel ECGs are used usually the number of channels is limited to a small number of 3 to maximally 12 channels. Henceforth, if we used the channel as the embedding, the dimension would be 1 in our case, i.e., $d = 1$. This is way too small to capture interesting patterns and, indeed, a test showed that the gradient descent does not converge, but stays constant after one or two initial updates. Furthermore, as depicted in the previous section, the self-attention suffers from an $\mathcal{O}(n^2)$ problem. We propose the following architecture to solve both problems simultaneously:

1. Assuming $m \ll n$. For simplicity of the notation, we assume without loss of generality that n is divisible by m, i.e., $n = mw$. This effectively segments the time-series n into n_m "windowed" segments of length w, where $m \in 1, \dots k$ with $k := n/w$. If n is not divisible without remainder, we could fill the time series with zeros (padding).

2. For each "windowed" sub- time-series t_{n_k} we calculate the decomposition to a chosen (fixed) wavelet by performing a discrete wavelet transformation (DWT), see below. Assuming that the result of applying the DWT is in dimension p, we have transformed the input from $\mathbb{R}^n$ into $\mathbb{R}^{m \times p}$ depicted in Fig. 6.3.

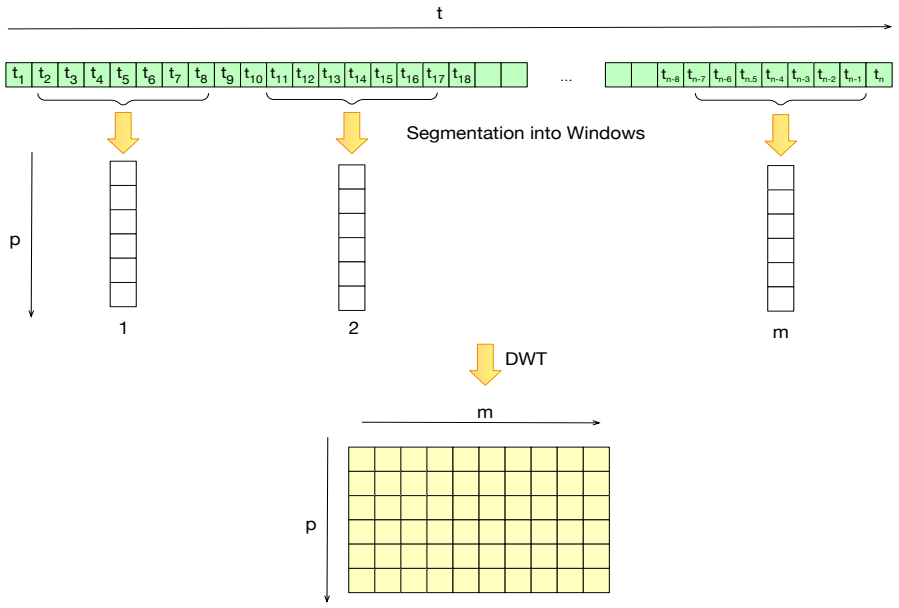


FIGURE 6.3: Dimensional Embedding

Remark 6.1. In this paper, we propose a deterministic embedding using DWT rather than a learned, randomly initialized embedding, which is an alternative approach that has been used in other attention architectures in the past. This should–in theory–require less training data–a conjecture that we want to validate in future work using synthesized data.

Remark 6.2. Note, that within the context of dictionary-based learning, a deterministic embedding using DWT could be considered as a "predefined analytical dictionary" [157]. Contrary to a fixed feature design using wavelet components, however, an embedding with an attention/transformer architecture has a flavor of learning the representation dynamically from the data as the proper amount of *attention* is learned from the data indeed. A systematic investigation of these aspects is deferred to future work, too.

More mathematical details on wavelets as feature extractors and dimensional embeddings are discussed in Chapter 7.

6.3.2.3 Transformer Architecture and Algorithm

The DWT can be used not only for dimensional embedding but also, for noise reduction as one can ignore some or all coefficients for the detailed spaces. To explore the impact, we tried several configurations, see Table 6.1[2].

TABLE 6.1: Attention Models

	Wavelet	Spaces	Dim Embedding	Hid Dim	Frequency	Epochs	Accuracy
ATT2	db8	V_0	13	150	0.1k	1k	98.14
ATT3	db8	V_0	12	150	0.1k	1k	98.83
ATT4	db8	$V_0 \oplus W_0$	24	150	0.1k	1k	99.18
ATT5	db8	$V_0 \oplus W_0$	24	150	0.1k	1k	99.11
ATT6	db8	$V_0 \oplus W_0$	26	250	0.1k	1k	99.45
ATT7	sym6	$V_0 \oplus W_0$	22	250	0.1k	1k	99.38
ATT8	db8	$V_0 \oplus W_0$	26	250	10k	1k	99.59
ATT9	db8	$V_0 \oplus W_0$	26	250	10k	2k	99.59
ATT10	db5	$V_0 \oplus W_0$	20	250	10k	1k	99.24
ATT11	db5	$V_0 \oplus W_0 \oplus W_1$	28	250	10k	1k	99.73

We deployed a transformer architecture with one head, four transformer encoder layers, and a dimension of 150 or 250 hidden units of the feed-forward network. The network's architecture of the best model, ATT11, is illustrated in Fig. 6.4.

```
==========================================================================================
Layer (type:depth-idx)                                                   Param #
==========================================================================================
click to unscroll output; double click to hide
├─PositionalEncoding: 1-1                                               --
├─TransformerEncoder: 1-2                                               --
│    └─ModuleList: 2-1                                                  --
│    │    └─TransformerEncoderLayer: 3-1                                17,638
│    │    └─TransformerEncoderLayer: 3-2                                17,638
│    │    └─TransformerEncoderLayer: 3-3                                17,638
│    │    └─TransformerEncoderLayer: 3-4                                17,638
├─Dropout: 1-3                                                          --
├─Linear: 1-4                                                           954
==========================================================================================
Total params: 61,762
Trainable params: 61,762
Non-trainable params: 0
==========================================================================================
```

FIGURE 6.4: Transformer Architecture of Model ATT11

The original input tensor has a dimension of $[14552, 1, 187]$ corresponding to 14522 data rows, 1 feature, and a sequence length of 187. The sequence was split into 17 chunks, with a window length of 11. Each subsequence of length 11 was converted using the DWT. For

[2]Frequency refers to the frequency of the positional encoding. It should be remarked, that model ATT5 differs from ATT4 by an additional residual connection.

instance, for db8 for ATT7, this leads to a transformed tensor of $[14552, 22, 17]$. (Note, that due to boundary effects, the embedding dimension is not always a multiple of 11.).

As a positional encoding, we tried the usual Fourier encoding and used frequencies f of $f = 100$ and $f = 10.000$, resp. While adding positional encoding is questionable after embedding using DWT, we experimentally found the results to be improved by a small amount.

All models were trained with a batch size of 256 and a learning rate of 0.001 using the Adam optimizer [128]. Algorithm 4[3] layouts the algorithms for extracting features from ECG data and classifying them using a transformer/attention model.

Algorithm 4: Classification of ECG signals with Raw Signals using Attention/-Transformer

Input: A time series ECG raw data ts
Output: The classified label l

$X \leftarrow$ RAW_VALUE_EXTRACTION(ts)
$(V_0 \oplus W_0 \oplus W_1 \oplus \ldots \oplus W_m) \leftarrow \text{DWT}(X)$
$(V_0 \oplus \ldots \oplus W_m) \leftarrow (V_0 \oplus \ldots \oplus W_m) + POS\_ENC(V_0 \oplus \ldots \oplus W_m)$
for $i \in$ Layers **do**
    $X \leftarrow \text{TRANSFORMER}_i(V_0 \oplus \ldots \oplus W_m)$
end for
$l \leftarrow$ LINEAR_FEED_FORWARD(X)
return l

6.3.3 Complexity Analysis

6.3.3.1 Runtime Complexity Analysis

In general, the runtime complexity for attention based RNN/LSTM and CNN architectures are known as depicted in Table 6.2, see e.g. [249][4], where n denotes the sequence length, d the embedding dimension, and k the size of the kernel (in case of CNN).

TABLE 6.2: Generic Runtime Complexity Analysis

Layer Type	Layer Compl.	Seq. Ops	Max Path Length
Attention	$\mathcal{O}(n^2 d + nd^2)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$
RNN/LSTM	$\mathcal{O}(nd^2)$	$\mathcal{O}(n)$	$\mathcal{O}(n)$
CNN	$\mathcal{O}(knd^2)$	$\mathcal{O}(1)$	$\mathcal{O}(\log_k(n))$

[3] Note, that for simplicity of notation we use the expression $(V_0 \oplus \ldots \oplus W_m)$ from the decomposition in equation 7.10 generically, i.e., some of the components of $(V_0 \oplus \ldots \oplus W_m)$ might be empty.
[4] Note, that we have added the complexity caused by the query matrices which were omitted in [249].

Note, that the maximum path length measures the maximum length between any two input and output positions in the networks. Shorter path length makes it easier to learn long-range dependencies.

Considering, that the dimensional embedding using wavelets is of $\mathcal{O}(n)$ and has to be computed only once and henceforth can be ignored compared to $\mathcal{O}(n^2d + nd^2)$, we conclude from Table 6.2 the complexity of our algorithms as depicted in Table 6.3.

TABLE 6.3: Runtime Complexity Analysis for Algorithms 3 and 4

Algorithm	**Total Complexity**	**Max Path Length**
Alg 3	$\mathcal{O}(n^2d^2)$	$\mathcal{O}(n + \log_k(n)) = \mathcal{O}(n)$
Alg 4	$\mathcal{O}(n^2d + nd^2)$	$\mathcal{O}(1)$

From this analysis, we can conclude that Alg 3 is always inferior to Alg 4 in terms of algorithmic complexity. In addition, the transformer can be easily paralleled (typically on a GPU), contrary to a CNN-LSTM.

Please note, that the above analysis assumes that the matrix multiplication of two matrices $\mathbf{A} \in \mathbb{R}^{nm}$ and $\mathbf{B} \in \mathbb{R}^{ml}$ is in $\mathcal{O}(nml)$, which corresponds to a naive implementation of matrix multiplication. Although this can be improved, compare Strassen's algorithm [230], and–more recently–Josh Alman and Virginia Vassilevska Williams algorithm [8], these algorithms are normally *not* implemented in the machine learning frameworks used and henceforth the naive implementation as the usual convention is assumed.

6.3.3.2 Memory Complexity Analysis

As for backpropagation, the weights optimized have to be kept in memory, to optimize the algorithms efficiently, we have essentially the same space as time complexity, particularly space complexity of any attention-based model of $\mathcal{O}(n^2)$.

6.4 Data Preparation and Experimental Setup

Since our study includes multiple data sets, therefore this section explains the preparation steps taken for each data set and the overall experimental setup. Experiments were performed using a GPU server. All the experiments were implemented using the PyTorch library because of its supportive architecture with GPUs. The main aim of the experiments was fall and MI detection using ECG signals in an automated and efficient manner.

6.4.1 ECG Data Set for Fall Detection

To the best of our knowledge, the ECG HAR data set is the only one for the detection of different human activities including falls, using ECG signals. It was originally collected by [34], as an experiment that was part of the study by [25]. It originally consisted of two classes: one for the ECG of a person falling from the bed and another one for the ECG of a resting person. It was later augmented with two more data sets, [124] and [127], by up-sampling the original data set. In addition to that, another augmentation method called slicing was applied to the data set. Slicing has been explained in detail in [55]. After the addition of new data sets, the final version has three classes namely: fall, rest, and daily activities.

The overview of the final class distribution in data set is depicted in Table 6.4.

TABLE 6.4: Total Number of Samples in the ECG HAR Data Set [34]

Label	**Total Count**
Fall	500
Rest	474
DA	296
Total	1270

In the previous experiments, the data set was filtered, converted to wavelet transform, and later to 3-D images called scalograms. These scalograms were first used to fine-tune and then train, a pre-trained AlexNet and GoogLeNet. The state-of-the art validation accuracy obtained for classification with this data set is 98.44%. This accuracy was obtained after applying extensive pre-processing to the data set. Our current model outperforms the state-of-the-art validation accuracy and achieves a 99.21% accuracy with no pre-processing and only fine tuning the ensemble model.

6.4.2 PTB Diagnostics Data Set

After working with the ECG for falls and daily activities, the model had to be tested on a standardized data set that is publicly available. In the second set of experiments, a publicly available data set called the PTB diagnostic was used, which is freely available but is used as a standard for ECG classification tasks.

The original PTB data set consists of 549 records from 290 subjects which were aged 17 to 87 years, with a mean age of 57.2. A total of 209 subjects were males with a mean age of 55.5 and 81 females with a mean age of 61.6 (for 1 female and 14 male subjects; age was not recorded). Each record has 15 measured signals: the conventional 12 leads (i, ii, iii, avr, avl, avf, v1, v2, v3, v4, v5 and v6) together with the 3 Frank lead ECGs

(vx, vy and vz). The data from lead II were used to train the model which outperforms the databases which even use all 12 lead data [27]. ECG beats were extracted using the method described in [118]. The data set used was divided into two classes: normal and abnormal (myocardial infarction). In the previous prominent study [91], all 12 leads were separately evaluated to determine which leads contributed the most to the classification. We used lead II of the data set to differentiate between healthy controls and that with myocardial infarction. Since only one lead of ECG was used in the previous two experimental phases, we used another publicly available data set and used all 12 of its leads to reaffirm the usefulness of the ensemble model for both uni-variate and multivariate data sets.

6.4.3 PTB XL Diagnostics

PTB-XL is one of the largest freely accessible ECG data sets available. It was collected over a span of seven years between 1989-1996. It was made publicly available in 2020 in a structured database by Physikalisch-Technische Bundesanstalt (PTB). The data set consists of a total of 21837 records of 12-lead ECG each comprising of 10 s. It is a gender-balanced data set with 52% male and 48% female records and an age range of 0-95 years. The data set consisted of various diagnoses and a large number of healthy controls as well [232]. PTB XL has a standardized set of pre-processing instructions for the data set. Because different labels are heavily imbalanced and imbalanced classes can introduce bias in the trained model, it is important to divide the data set in a way that each of the classes is represented equally in each subset. Stratified sampling was used to divide it into training-validation-testing data sets. The data set has multiple classification categories as shown in Fig. 6.5. The goal is to classify MI from other heart conditions, and models were trained for diagnostic superclass and myocardial infarction detection using Algorithm 3.

6.5 Results

Several methods to evaluate a DL classifier exist in the literature. We evaluated our classifiers using the accuracy, area under the curve (AUC), confusion matrix, sensitivity, and specificity. The details of the results obtained by applying each of the algorithms and the respective data set is explained in this section. In the sequel, we present the results acquired for each data set and also compare our results to the other state-of-the-art work. An overview of the section is presented in Table 6.17.

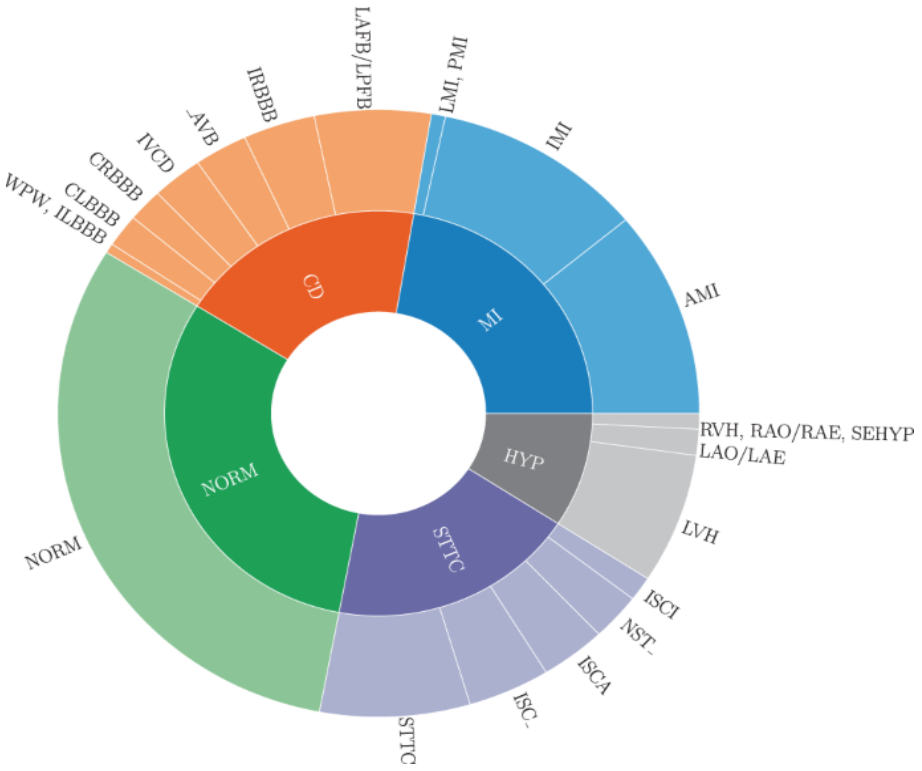


FIGURE 6.5: Class Distribution for PTB-XL Data Set

6.5.1 ECG HAR Data Set

The data set was initially trained using a plain LSTM to compare the model performance with the previous experiments. The data were fed into the model without any pre-processing. The LSTM initially yielded an accuracy of 49.80% which was increased to 54% by fine-tuning the hyperparameters. In the previous experiments, extensive pre-processing was carried out to extract the related features and then those features were fed into the model. Although that approach yields excellent accuracy, it is not automated. LSTMs have been shown to have a sense of previous timestamps or history in the time series, but CNNs have a superior feature extraction capability. To test our hypothesis, a CNN was placed on top of the LSTM layer. The accuracy immediately improved to 93%. After some fine tuning the hyperparameters and adjusting the number of CNN layers, the validation accuracy got better than the state-of-the-art results. A testing accuracy of 99.21%–100% was achieved and a validation accuracy of 99.21% was achieved. For the first data set, the results were almost perfect with a validation accuracy of 99.21% and a testing accuracy of 99.21%–100%. The previous work achieved similar accuracy but with transfer learning and pre-processing the signals by converting them into wavelet transforms and then into scalograms. This model achieves similar accuracy even by avoiding all those steps. The following Table 6.5 depicts the confusion matrix for the testing data set showing an almost perfect accuracy of 99.22%.

TABLE 6.5: Confusion Matrix for Fall Detection ECG Data Set using CNN-LSTM (Algorithm 3)

		Actual		
		DA	Fall	Rest
Predicted	DA	30	0	0
	Fall	1	52	0
	Rest	0	0	45

Fall detection using ECG signals was also performed by applying Algorithm 4 to the HAR data set. Each sequence in the data set consists of 4000 time stamps. The initial tensor size was [1273, 1, 4000], which is in the format [total Sequences, number of features, sequence length]. Each of the ECG sequences was divided into 100 chunks of 40 time stamps each, and then the wavelet transform was calculated for each chunk resulting in a final dimension of [1273, 108, 40]. The model was trained in 403.39 s. This result was again achieved without any manual feature extraction or transfer learning model. The following Table 6.6 depicts the confusion matrix for the testing data set showing also the accuracy of 95.31%

TABLE 6.6: Confusion Matrix for Fall Detection ECG Data Set using Attention (Algorithm 4)

		Actual		
		DA	Fall	Rest
Predicted	DA	29	1	0
	Fall	2	49	2
	Rest	0	0	44

6.5.2 PTB Diagnostics

Algorithm 3, i.e., CNN-LSTM was used to model the PTB diagnostic to differentiate normal from abnormal heartbeats. Previous studies have emphasized feature extraction before feeding into the neural network, or transfer learning where the model is initially trained with an existing data set and later on trained with the same learned weights on the desired data set such as in [265]. In the current benchmark for MI classification using PTB diagnostic, ConvNetQuake neural network model was adapted to achieve an accuracy of 99.44%. Similarly, heavy pre-processing, such as wavelet transformation [2], data balancing [191], and transfer learning [118], are used in the literature to achieve higher accuracy for ECG signal classification. In our study, no pre-processing of the individual readings was applied, and the model achieving 99.66% accuracy, exceeded the state-of-the-art accuracy for normal versus abnormal classification, which was previously 99.43%.

Algorithm 4, i.e., the attention /transformer model was also used to model the PTB diagnostic to differentiate between normal and abnormal heartbeats. This yielded an accuracy of 99.73%, a precision of 99.73%, a sensitivity of 99.2%, and a specificity of 99.91%.

The confusion matrices of both algorithms are depicted in Table 6.7 below.

TABLE 6.7: Confusion Matrices for PTB Data Set

		Predicted			
		Algorithm 3		Algorithm 4	
		NORMAL	ABNORMAL	NORMAL	ABNORMAL
Actual	NORMAL	371	1	372	1
	ABNORMAL	2	1081	3	1080

In comparison to CNN-LSTM, i.e., Algorithm 3, the attention model with wavelet embedding has been shown to be more efficient as it uses fewer parameters and less training time as compared to the CNN-LSTM model as shown in Table 6.8. However, the time and number of parameters for Algorithm 4 might increase eventually with the increase in the number of attention heads and encoder layers.

TABLE 6.8: Parameter Comparison between State of the Art and Our Work. Key for Training Hardware: 1 = 2 NVIDIA Titan Xp GPUs ,2 = 2 NVIDIA 2080Ti GPUs, 3= i5 core, NVIDIA graphics card, 4= NVIDIA A100-PCIE-40GB

Work	Total Parameters	LR	Time to Train	Training Hardware	Batch Size	Epochs	Optimizer
Liu et al.[145]	NA	1e-3 - 1e-6	3557.26 s	1	32	100	NA
Wang et al.[252]	NA	0.001	NA	2	12	10	NA
Rai et al. [191]	NA	0.001	1263 s - 2285 s	3	128	100	Adam
Our work (HAR): Alg. 2	**461,583**	0.00002	**403.39**	4	20	500	Adam
Our work (HAR): Alg. 1	2,471,503	0.001	80.56s	4	64	100	Adam
Our work (PTB): Alg. 2	**61762**	0.001	**823.54s**	4	256	1000	Adam
Our work (PTB): Alg. 1	210,025	0.001	1923.87s	4	64	1000	Adam

A comparison between the reference parameters used across the state-of-the-art similar work and our work is shown in Table 6.8. However, it must be noted that Table 6.8 is not complete because not all parameters can be found for all related work and NA in the table refers to not available.

Multiple metrics were used to evaluate the model performance. The terms tp, fp, tn, and fn refer to the true positives, false positives, true negatives, and false negatives respectively. In medical terminology, true positive would refer to the medical condition being diagnosed, so tp in our context refers to the diagnosis of MI. The performance metrics were calculated using the following formulas:

$$\begin{aligned} Accuracy &= (tp + tn)/(tp + fp + fn + tn) \\ Precision &= tp/(tp + fp) \\ Sensitivity &= tp/(tp + fn) \\ Specificity &= tn/(tn + fp) \end{aligned}$$

The results for our leading models are summarized in Table 6.9.

TABLE 6.9: Metrics for Leading CNN-LSTM and Attention

		Predicted	
		CNN-LSTM	ATTENTION
Actual	Accuracy	99.79	99.73
	Precision	99.91	99.91
	Sensitivity	99.81	99.72
	Specificity	99.73	99.73

Fig. 6.6 and Fig. 6.7 show examples of the training accuracy and losses for the PTB data set, respectively.

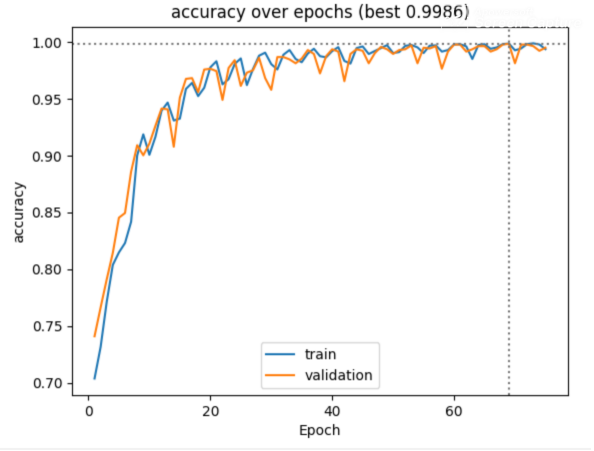


FIGURE 6.6: Training and Validation Graph over Epochs for the PTB Data Set for Algorithm 3

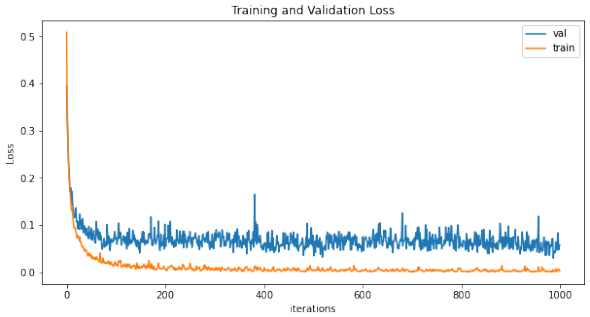


FIGURE 6.7: Training and Validation Loss for the PTB Data Set for Algorithm 4

6.5.3 PTB XL Diagnostics

Since PTB XL is a relatively new data set, many recent studies using this data set have adapted it for different classification tasks such as super diagnostic, sub-diagnostic, and form etc. In our study, five super diagnostic (SD) classes were classified. A validation accuracy of 75.70% and a testing accuracy of 74.33% were achieved. An AUC score of 0.8395 was obtained, see Table 6.10.

TABLE 6.10: Performance Indicators

Classification classes	Val. Accuracy	Testing Accuracy	AUC score
Super-diagnostic Classes	75.70%	74.33%	0.8395
MI vs. Normal Class	90.94%	89.1%	0.87
MI vs. Super-diagnostic classes	91.07%	90.2%	0.762

A direct comparison to the state-of-the-art is not very straightforward for PTB-XL mainly because it is a newer data set and many new studies that use it focus on different classification tasks. Although our results for this data set do not exceed the state-of-the-art accuracy, they are still comparable. Virginia et al. [200] and Martin et.al [154] achieved accuracies of 90.8% and 77.12%, respectively, for MI classification respectively. Similarly, Śmigielet et al. [278] achieved an accuracy of 78.0–75.2% for five-class classification.

Our Algorithm 3 applied for myocardial infarction detection using the PTB XL data set, yields a validation accuracy of 90.94% when it is MI versus the normal class in super diagnostic, and 91.27% when it is MI vs other four superclasses. Please note the AUC as a metric was calculated only for PTB-XL data set as it is widely used for

comparison of this particular data set in the literature. The experiment with Algorithm 4 on multi-lead data sets such as PTB XL is still in a preliminary stage and requires further investigation on how to merge the natural domain dimension of multi-lead with the dimensional embedding technique. However, similar results for another research project of ours yield promising results for the Algorithm 4 with a multidimensional or multi-featured data set, see [210]. Henceforth these experiments have not been mentioned and as work in progress will be published in a future contribution.

6.5.4 Statistical Analysis

To assess the statistical validity of the results, we computed a five-fold cross-validation for both algorithms and all data sets, see Table 6.11. For PTB-XL data set the k-fold validation was done only for super-diagnostic classes.

Table 6.11: Five Fold Cross-Validations

		Fold					
Data	Alg.	0	1	2	3	4	Avg
ECG HAR	1	97.26	98.43	98.43	98.03	98.82	98.19
	2	94.90	90.59	91.37	93.70	93.31	92.77
PTB	1	99.00	99.56	99.28	99.07	99.31	99.24
	2	96.39	97.11	97.49	96.25	96.49	96.75
PTB XL	1	76.22	75.42	75.38	74.00	76.49	75.50

As one can see, the results are consistent with the findings in sub section 6.5.1, the difference in the average results from the reduced training in case of five fold cross validation.

However, to assess the statistical meaningfulness, a more sophisticated approach is required. In a seminal paper [65], Dietterich analyzed five approximate statistical tests for determining whether one learning algorithm outperforms another on a particular learning task. It includes the well-known McNemar's test for a single pass validation and also proposes a new 5x2cv test designed for algorithms where at least 10 validations can be carried out. In the paper, Dietterich shows that the null-hypothesis of the two algorithms to compare having the same performance, the off-diagonal elements of the confusion matrix should be the same, which can be checked statistically for significance using a χ^2 test or a test for t statistics.

Although Dietterich's paper has been very well received and is cited many thousand times, in the practice of machine learning–contrary to other disciplines like e.g., the medical sciences–, statistical analysis of significance is still not common and is therefore

usually not included in publications, unfortunately. This limits not only the interpretation of published results but also limits the ability to rigorously compare benchmarks. For instance, the tests proposed in [65] assume the availability of the data set backing the confusion matrix. In particular, to apply any of the tests in order to compare two algorithms A_1 and A_2, one must determine the incorrectly classified samples from A_1 and check whether these are correctly classified by algorithm A_2 and vice versa in order to determine the statistical parameters needed for the test. Even if the data are publicly available, information on which sub samples are incorrectly classified is usually *not* available from the publication. In our case, these data are clearly not available for any of the bench marking publications. Therefore, we could only compare our algorithms 3 and 4 against each other, but *not* against any of the bench marked algorithms.

The implementation of the statistical analysis and the results are discussed in the next sub section.

6.5.4.1 McNemar's test

The McNemar's test is a standard paired test used in the medical field for the verification of usability of the new drugs etc. However, it is not very commonly applied in the field of deep learning for model comparison. Because we used biomedical data in this study, McNemar's test was used to verify the statistical significance of the results obtained using the proposed algorithms. To apply McNemar's test, our data set was partitioned into training and testing sets, called R and T, respectively. After training both models A_1 (Algorithm 3) and A_2 (Algorithm 4) on t, the classifiers were tested on each instance of R and eventually the following statistics are collected:

- n_{00} : Number of classes misclassified by both classifiers A_1 and A_2.
- n_{01} : Number of data instances misclassified by A_1 but not A_2.
- n_{10} : Number of data instances misclassified by A_2 but not A_1.
- n_{11}: Number of data instances misclassified by neither A_1 nor A_2.

Additionally, $n_{00}+n_{01}+n_{10}+n_{11} = n$, where n is the total data instances in the test set T. The contingency table layout for McNemar's test is presented in Table 6.12.

The McNemar's test was performed for both PTB and ECG HAR data set. The null hypothesis (H_0) is that both algorithms have the same error rate, i.e., $n_{01} = n_{10}$. The confidence interval for all tests was 95%. The statistics obtained for the ECG HAR data set are presented in Table 6.13.

n_{00}	n_{10}
n_{01}	n_{11}

TABLE 6.12: Layout

6	31
1	217

TABLE 6.13: ECG Data Set

1	11
4	1440

TABLE 6.14: PTB Data Set

TABLE 6.15: McNemar's Contingency Tables

For an alpha value of 0.05, the p-value is calculated to be numerically 0.000, which implies that our test is significant enough to reject the H_0 and we conclude that both models have different proportions of errors and are significantly different in this data set. The same test was repeated for the PTB data set and the obtained statistics are listed in Table 6.14.

The p-value obtained for this test was 0.118 which is greater than 0.05 hence, there was no significant evidence to reject H_0.

6.5.4.2 The 5x2 cv *t* test

The 5x2 cv t test is introduced in Dietterich [65] and recommended therein: "For algorithms that can be executed ten times, the 5x2 cv test is recommended as it is slightly more powerful and because it directly measures the variance due to the choice of training set". For this test, two-fold cross validation was performed for five repetitions. During every repetition, the data set was randomly partitioned into two equal-sized sets S_1 and S_2. Both algorithms were trained on each set and tested on the other set. This results in four error estimates: $P_{A_1}^{(1,2)}$ and $P_{A_2}^{(1,2)}$ with A_1 or A_2, resp. trained on S_1 and tested on S_2 and $P_{A_1}^{(2,1)}$ and $P_{A_2}^{(2,1)}$ with A_1 or A_2, resp. trained on S_2 and tested on S_1. Estimated differences are obtained by subtracting the corresponding error estimates $P^{(1,2)} = P_{A_1}^{(1,2)} - P_{A_2}^{(1,2)}$ and $P^{(2,1)} = P_{A_1}^{(2,1)} - P_{A_2}^{(2,1)}$. From these differences, the estimated variance σ^2 is calculated as $\sigma^2 = (P^{(1,2)} - \bar{P})^2 + (P^{(2,1)} - \bar{P})^2$, where $\bar{P} = (P^{(1,2)} + P^{(2,1)})\ /2$. Let σ_i^2 be the variance calculated from the i-th replication. Then the 5x2cv $\bar{t}$ statistic is

calculated as follows:

$$\bar{t} = \frac{P_1^{(1,2)}}{\sqrt{\frac{1}{5}\sum_{i=1}^{5}\sigma_i^2}}$$

Under the H_0, $\bar{t}$ has approximately a t distribution with five degrees of freedom. The calculated t statistic for PTB data set and ECG HAR data set is 3.002 and 3.286 respectively. Detailed tables for the five repetitions are presented in Appendix 6.7.2 in Table 6.18 and Table 6.19. As both would have a corresponding p value of 0.030 and 0.0218 respectively, it clearly shows that both models are significantly different from each other with different error estimates.

6.5.4.3 Interpretation

Taking the results from both the McNemar and the more powerful 5x2 cv t test we can conclude that our algorithms differ significantly and that the obtained Key Performance Indicator (KPIs) are statistically meaningful for the standard confidence interval of 95%.

6.5.5 Summary

As seen in Table 6.16, our model leads to almost all of the evaluation criteria for the classification of PTB data set.

6.6 Discussion

As mentioned earlier, state-of-the-art accuracies were achieved using the CNN-LSTM model for three data sets and the attention model for PTB data set. In similar previous studies, mostly one set of experiments is performed with a single database to prove the usability of the models. However, we worked on three data sets separately. The first data set was used to classify human activities including falls. The second one consists of extracted heartbeats for the classification of MI vs normal heartbeats. The third data set consists of a 12-lead ECG data set for multiple cardiovascular conditions. The success of our proposed algorithms on all three data sets generalizes their usefulness for ECG classifications over multiple tasks.

Hybrid models help to combine the features of the base models. This is often more powerful than very deep models with hundreds of layers because deeper models tend to over-fit for medium-sized data sets. An LSTM model keeps track of the past trends in the time series and can also help in the prediction of the next time stamps. In our study,

TABLE 6.16: Our Result Compared with other similar Studies in Literature which used PTB Database (Built upon [91])

Work	**Accuracy(%)**	**Sensitivity(%)**	**Specificity(%)**	**Precision(%)**
Acharya et al. [2]	93.5	93.7	-	92.8
Safdarian et al. [204]	94.7	-	-	-
Kojuri et al. [132]	95.6	93.3	-	97.9
Sun et al. [234]	-	92.6	-	82.4
Liu et al. [143]	94.4	-	-	-
Sharma et al. [220]	96	93	-	99
Kachuee et al. [118]	95.9	95.1	-	95.2
Remya et al. [196]	93.61	93.22	94.28	-
Reasat et al. [195]	84.54	85.33	84.09	-
Zewdie et al. [273]	98.3	98.7	96.4	-
Feng et al. [75]	95.4	98.2	86.5	-
Strodthoff et al.	-	93.3	89.7	-
Huang et al.	96.96	99.89	92.51	95.35
Liu et al. [145]	98.59	99.53	94.50	-
Gupta et al.[91]	99.43	99.40	99.45	99.46
Ours (CNN-LSTM)	**99.93**	**99.81**	99.73	**99.91**
Ours (Attention)	**99.73**	99.72	**99.73**	**99.91**

TABLE 6.17: Overview of the Experiments with Different Data Sets and the Acquired Performances

Classification task	**Data set**	**Achieved Accuracy**	**Algorithm**	**State-of-the-art accuracy**
Fall detection	ECG HAR	99.21%	Attention	98.44% [34]
Fall detection	ECG HAR	99.21%	CNN-LSTM	98.44% [34]
MI detection	PTB	99.73%	Attention	99.44% [91]
MI detection	PTB	99.93%	CNN-LSTM	99.44% [91]
SD Class	PTB XL	75.70%	CNN-LSTM	-
MI detection	PTB XL	91.07%	CNN-LSTM	-

the results of the CNN-LSTM model have shown to be always better than both of the models implemented individually. This was verified for the HAR data set by [188] and we compare the results from Table 6.16 for PTB data set where multiple variations of CNN and LSTMs have been applied separately in the previous works. The performance of the model on the HAR data set is observed to increase up to a certain level with the increase in a) the number of filters in the conv1d layer for CNN-LSTM and b) the number of

dimensions in the dimensional embedding with the attention model. Since the data set is not very large, a final conclusion cannot be drawn at this stage but it merits further investigation. The attention algorithm clearly has a computational advantage over the CNN-LSTM algorithm as seen in Table 6.8. It takes less time to converge and even has fewer parameters to train than the CNN-LSTM algorithm. Our study had the following advantages:

- A hybrid CNN-LSTM model and attention with a discrete wavelet transformation as an embedding are proposed.
- No or very little manual feature extraction is required for training the model.
- Three publicly available data sets were used separately for the training using the proposed models.
- State-of-the-art accuracy of 99.86% and 99.44% is achieved for the PTB data set and ECG for HAR classification respectively without any feature extraction or pre-processing.
- Multiple standard statistical analysis techniques were applied to the acquired results to statistically support our algorithms.

Hence, we addressed our research question and achieved results equivalent to many recent studies without any pre-processing or feature extraction. We have also shown to train the models in an efficient manner computationally.

As part of the future work, the authors would like to explore the difference between the two algorithms using explainable AI. Looking deeper into the gradients for each layer would shed light into the learning process.

6.7 Conclusion

The models proposed and explained in this paper aim to better classify ECG time series for different conditions using minimum pre-processing steps. Publicly available data sets have made it possible to verify the robustness and usefulness of the proposed models by achieving state-of-the-art accuracy using multiple data sets. This would eventually help medical practitioners to identify multiple heart conditions automatically with minimum feature extraction. Specifically for the MI classification, because the results are close to 100%, the model is ready to be deployed for medical evaluation.

6.7.1 Data and Code Availability

See Appendix B.

6.7.2 5x2 cv t-test table

The tables including full details of the two 5x2 cv t-tests, Table 6.18 and Table 6.19, resp.

TABLE 6.18: 5x2 cv Test Contingency Table for PTB Data Set

	Rep 1	Rep 2	Rep 3	Rep 4	Rep 5
Model A	$P_A^{(1)} = 0.9915$ $P_A^{(2)} = 0.9922$	$P_A^{(1)} = 0.9923$ $P_A^{(2)} = 0.9904$	$P_A^{(1)} = 0.9934$ $P_A^{(2)} = 0.9889$	$P_A^{(1)} = 0.9934$ $P_A^{(2)} = 0.9927$	$P_A^{(1)} = 0.9929$ $P_A^{(2)} = 0.9894$
Model B	$P_B^{(1)} = 0.9786$ $P_B^{(2)} = 0.9839$	$P_B^{(1)} = 0.9839$ $P_B^{(2)} = 0.9733$	$P_B^{(1)} = 0.9778$ $P_B^{(2)} = 0.9812$	$P_B^{(1)} = 0.9799$ $P_B^{(2)} = 0.9805$	$P_B^{(1)} = 0.9759$ $P_B^{(2)} = 0.9789$

TABLE 6.19: 5x2 cv Test Contingency Table for ECG Data Set

	Rep 1	Rep 2	Rep 3	Rep 4	Rep 5
Model A	$P_A^{(1)} = 0.96860$ $P_A^{(2)} = 0.96698$	$P_A^{(1)} = 0.95918$ $P_A^{(2)} = 0.977987$	$P_A^{(1)} = 0.97327$ $P_A^{(2)} = 0.96232$	$P_A^{(1)} = 0.9545$ $P_A^{(2)} = 0.9733$	$P_A^{(1)} = 0.96232$ $P_A^{(2)} = 0.9733$
Model B	$P_B^{(1)} = 0.86656$ $P_B^{(2)} = 0.8522$	$P_B^{(1)} = 0.8477$ $P_B^{(2)} = 0.8805$	$P_B^{(1)} = 0.8349$ $P_B^{(2)} = 0.85714$	$P_B^{(1)} = 0.85714$ $P_B^{(2)} = 0.88522$	$P_B^{(1)} = 0.85714$ $P_B^{(2)} = 0.8349$

Chapter 7

Feature Extraction using Wavelet Transformation

> "The supreme goal of all theory is to make the irreducible basic elements as simple and as few as possible without having to surrender the adequate representation of a single datum of experience." – (Einstein's razor)

As discussed in chapter 4, feature extraction is an important pre-processing step in time series analysis. In the classical approach, this step is largely based on prior domain knowledge and classical statistical methods like mean, median, kurtosis, percentiles, or temporal aggregations like resampling, rolling averages and standard deviations are used. Many latest models use state-of-the-art feature extraction techniques like Fast Fourier Transform, Spectral Analysis, Principal Component Analysis (PCA), Auto and Cross correlation, and Wavelet Transformation etc. to name a few. In this study, we have used the wavelet transformation as a major feature extraction technique. Therefore, this chapter provides a comprehensive background and highlights the properties of wavelets that makes them suitable for feature extraction particularly for time series analysis.

7.1 Feature Extraction for Time Series and Bio Medical Signals

Pattern recognition tasks require the data to be pre-processed for extracting features and later selecting the relevant ones before feeding them to the task. Feature extraction is one of the most important pre-processing steps because right extraction directly effects the effectuality of the recognition task as more discriminate features assist in effective

identification of the classification groups. Though originally pattern recognition techniques were not crafted to manage large amount of irrelevant features, it is a necessity with the current plethora of data sets to combine feature selection and feature extraction with these algorithms.

The feature selection have many benefits including but not limited to the following [203][92]

- It aids data understanding by visualization and by gaining a deeper insight into the under laying process of the data generation.
- Improves the performance of the prediction algorithm by avoiding over fitting and improve model performance, i.e. prediction performance in the case of supervised classification and better cluster detection in the case of clustering.
- Reduces the storage and measurement requirements by defying the curse of dimensionality.
- By reducing the training and utilization time.

However the feature selection and extraction techniques creates an additional layer of complexity in pursuit of the search for a subset of relevant features. Consequently, the exploration within the model hypothesis space is expanded by an additional dimension, involving the quest for the optimal subset of pertinent features [76]. The end result of feature extraction and feature selection is a set of features usually named feature vector, which is a projection or representation of the data. The classification model then maps the feature vector to a classification class [76] [176].

Despite them being non-stationary signals, the variance of biomedical signals can be indicators of a potential disease, an abnormality or an indicator of an existing disease. These indications can vary over different windows of observation i.e., intermittently, persistently or even at random. This adds an additional computational challenge to analyse and detect the abnormalities from the normal signals [76]. Hence, the importance of feature extraction step is evermore while dealing with biomedical signals.

7.2 Wavelets – An Introduction

The signals can be analyzed both in time and frequency domain depending on the functional scenario. Fourier transform analyzes the signal in frequency domain and not spatial domain. However, the signals are represented in time and frequency locally by wavelet

transformation and if they are chosen properly, the wavelet transform also serves as a basis with compact support in both the frequency and time domains.

Wavelets are a family of basis function which are derived by translating and dilating operations on a single generating function called mother wavelet. Dilation or scaling stretches the mother wavelet and translation shifts it along the time axis. Introduced in 1910 by Haar [94] as a piece-wise constant function

$$\Psi(t) = \begin{cases} 1, & if \quad 0 \leq t < 1/2 \\ -1, & if \quad 1/2 \leq t < 1 \\ 0, & otherwise \end{cases} \tag{7.1}$$

whose orthonormal basis of the space $L^2(\mathbb{R})$ are generated by the dilation and translations given by

$$\left\{ \Psi_{j,n}(t) = \frac{1}{\sqrt{2^j}} \Psi \left(\frac{t - 2^j n}{2^j} \right) \right\}_{(j,n)\in\mathbb{Z}} \tag{7.2}$$

of signals having a finite energy

$$\|f\|^2 = \int_{-\infty}^{+\infty} |f(t)|^2 dt < +\infty. \tag{7.3}$$

If the inner product of the function g is written as $\langle f, g \rangle = \int_{-\infty}^{+\infty} f(t) g^*(t) dt$, any finite energy signal f can thus be represented by its wavelet inner-product coefficient

$$\langle f, \Psi_{j,n} \rangle = \int_{-\infty}^{+\infty} f(t) \Psi_{j,n}(t) dt \tag{7.4}$$

and can be recovered by summing them in this wavelet orthonormal basis:

$$f = \sum_{j,n\in\mathbb{Z}^2} \langle f, \Psi_{j,n} \rangle \Psi_{j,n} \tag{7.5}$$

The systematic approach towards creating orthonormal wavelet bases was established by Meyer and Mallet [149]. Wavelets can be continuous or discrete. The continuous wavelets are given by multi resolution signal approximations.

$$CWT(a, b) = \int_{-\infty}^{+\infty} x(t) \Psi^*_{a,b}(t) dt, \tag{7.6}$$

where $x(t)$ is the analyzed signal, a and b represent the scaling factor (dilatation/compression coefficient) and translation along the time axis (shifting coefficient), respectively,

and the superscript asterisk $(*)$ denotes the complex conjugation. $\Psi_{(.)}$ is obtained by scaling the wavelet at time b and scale a:

$$\Psi_{a,b}(t) = \frac{1}{\sqrt{|a|}}\Psi\left(\frac{t-a}{b}\right) \tag{7.7}$$

where $\Psi(t)$ is the mother wavelet. Since the windows are adaptable to the transient of each scale, wavelets address the issue of application to non-stationary signals.

Representation of a signal can be optimized by creating a signal model that would carry a prior information. To this end, we can model a signal f as a realization of a random process F, whose probability distribution is known as a priori. Then we try to minimize the expected approximation error through a Bayesian approach. Linear approximations because of their dependence only on covariance are simpler. On the other hand, the non linear approximations require the full probability distribution of F to be known which is not so often the case for music or images because the transient structures are not properly modelled as realizations of known processes such as Gaussian ones. However, to optimize the non linear representations, a weaker but deterministic model can be approached. A deterministic model would specify a set Θ, where the signal would belong and the set is defined by any prior information e.g., time-frequency localization of the transients in ECG signals.

7.2.1 Discrete Wavelet Transformation (DWT) for Dimensional Embedding

For the reader's convenience, we recall a few well-known definitions and theorems wavelet theory following [61] and [150] and using the notation from [248]. This DWT was used in Chapter 6 in front of the transformer to enhance its feature extraction capability.

The crucial idea is to decompose the space $L^2(\mathbb{R})$ into resolution spaces of different resolutions. First, the *resolution space* V_0 is defined as the space of piece wise constant functions on subintervals of $[n, n+1]$ with $n = 0, \ldots, N$. If we define the corresponding step function

$$\phi(t) = \begin{cases} 1, & if \quad 0 \leq t < 1 \\ 0 & otherwise, \end{cases} \tag{7.8}$$

then V_0 has dimension N, and the N functions $\boldsymbol{\phi}_0 := \{\phi(t-j)\}_{j=0,\ldots,N-1}$ constitute an orthogonal basis. Analogously the *refined resolution spaces* V_k are defined as the spaces of functions constant on each sub-interval $[n/2^k, (n+1)/2^k]$. This yields a nested

sequence of embedded spaces

$$V_0 \subset V_1 \subset V_2 \subset \ldots \subset V_k. \tag{7.9}$$

We denote the orthogonal complements of V_{k-1} in V_k as W_{k-1}, i.e., $V_k = V_{k-1} \oplus W_{k-1}$ and call it the *detail space*. This yields an orthogonal decomposition at level k as follows:

$$V_k = V_0 \oplus W_0 \oplus W_1 \oplus \ldots \oplus W_{k-1} \tag{7.10}$$

Then the k-level *discrete wavelet transformation* (DWT) is defined as the change of coordinates from $\boldsymbol{\phi}_k$ to $(\boldsymbol{\phi}_0, \boldsymbol{\psi}_0, \boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_{k-1})$, where $\boldsymbol{\phi}_k := \{\phi_{jk}\}_{j=0,\ldots,N-1}$ and $\boldsymbol{\psi}_k := \{\psi_{jk}\}_{j=0,\ldots,N-1}$, resp. denote the family of functions obtained from the mother wavelet.

This yields a filter bank interpretation of DWT, wherein each step the signal is decomposed into an averaged and a detailed signal using low-pass and a high-pass filter depicted graphically in Fig. 7.1.

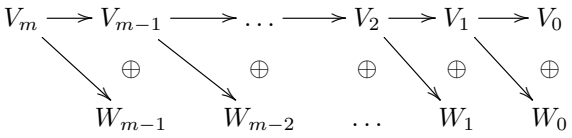


FIGURE 7.1: Wavelet Transform Pyramid

Thus, any signal can be decomposed into an averaged and a detailed signal, namely V_0 and W_0. Due to the recursive nature, this can be extended to any desired level k. Please note, that due to the dyadic nature the data size is reduced by a factor of 2 in each step. For further details, we refer to [61] and [150] as well as e.g. [248].

We used Haar and Daubechies wavelets as well as symlets (symmetrized version of Daubechies wavelets) as mother wavelet.

7.3 Wavelets as Feature Extractors

Wavelets have been used as feature extractors in many domains see [266],[247], [152], [56], [275]. We propose and experimentally show as part of this study in the next Chapter 8, that using wavelet transformation is the efficient way to adapt the dimensional embedding part of the transformers for time series classification.

Section 7.3.1 - 7.3.4 summarizes ideas from the pre-print from Schäfer [210] in order to lay the mathematical foundation for the argument of using wavelet transformation as feature extractors for time series.

7.3.1 Groups and Group Representations

In physics one of the most powerful principles to study the laws of nature are groups and their representations which are related to invariant.

A group G is a set G together with a binary operation on G denoted $\cdot$ (called multiplication) satisfying

- (Associativity) For all a, b, b in G we have $(a \cdot b) \cdot c = a \cdot (b \cdot c)$.
- (Identity) There exists an element $e \in G$ such that for all $a \in G$ we have $e \cdot a = a \cdot e = a$.
- (Inverse) For all a in G there exists an element $a^{-1} \in G$, s.t. $a \cdot a^{-1} = a^{-1} \cdot a = e$.

Below, we give three important examples of groups that are relevant in the context of time series.

Definition 1. The translation group T is the group $(\mathbb{R}, +)$ where the $\cdot$ is identified with the usual addition.

Definition 2. The dilatation group D is the group $(\mathbb{R}^+, *)$ where the $\cdot$ is identified with the usual multiplication.

Definition 3. The a, b group G_{ab} is the semi-direct product of the previously defined translation and dilatation group, i.e., the following sub group of 2 x 2 matrices defined with the usual matrix multiplication:

$$G_{ab} = \left\{ \begin{pmatrix} a & b \\ 0 & 1 \end{pmatrix}, a \in \mathbb{R}^+, b \in \mathbb{R} \right\}. \tag{7.11}$$

The group axioms are satisfied trivially with the group multiplication computed as $(a, b) \cdot (a', b') = (aa', ab' + b)$.

As in any category homomorphisms and isomorphisms are important.

Remark 7.1. We note that translation and dilatation group correspond to the $(\mathbb{R}, +)$ and $(\mathbb{R}^+, *)$ and we have a group isomorphism mediated by the natural log or exp.

One particularly important (morphism) concept is the concept of a group representation:

Definition 4. A *representation* ρ of a group G on a vector space V over a field K is a group homomorphism from G to $GL(V)$, denoting the general linear group on V, i.e., the linear transformations on V. Henceforth, a representation is a map

$$\rho : G \to GL(V), \text{such that} \tag{7.12}$$

$$\rho(g_1 g_2) = \rho(g_1)\rho(g_2), \tag{7.13}$$

i.e., ρ has to preserve the structure (morphism). We call V the representation space and its dimension the dimension of the representation.

If G carries a topological structure and we have therefore some notion of continuity we can define continuous representations as follows:

Definition 5. Let G be a topological group, then we call a continuous representation of G on a topological vector space V a representation ρ such that the map $g, v \to \rho(g)(v)$ is continuous (in product topology).

Vector spaces with a scalar (inner) product induce a norm and metric. Particularly nice such spaces are Hilbert spaces which are complete w.r.t. norm[1]. Therefore, if V carries the structure of a Hilbert space, we denote $H = V$ and we can define further:

Definition 6. Let G be a topological group. A strongly continuous unitary representation of G on a Hilbert space H is a group homomorphism ρ from G into the unitary group $U(H)$ of H

$$\rho : G \to U(H), \tag{7.14}$$

such that $g \to \rho(g)x$ is a norm continuous function for every $x \in H$.

(Here we denote by $U(H)$ the unitary transformations, i.e those for which $\langle Ux, Uy\rangle = \langle x, y\rangle$ holds for all x , $y \in H$.)

Recall that we model time series $x_t := x(t)$ as elements of $\mathcal{L}^2(\mathbb{R})$, i.e., $x_t \in \mathcal{L}^2(\mathbb{R})$. Now, $\mathcal{L}^2(\mathbb{R})$ as an Hilbert space carries a canonical topological structure via the induced norm (and metric).

The first two assertions are a trivial exercise using the fact that the Lebesgue measure is invariant (as the Haar measure) for translations and transforms with Jacobean $1/a$ under dilatation. The third is an immediate consequence of the first two claims.

[1] Only relevant for infinite dimensional spaces.

7.3.2 Quantization

While unitary representation on $\mathcal{L}^2(\mathbb{R})$ are certainly nice, for actual computations we need a *quantization* of the data, i.e., a mapping that discretizes the continuum. We denote this mapping by $Q_V : \mathcal{L}^2(\mathbb{R}) \to V$, where V is a finite dimensional vector space.

Definition 7. Let V be a finite dimensional vector space. A quantization of $\mathcal{L}^2(\mathbb{R})$ is a linear mapping Q_V from $\mathcal{L}^2(\mathbb{R})$ to V.

Traditionally for time series, quantization Q_V is achieved by sampling time at discrete intervals t_i, $i \in (1, \ldots, n)$, henceforth $Q_V(x_t) = (x_{t_1}, x_{t_2}, \ldots, x_{t_n})$. (Note as $x(t) \in \mathcal{L}^2(\mathbb{R})$ being defined only modulo equivalence of Lebesgue nulls sets, the notation $(x_{t_1}, x_{t_2}, \ldots, x_{t_n})$ has to be interpreted as representing a piece-wise constant step function.) The time steps are usually deterministic and of the same size (period) δ, i.e., $t_i = \delta * i$, but other (random) sampling / quantization schemes are used as well. In this paper, however, we do *not* require the quantization operator to be of this simple form, *any* mapping from $Q_V : \mathcal{L}^2(\mathbb{R}) \to V$ might be used [2].

7.3.3 Admissible Embeddings

In this section, we try to provide some properties that an admissible embedding should satisfy: We can provide the following reasons to justify this definition:

1. The embedding should not be affected by any translation as time is homogeneous.
2. The embedding should not be affected by any dilatation as scales are arbitrary (only the *ratio* between scales should matter)

Now if we require our embedding to have no effect for the variance i.e, it should display in-variance that $\Phi(\rho_{a,b}x_t) \equiv \Phi(x_t)$ is actually a heavy constraint onto the actual problem and hinders coming to any solution. But in reality it is not often required. All computations within attention mechanism revolve around inner product which ensures that an *equivariance* in the sense of a mapping into unitary operators would be sufficient to ensure that the network training is not affected.

Further evidence is that translation invariance without dilatation is impossible. Indeed, if G is a unimodular group (left-invariant measure m on G is right-invariant), having an admissible vector then G is necessarily discrete, see Proposition 0.4 in [79]. As $\mathbb{R}$ is unimodular the claim follows.

[2] In the sequel, as we will see, a composition of time sampling at discrete intervals and a projection (onto wavelet coefficients) is suggested and analysed.

Therefore, we believe the following definition is quite natural:

Definition 8. Let G_{ab} be the dilatation and translation group. An admissible embedding is a strongly continuous unitary representation of G_{ab} on Hilbert space V_i such that for each i there is a group homomorphism ρ_{V_i} from G_{ab} into the unitary group $U(H)$ of V_i and the associated diagrams commute.

Remark 7.2. One might argue, why requiring continuity (and in $\mathcal{L}^2(\mathbb{R})$ induced topology) is a canonical choice. We claim, that contrary to the situation in NLP we cannot expect any notion of closeness that is *not* compatible with the $\mathcal{L}^2(\mathbb{R})$ metric because time series result *always* from (physical) sensors i.e., physics with measurement error(s). Henceforth if signals are close to each other in the sense of being within the sensor error tolerance they have to be close in $\mathcal{L}^2(\mathbb{R})$. Therefore, if they would be semantically different–as could happen in an NLP setting–we would be lost anyway as these signals cannot be distinguished by our measurement devices.

7.3.4 Wavelets as Admissible Embeddings

Nowadays wavelets are defined via a multi scale analysis as introduced by [151]. We have an increasing sequence of subspaces V_n of $\mathcal{L}^2(\mathbb{R})$:

$$0 \subset \ldots \subset V_2 \subset V_1 \subset V_o \subset V_{-1} \subset V_{-2} \ldots \subset \mathcal{L}^2(\mathbb{R}) \tag{7.15}$$

so that

$$\overline{\bigcup_{m\in\mathbb{Z}} V_m} = \mathcal{L}^2(\mathbb{R}) \tag{7.16}$$

$$\bigcap_{m\in\mathbb{Z}} V_m = 0 \tag{7.17}$$

$$f(\cdot) \in V_m \Leftrightarrow f(2^m\cdot) \in V_0 \tag{7.18}$$

There exist scaling function $\phi \in \mathcal{L}^2(\mathbb{R})$ such that

$$V_0 = \overline{\text{span}\{\phi(\cdot - k)|k \in \mathbb{Z}\}}. \tag{7.19}$$

With the help of this scaling function one can construct wavelets from a mother wavelet Ψ , see [150] or [61]. The continuous wavelet transform is defined as follows:

$$L_\Psi x(a,b) := \frac{1}{\sqrt{c_\Psi}}|a|^{-\frac{1}{2}} \int_{\mathbb{R}} x(t)\Psi\big(\frac{t-b}{a}\big)\, dt, \tag{7.20}$$

where c_Ψ denotes an appropriate normalisation constant. It is well known that this constitutes an admissible, continuous representation of the $G_{a,b}$ group onto $\mathcal{L}^2(\mathbb{R})$. Henceforth if we define $\rho(ab)x(t) := L_\Psi x(a,b)x(t)$ we have an admissible representation in $\mathcal{L}^2(\mathbb{R})$.

In order to get representations on the finite dimensional subspaces V_i one discretises (quantises) the continuous wavelet transform. The usual discrete wavelet transform DWT, however, is not a faithful representation as it is *not* invariant under translation. The reason is a decimation applied for optimal data representation, where every second index is dropped, and henceforth the resulting coefficients are not preserved under translation. This has been known for a long time and as for some data analysis, the non invariance is problematic, remedies have been proposed. The most prominent one is the translation invariant *stationary wavelet transform* (SWT) or also called 'à trous' where translation-invariance is achieved by removing the down samplers and up samplers in the DWT and upsampling the filter coefficients by a factor of $2^{(j-1)}$ in the jth level of the algorithm, see [78]. The SWT is an inherently redundant scheme but useful in our context.

Definition 9. We define a wavelet embedding by setting $\rho(ab)x(t) := L_\Psi x(a,b)$ for $\mathcal{L}^2(\mathbb{R})$ and $Q_V x(t) := Q_W x(t) := (x_{t_1}, \ldots, x_{t_n})$, where the t_i correspond to the sampling as defined in SWT. This induces a representation of $\rho_n : V \to V$.

Lemma 7.3. *The wavelet embedding is admissible.*

Proof. The continuous part is well known. As in the SWT all coefficients are kept a translation is just a permutation matrix which is clearly in $U(\mathbb{R}^n)$. It is also known that the wavelet basis provide a basis for $\mathcal{L}^2(\mathbb{R})$ and that the coefficients converge in $\mathcal{L}^2$ norm. Henceforth we have continuity. □

Remark 7.4. There are many approaches to deal with non translation invariance that have been studied in the literature. For instance one might apply the continuous wavelet transform (CWT) with quantization *after* the wavelet decomposition, or use over complete discrete wavelet transform (OCDWT), or approximative shift-invariance by limiting the sub-band sub- sampling as the power shiftable discrete wavelet transform (PSDWT) or dual tree complex wavelet transform (DTCWT), see [29] for a good overview.

Remark 7.5. In the paper [28] the authors provide another intuition which supports our claim: "We begin with a piece of motivation. Consider a separable Hilbert space H evolving discretely over time through the action of a unitary U, and fix a countable collection of "sensors" $A = \{u_i | i \in I\}$ in H. Suppose we use these sensors to measure the evolutions of various $v \in H$, thereby obtaining the data $Tv = \{\langle U^k v, u_i\rangle | k \in \mathbb{Z}, i \in I\} = \{\langle v, U^{-k} u_i\rangle | k \in \mathbb{Z}, i \in I\}$. In general it is possible to stably reconstruct any $v \in H$ from Tv if and only if the system $E(A) := \{U^k u_i | k \in \mathbb{Z}, i \in I\}$ is a *frame* for H[1, 2].

One of the goals of this paper is to classify such systems." Frame on the other hand are deeply and naturally connected to Wavelet decomposition as any Wavelet basis provides a frame and non-orthogonal, redundant Wavelets still provide frames, see [61].

In the next chapter we perform a series of experiments on different time series and record the results to provide a basis for comparison of different dimensional embeddings as a form of pre processing.

Chapter 8

Dimensional Embeddings for TSC in Transformers

> "The key to artificial intelligence has always been the representation." – (Jeff Hawkins)

Many deep learning models have been used for both classification and forecasting of bio signals in recent times. One of the gravitating fields is Electrocardiogram classification for different heart conditions. Transformer models have continuously been shown to outperform their contemporaries in the field of natural language processing. However, they still have to be adopted carefully to obtain comparable results in the field of time series classification. In this study, we explore the effect of different dimensional embedding in time series classifications for the first time to the best of our knowledge. We use wavelet transformation; discrete, continuous, scattering, and feature maps from convolutional neural networks for performance comparison. We use two ECG data sets for both multi-class and binary classification. In all the experiments, it is shown that deploying relevant feature extraction techniques as dimensional embedding almost always outperforms a plain transformer.

8.1 Introduction

Transformers were introduced in 2017 by Vaswani et al.[249] in the seminal paper and have proven pioneering and impactful in the realm of natural language processing.

However, their main potential lies in drawing the dependencies between input and output using mainly attention techniques and allowing significantly more parallelization. This has been a huge advancement since recurrent neural networks.

Their utility in other deep learning fields like computer vision, time series classification and prediction has shown them to be effective and powerful across different domains. In this study, we present an overview for the first time, to the best of our knowledge, of the adaptation of the dimensional embedding layer for time series classification in the recent literature. In this study, we introduce Tsc-transformer, a simple transformer architecture for time series classification. The architecture replaces the dimensional embedding with feature extraction and surpasses a plain transformer in the performance with five data sets.

8.2 State of the Art and Our Contribution

In the previous chapters based on [37], an effort towards automatic feature extraction was made by introducing two algorithms. Both used one deep learning model and a feature extractor in front of it. One algorithm used CNNs in front of the LSTMs and the other algorithm used transformers as the base deep learning model and used wavelet transformations of the input time series as dimensional embeddings. The results were on par with a CNN-LSTM architecture, which makes the algorithm more computationally effective. For more details please refer to [37]. As a continuation of the previous study, we have explored the different feature extraction techniques as dimensional embeddings for both uni and multivariate time series and compared the performance with plain transformers and LSTMs and have found that in all experiments, proper feature extraction as dimensional embeddings is more effective in classification for multivariate and univariate time series as compared to transformers without any dimensional embeddings. Hence we have supplemented the dimensional embeddings with a feature extractor. We have also designed an ablation study to verify whether positional encoding is required or not in the case of wavelet transformations as localization information is already inserted in the data. Thus, the following hypotheses are formulated:

Hypothesis 1: Positional encoding does not play any vital role in TS classification if we already have a wavelet transformation representation of our TS.
Hypothesis 2: For a given time series, if an appropriate wavelet representation is achieved, the classification task can be performed by a simple classifier.

Hypothesis 2 was first proposed by Andén and Mallat in [11], where scattering wavelet transformation was applied to music and phone genre time series, and state-of-the-art results were obtained using a SVM.

8.2.1 Transformers and Attention Mechanism

The following definitions have been adapted from Sabeen et al. [5]. More formal definition for transformer layers can be found in [184] and [241].

The foundation of transformer architecture is finding a connection or association between multiple inputs sequences after performing dot product. Let $\{\mathbf{x}_i\}_{i=1}^{n}, \mathbf{x} \in \mathcal{R}^d$ be n data point in a sequence. The subscript i represents the position of the vector or the position of the word in the original sentence or word sequence. The weighted dot product of these input vectors with each other is known as the self-attention operation [5], which can thought of as a two-phase process. The initial phase involves computing a normalized dot product for each pair of input vectors within a specified input sequence. This normalization process utilizes the softmax operator, which adjusts a set of values to ensure that the resulting numbers collectively add up to one.

The normalized correlations are calculated between an input segment $\mathbf{x_i}$ and all others $j = 1, ..., n$

$$w_{ij} = Softmax\left(\mathbf{x}_i^T \mathbf{x}_j\right) = \frac{e^{\mathbf{x}_i^T \mathbf{x}_j}}{\sum_k e^{\mathbf{x}_i^T \mathbf{x}_k}}, \tag{8.1}$$

where $\sum_{j=1}^{n} w_{ij} = 1$ and $1 \leq i, j \leq n$. In the second step, a new representation $\mathbf{z_i}$ is found for a given input segment $\mathbf{x_i}$, which is a weighted sum of all input segments.

$$\mathbf{z_i} = \sum_{j=1}^{n} w_{i,j} \mathbf{x}_j, \quad \forall \quad 1 \leq i \leq n. \tag{8.2}$$

In Equation 8.2, it can be seen that for any input segment $\mathbf{x_i}$, the weights $\mathbf{w_{ij}}$ would add up to 1. This implies that the resulting vector $\mathbf{x_i}$ would be similar to the input vector with the largest attention weight w_{ij}. The attention weight which is largest is the result of the greatest correlation value which results as a normalized dot product between $\mathbf{x_i}$ and $\mathbf{x_j}$.

To obtain self-attention operation, three linear weighted vectors are built from the input $\{\mathbf{x}_i\}_{i=1}^{n}$. They are called, Query: $\mathbf{q} \in \mathbb{R}^{s_1}$, key $\mathbf{k} \in \mathbb{R}^{s_1}$ and values $\mathbf{v} \in \mathbb{R}^{s}$ which can be calculated as follows:

$$q_i = W_q x_i, k_i = W_k x_i, v_i = W_v x_i \tag{8.3}$$

where W_q and $W_k \in \mathbb{R}^{s_1 \times d}$, $W_v \in \mathbb{R}^{s \times d}$ are learnable weight matrices. The output vectors $\{\mathbf{z}_i\}_{i=1}^{n}$ are given by:

$$Z = \sum_j \text{softmax}\left(q_i^T k_j\right) v_j, \tag{8.4}$$

The weight of the vector $\mathbf{v}_i$ is dependent on the correlation between the query vector $\mathbf{q}_i$ at position i and the key vector $\mathbf{k}_j$ at position j. The dot product tends to grow in value with the increasing size of the query and key vectors. Since the softmax operation is known to be sensitive to large values, the attentions weights are scaled by the square root of the size of the query and key vectors d_q as shown in

$$Z = \sum_j \text{softmax}\left(\frac{q_i^T k_j}{\sqrt{d_q}}\right) v_j, \tag{8.5}$$

It can rewritten in matrix form as follows:

$$Z = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V, \tag{8.6}$$

where Q and $K \in \mathbb{R}^{s_1 \times n}$, and $V \in \mathbb{R}^{s \times n}$, $Z \in \mathbb{R}^{s \times n}$ and ^{T} represents the transpose operation.

8.3 Dimensional Embeddings

In the field of NLP, for preserving the notion of 'similarity', the tokens $x \in \mathcal{X}$ (where $\mathcal{X}$ denotes a *finite* set, i.e., a language) need to be mapped to vector space $E \simeq \mathbb{R}^n$. To this end, usually, a dot product is applied and then the mapping $\Phi : \mathcal{X} \to \mathcal{E}$ is designed by the domain experts to formulate a dictionary based on statistical techniques or other word embeddings e.g. the famous GloVe embedding [181] and references therein. However, for time series we face the following challenges [211]:

1. Since the time in time series is continuous and henceforth input space is infinite dimensional which cannot be represented naturally.
2. The notion of position in sentences is not exactly similar to the time stamps in time series. The order is important but it is not analogous to the position of the word in a sentence. This brings into light the role of positional encoding in a time series transformer.
3. Time series can be both univariate and multivariate. Transformers have been proven experimentally to work better with multiple dimensions as described in

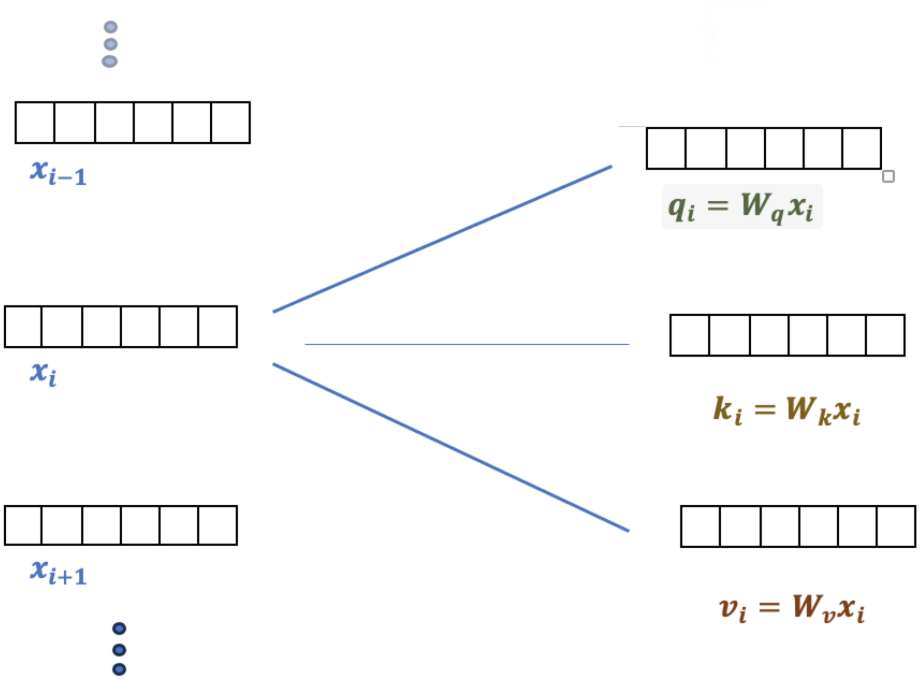


FIGURE 8.1: Input vector of length $d = 6$ is shown along with the linear weighted transformations. The resultant vectors, each of size $s_1 = s = d$ are called Query, key and value. In every sequence handled by the Transformer, there exist n inputs in total, leading to the generation of n query vectors, n key vectors, and n value vectors.

[272]. This is why transformers may not function fully optimally with univariate time series but this requires more investigation.

To this end, we propose some modifications for dimensional embeddings for usage for classification in time series. An ablation study is designed to observe the effect of Positional encoding and dimensional embedding on time series analysis for classification. We only focus on the classification - both binary and multi-class. Data sets used for the experiments were largely but not limited to ECG signals. In addition to ECG, CSI dataset and UCR data sets were also used.

8.3.1 Transformers and Embeddings for Time Series

Transformers have achieved state-of-the-art performance in not only representing the natural language but also several sequence generation domains such as music as in Zerveas et al. [272]. Transformers have also been used for time series classification, forecasting, and generation. However, almost all the studies that use the transformers for time series have to modify or adapt the dimensional embedding part of it since time series are been dealt with. An overview on the transformers for time series is presented in Wen et al.

[258]. Sabeen et al. [5] also presents a comprehensive overview, challenges and how other studies have adopted transformers for time series learning both in field of forecasting and classification.

In the field of NLP, learning the numerical vector representations called embeddings is a major area of research and development. In a pioneering paper wav2vec [18], huge quantities of text are used to learn the vector representations. This has followed significant improvements in the domain of embedding related works. Many recent developments have made NLP field learning tasks on par with that of computer vision.

In [144], a gated transformer was introduced for multivariate time series. In this study, dimensional embedding was replaced with a fully connected layer. Similarly in [271], an observational embedding is used for satellite image time series (SITS). Observational embedding is a concatenation of positional encoding of the time series and projection of the input into higher dimensions by using a linear dense layer.

For the ECG classification, many studies have used convolution operation in one form or another for dimensional embeddings in transformers. [171] uses a a wide and deep transformer network for the classification of ECG and uses convolution operations applied to the original waveform to capture the latent space representation of the signal. Similarly [164],[106], [87] uses CNN feature maps as dimensional embeddings for ECG classifications. Table 8.1 presents an overview of the recent studies that use transformers

Paper	**Type of DE**
Zerveas et. al [272]	Each time stamp is linearly projected to a vector of same dimensionality as the internal representation
Sercan et. al [13]	Self supervised
Shankaranrayana et. al [219]	CNN feature blocks
Cai et. al [42]	GNN
Liu et.al [144]	Positional embedding to connected layer followed by non linear function (tanh)
Song et. al [228], Yan et. al [264]	1–D convolution
Yuan et. al [271]	observational embedding layer which is by concatenating dense layer and PE
Natarajan et.al [171]	Convolution operations
Meng et. al[164], Hu et. al [106], Guan et. al[87]	CNN feature maps

TABLE 8.1: An overview of the current literature for time series classification and the corresponding adaptation of dimensional embedding

for the time series classification and the way they adopt dimensional embedding layers to the architecture.

All of the studies mentioned use positional embedding after adopting the corresponding dimensional embedding. Positional encoding can be absolute or relative. In the original paper, an absolute positional encoding was introduced. Since then, many papers have introduced many different relative and absolute embeddings. One such absolute and relative positional encoding was introduced in [77] for multivariate time series classification. According to [272], positional encoding does not interfere with the numeric information of the time series in a similar manner to that of the word embedding because positional encodings are learned to occupy an approximately orthogonal subspace to the one where the projected time series samples reside, and this condition is easier to satisfy in high-dimensional spaces. This is why we hypothesize that if our dimensional embedding already has information related to localization e.g. in case of wavelet transformation, positional encoding may not add any additional information and we can remove it as well.

We propose to the best of our knowledge for the first time, the usage of wavelet transformation as dimensional embeddings for time series classification and we propose that it achieves on par, if not better, results with those of using convolution operation before transformers.

8.4 Proposed Architecture and Experimental Setup

The traditional transformer model starts with a learned embedding which translates sentences into meaningful vectors and then adds them into positional encoding. The two blocks, encoder, and decoder are applied. The encoder part learns the data layout through the attention mechanism and the decoder tends to learn the generation of the similar data with a masking mechanism in attention.

Our model (see Fig.8.2) consists of two components:

The first component is a deterministic dimensional embedding which is pre-calculated from the input data. This dimensional embedding is later on concatenated with positional encoding to fix the positions. The second component is the encoder part of the transformer which includes attention and feed forward module. Since our target is learning of the classifiers for ECG signals, we do not use the decoder part of the transformer. It was also tested and adding decoder for classifying did not make any significant difference in the results. That is why, to keep the model as simple and efficient as possible, the decoder part was also omitted. At the end we have a Softmax layer which is adjusted for the classification task. Here different kind of DE have been compared. Discrete wavelet transformation and scattering wavelet transformation have been used.

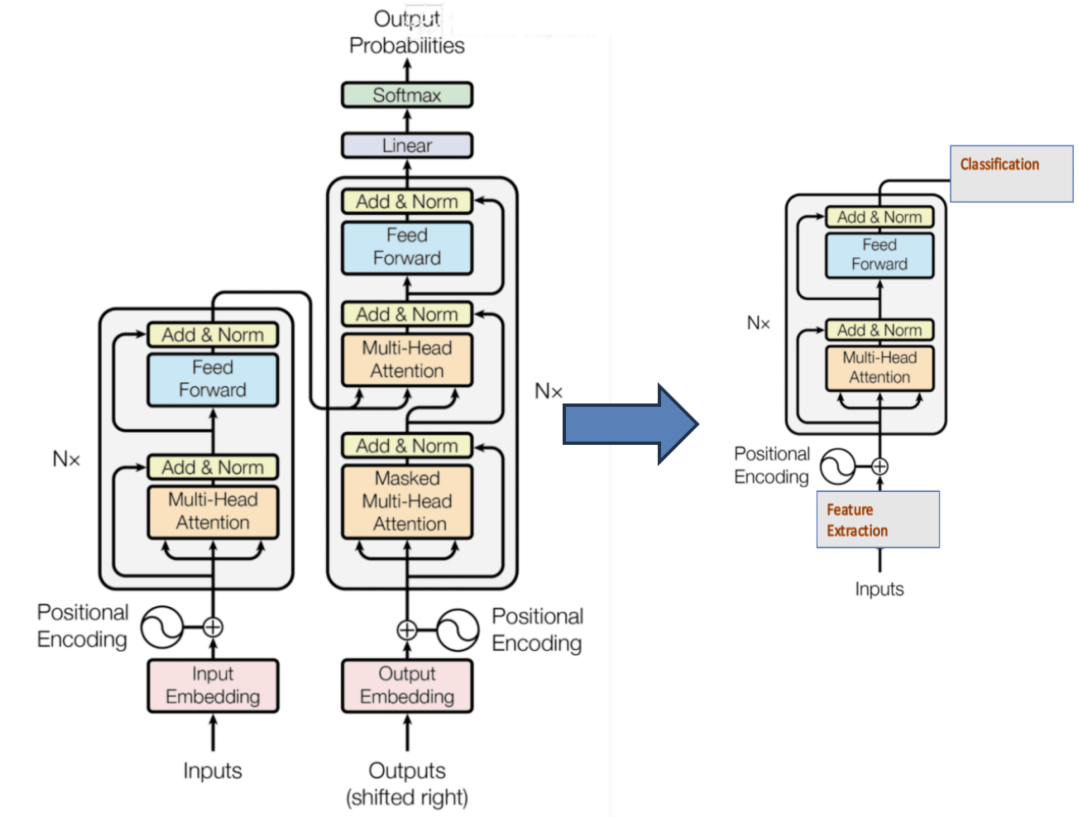


FIGURE 8.2: The proposed transformer architecture for classification as compared to an original transformer architecture [249]

```
=================================================
Layer (type:depth-idx)                   Param #
=================================================
LSTM                                     --
├─LSTM: 1-1                              740,352
├─Linear: 1-2                            80,647
=================================================
Total params: 820,999
Trainable params: 820,999
Non-trainable params: 0
=================================================
```

```
=================================================
Layer (type:depth-idx)                   Param #
=================================================
TransformerModel                         --
├─CNN: 1-1                               --
│    └─ReLU: 2-1                         --
│    └─Sequential: 2-2                   --
│    │    └─Conv2d: 3-1                  5,824
│    │    └─BatchNorm2d: 3-2             128
│    │    └─ReLU: 3-3                    --
│    └─Sequential: 2-3                   --
│    │    └─Conv2d: 3-4                  4,160
│    │    └─BatchNorm2d: 3-5             128
│    │    └─ReLU: 3-6                    --
│    │    └─Dropout: 3-7                 --
│    │    └─MaxPool2d: 3-8               --
├─TransformerEncoder: 1-2                --
│    └─ModuleList: 2-4                   --
│    │    └─TransformerEncoderLayer: 3-9 36,310
│    │    └─TransformerEncoderLayer: 3-10 36,310
│    │    └─TransformerEncoderLayer: 3-11 36,310
│    │    └─TransformerEncoderLayer: 3-12 36,310
├─Dropout: 1-3                           --
├─Linear: 1-4                            358,407
=================================================
Total params: 463,967
Trainable params: 463,967
Non-trainable params: 0
=================================================
```

FIGURE 8.3: The LSTM model(left) and Transformers(right) with feature maps as dimensional embeddings

Algorithm 5: Classification for time series with TTS

Input: A time series raw data Ts

Output: The classified activity label l

$WTS \leftarrow WAVELET\_TRANSFORMATION(Ts)$

$ATTEN \leftarrow WTS + MULTIHEAD\_ATTENTION\_MECHANISM(WTS)$

$FC \leftarrow FULLY\_CONNECTED(ATTEN)$

$l \leftarrow SOFTMAX(FC)$

return l

Another DE tested was feature maps, which were obtained by performing 2-D convolutional operation between raw TS and filters. This yielded 2-D feature maps which were then fed into the transformer layer of the encoder as shown in Fig 8.3. For all the datasets, LSTM and plain transformer were also used to establish base results.

```
Layer (type:depth-idx)                   Param #
==========================================================
MyTransformerModel                       --
├─PositionalEncoding: 1-1                --
├─TransformerEncoder: 1-2                --
│    └─ModuleList: 2-1                   --
│    │    └─TransformerEncoderLayer: 3-1  28,577,140
│    │    └─TransformerEncoderLayer: 3-2  28,577,140
│    │    └─TransformerEncoderLayer: 3-3  28,577,140
│    │    └─TransformerEncoderLayer: 3-4  28,577,140
│    │    └─TransformerEncoderLayer: 3-5  28,577,140
│    │    └─TransformerEncoderLayer: 3-6  28,577,140
│    │    └─TransformerEncoderLayer: 3-7  28,577,140
│    │    └─TransformerEncoderLayer: 3-8  28,577,140
├─Linear: 1-3                            913,507
==========================================================
Total params: 65,977,587
Trainable params: 65,977,587
Non-trainable params: 0
==========================================================
```

```
Layer (type:depth-idx)                   Param #
==========================================================
TransformerModel                         --
├─TransformerEncoder: 1-1                --
│    └─ModuleList: 2-1                   --
│    │    └─TransformerEncoderLayer: 3-1  17,638
│    │    └─TransformerEncoderLayer: 3-2  17,638
│    │    └─TransformerEncoderLayer: 3-3  17,638
│    │    └─TransformerEncoderLayer: 3-4  17,638
│    │    └─TransformerEncoderLayer: 3-5  17,638
│    │    └─TransformerEncoderLayer: 3-6  17,638
│    │    └─TransformerEncoderLayer: 3-7  17,638
│    │    └─TransformerEncoderLayer: 3-8  17,638
├─Dropout: 1-2                           --
├─Linear: 1-3                            954
==========================================================
Total params: 142,058
Trainable params: 142,058
Non-trainable params: 0
```

FIGURE 8.4: The transformer model with(left) and without(right) positional encoding and attention mechanism

8.5 Datasets

The datasets used are majorly from the domain of ECG signals. For this, 3 datasets from the famous time series data mining UCR dataset [60] have been used. We have also used a multivariate dataset from channel state information(CSI) from the human activity recognition domain. The major use cases have been Human activity recognition using ECG and CSI and classification of ECG signals in different health conditions. Table 8.2, presents a brief overview of each dataset. More information on the datasets can be looked up in Appendix B. The CSI dataset, uses human activity recognition as the use case and it has 7 classes namely, EMPTY, LYING, SIT, STAND, SIT-DOWN, STAND-UP, WALK, and FALL. For more information on data acquisition and experimental set up please refer to Schäfer et al [215]. Each dataset is further described in reference table 8.2

Name	Complete name	Features	Classes	Benchmark Accuracy	Our Accuracy
ECG1	PTB	1	2	99.99%[37]	99.97%
ECG2	UCR ECG 5000	1	5	98%[182]	96.40%
ECG3	UCR ECG 200	1	2	91% [89]	94%
ECG4	UCR fetal Thoarax	1	42	NA	92.56%
ECG5	HAR Dataset	1	3	100% [37]	100%
CSI	Channel State Information	90	7	97-100 %[215]	97- 100%

TABLE 8.2: An overview of the datasets used for the experiments. NA refers to not available

8.6 Results

Table 8.3 presents an overview of the obtained results. The recorded results are validation accuracy over 500 epochs. The chosen hyper-parameters are learning rate as 0.03 and 0.003, and the optimizer used is Adams optimizer. All the experiments were done on the GPU device NVIDIA A100-PCIE-40GB.

It can be seen that transformers with a feature extraction embedding almost always perform better than a plain transformer for classification. It is always the models with wavelet transformation who perform better than the ones without it. We can always see that in wavelet transformation there is no significant difference between the transformers with and without positional encoding. This is in line with the proposed hypothesis that if we have wavelet transformation as the chosen dimensional embedding then we already have the information of the position inserted in the data via the wavelets. Removing positional encoding should not affect the results as it would act as redundant information.

We also surpass in the benchmark for all the datasets except for ECG200 using wavelets as dimensional embedding.

Dataset	WT + SVM	LSTM	SW + LSTM	Plain Trans		Scattering + Trans		FM + Trans		DWT+ Trans	
				WoP	WP	WoP	WP	WoP	WP	WoP	WP
ECG1	95.09%	98.56%	99.18%	82.34%	82.61%	92.37%	90.45%	90.79%	91.55%	**99.52%**	**99.52%**
ECG2	93.95%	96%	94.58 %	93.95%	95.2%	92.76%	93.20%	94.53%	94.09%	96.00%	**96.40%**
ECG3	75%	92%	80%	70%	93.96%	86%	84%	94%	94%	86%	**94%**
ECG4	92.83%	92.22%	91.11%	91.11%	91.67%	92.22%	90.56%	86.44%	86.11%	**92.56%**	91.33%
ECG5	72.65%	87.40%	100%	68.70%	70.08%	98.43%	98.43%	81.10%	80.31%	97.64%	83.46%
CSI	90.22%	87.8-92.7%	95.12%	95.12%	95.12%	97.56%	**100%**	**100%**	100%	70.73%	78.05%

TABLE 8.3: The results for different datasets using different configurations for dimensional embedding with transformers. Key: WoP: Without Positional Encoding, WP: With Positional Encoding, Trans: Transformer, WT: wavelet transform, SW: scattering wavelet, FM: Feature maps, DWT: discrete wavelet transform

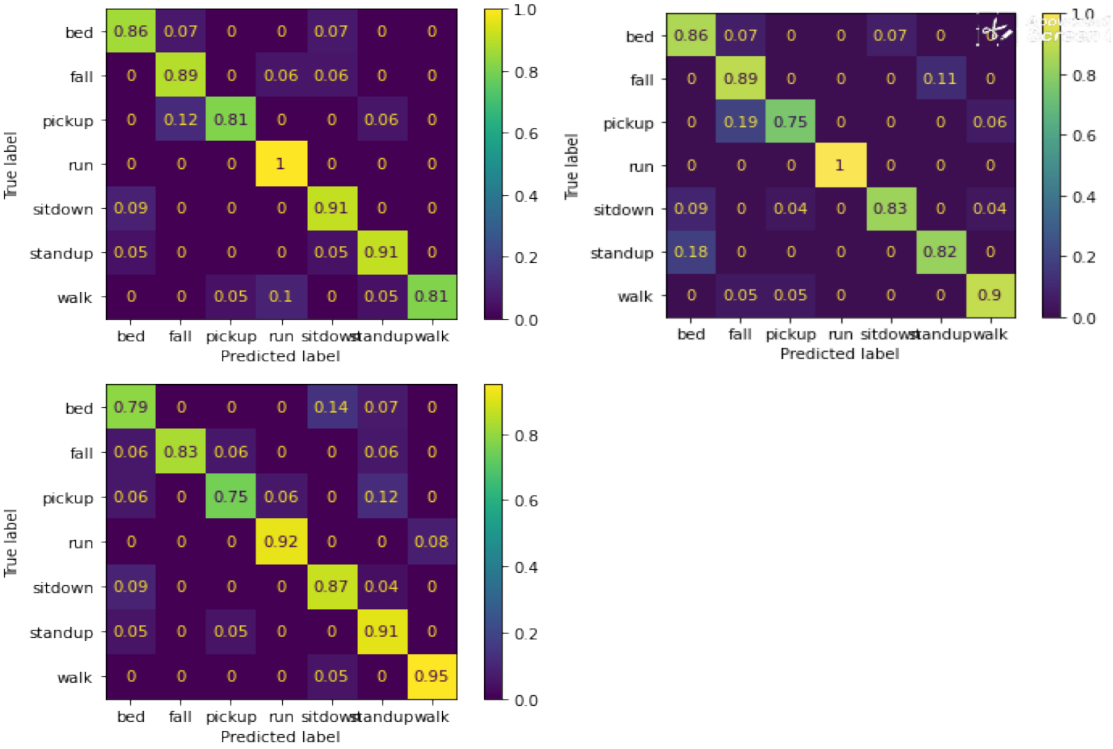


FIGURE 8.5: CSI dataset : The normalized confusion matrix for experiments with (top right) Transformer with feature maps, (top left)transformers with scattering wavelets , (bottom right)scattering wavelet transformation with LSTM

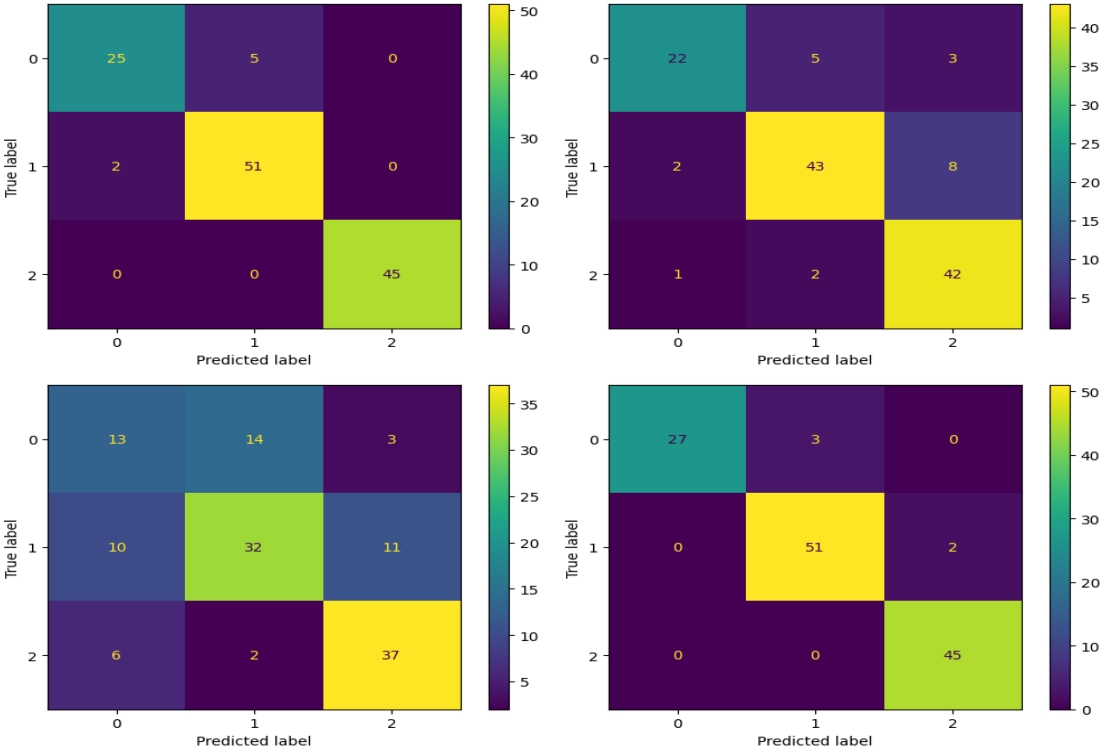


FIGURE 8.6: HAR dataset : The normalized confusion matrix for experiments with (top right) Transformer with scattering wavelets, (top left) Plain LSTM , (bottom right)Transformer with feature maps and (bottom left) Scaterring wavelet with LSTMs

8.7 Discussion

Positional encoding preserves the localization details before the attention operation. If we have another localization preserving mechanism like Fourier or wavelet transform, positional encoding could be omitted. It can be seen in the results as well that PE is mostly vital in vanilla transformer cases but with any kind of wavelet transform, with discrete or scattering, removing PE does not affect the end results.

Another vital observation is that the performance of SVM is not so far from the one with attention mechanism which leads to the hypothesis that 'is attention really all we need for time series classification?'.

Experimentally we can see that with proper representation, a linear classifier performs as well as a simple attention operation. This can be explained by the fact that classifiers in time series are generally looking for certain patterns and structures and if the right representation is achieved, a linear classifier like SVM can get as good performance as complex and deep transformers or LSTMs. This would also be a positive measure towards explainable AI as it is relatively simpler to explain linear classification than by deep learning methods. This in turn helps AI to be used over medical and other safety-critical systems. Whether this behaviour is prevalent in higher dimensional and across more complicated time series datasets is a matter for further investigation.

Another important reason for wavelets to work so well with transformers is that they tend to project the time series to higher dimensions which transformers tend to work well with.

8.8 Future Work and Conclusion

This study provides an overview of the effect of different dimensional embeddings concerning the TSC problem. We also proposed that for transformers to be effectively adapted for TSC, a relevant representation of time series is required. To this end, we proposed to obtain this representation as part of the dimensional embedding. We have laid the foundation and experiments in this study to confirm that using wavelet transformation extracts the relevant features of time series at earlier stages encoding the important information, which facilitates the usage of the transformer efficiently for TSC.

The reason to the keep the transformer model minimal without compromising the results was to make it adaptable for the next step after classification – Explainability. If we have a deep learning model, which is simpler with equivalent performance, preference should always be given to the simpler one following Occam's razor. As stated earlier, it is also

essential for models to be adaptable for explainability. Simpler models are a way forward toward explainable AI because intermediate variables that are to be explained and interpreted are in manageable numbers. This is not true for complex layered architectures which hinders the understanding of the decision. Explainablity and interpretability have not been explored in this study because of the limited time framework of the doctoral studies. However the pathway for it has been laid during this study so that the models can be explored efficiently.

Chapter 9

Findings with Respect to the Research Questions

> "That which can be asserted without evidence can be dismissed without evidence." – (Hitchens's razor)

The objectives described in Chapter 1, paved the path for automatic feature extraction techniques by performing the deeper analysis of time series with deep learning techniques. These objectives helped us to design precise research questions that were studied in the duration of this Ph.D study. In this chapter, each of the research questions presented in Chapter 1 is discussed in detail.

9.1 Are existing deep learning methods for time series classification as effective as they are for other domains like Natural language processing (NLP) and computer vision?

We discussed in Chapter 6 that the traditional deep learning models for time series classification do not always work well without adequate feature extraction mechanism. RNNs were designed for time series by capturing long term temporal dependencies, but they do not always work substantially well for classification. We have addressed this RQ in a different chapter with the help of current literature that for TSC, we need to have an appropriate representation of the time series which can be both domain dependent or independent for better prominent performance in classification tasks.

9.2 What are the hindrances in the actual realization of Industry 4.0 despite the availability of modern data analysis techniques?

This RQ has been observed and addressed practically during the length of the industrial collaboration. Many obstructions lay in between the start of the project and the actual application of the data science. One of the industrial use cases is described in detail in Chapter 3. This chapter highlights the shortcomings and limitations of the current process model to follow for the application of data science to industrial projects. Actual hindrances mentioned in the chapter are the lack of a viable use case, lack of relevant data and lack of role descriptions in current state of the art process models to name a few. Suggestions are put forward later in the chapter to address these issues and how to improve the process models so that better practices can be adapted to take advantage of the technology to handle industrial use cases. The findings of investigating this research question were published in [40].

9.3 How can data science be effectively used in an industrial setup to decrease production downtime?

Data science application in industrial setup is not as straightforward as it could have been. Hence, to address this RQ, we looked into the current practices and chose the most commonly used data processing model for industrial collaboration. During the collaboration, we discovered and confirmed some of the flaws not only in the process model but also in the actual practises of the industry.

Many industrial projects and prospective collaborations could not be realized in our research group due to these practises which include but are not limited to conservative personnel, hesitance to share complete data sets or domain information despite of NDAs in place, and not applying thorough domain knowledge to the ML models and relying solely on the data based learning. Chapter 3 and the publication [40] detail out different courses of action that can be taken to effectively apply data science in industrial setups. This includes working out a commercially viable use case, determining the availability and quality of data as suitable for the use case at the early stages, and including domain knowledge to the traditional DS and ML methods.

9.4 How features could be automatically extracted from different time series for classification problems?

To this end, Chapter 6 proposed two algorithms that avoid manual feature extraction and can extract features automatically which are characteristic of a certain signal. For automatic feature extraction, we proposed situating mathematical operations in front of deep learning models which would focus on only the relevant traits of a certain class. The chosen operations were convolutional operation and discrete wavelet transformation to be placed in front of LSTM and transformers respectively. This has shown experimentally to improve the performance of the models and have exceeded the benchmark for many datasets. This question was addressed and discussed in detail in the journal publication [37].

9.5 Can human activity including fall be detected from ECG signals?

This research question was initially discussed and presented in the master thesis [33]. However, the experiments were later augmented with publicly available ECG datasets for human activity recognition. The datasets were then resampled, and their DWT was obtained to create 3-D scalograms. Then the scalograms were used to train two deep pre-trained neural networks. The results verify the initial results and the fall could be distinguished distinctly from other human activities using ECG signals with an accuracy of 98%. Chapter 5 and subsequently [34] describe the experimental setup, the data pre–processing, model fine-tuning, training, and testing in detail.

9.6 Can wavelet transforms act as feature extractors for time series to be used in deep learning models?

This was analysed in detail in Chapter 6, Chapter 7 and in Chapter 8. The properties of wavelets that make them suitable feature extractors for time series are detailed in Chapter 7. The adept use of wavelet transformation is also shown experimentally in Chapter 6 and in Chapter 8 with various ECG and other time series data sets. It has been shown systematically that the wavelets help to focus on the most important aspects of the classification patterns. In the majority of the experiments performed during this study which include different datasets, incorporating a wavelet transformation has improved the performance of the transformer model. However, which exact patterns are selected

by the models in the classification task is still an open research question and can be investigated further.

9.7 How can transformers be adapted to time series classification?

Chapter 6 initially looks into the addition of feature extraction for the adaptation of transformers to time series classification. Later Chapter 8 looks in detail into this research question. A comprehensive state of the art has been presented comparing how other related studies have adopted the transformers for time series classification. A novel dimensional embedding technique using wavelet transforms is proposed to capture temporal and positional dependencies which characterize a time series. It is also proposed that with DE incorporating localization details, positional embedding step of the transformer operation can be omitted for time series. The performance of transformers for time series has been experimentally compared with different DE consisting of different wavelet transformations and feature maps from convolutional operations.

Any form of time series classification via transformers would require adaptation of PE and DE in a way that preserves the underlying patterns and dependencies of the time series. State-of-the-art literature shows that many of the transformer adaptations for time series classification use a convolution layer. Though many do not state the reason for using it, in this study we have proposed and confirmed that adding a convolution layer as dimensional embedding extracts the relevant features and forwards them to the attention layer.

9.8 Does positional encoding play any vital role in TS classification, if we already have a wavelet transformation representation of our TS?

This research question was experimentally tested and explained in Chapter 8. Positional encoding is mainly required before the attention mechanism specifically to integrate the sequence's order. Notably, self-attention operations do not encode information regarding the sequential arrangement of input data. Since in our experimental setup, we hypothesized that if dimensional embedding techniques included wavelet transformation, they would already encode positional encoding by definition. Hence PE step can be removed and the data can be processed directly to the attention layer to achieve similar classification results. Experimental results of the ablation study show that removing PE does

not degrade the classification capability of the models which re affirms our hypothesis. Although these are still initial results and have to be verified using statistical techniques similar to those used in Chapter 6.

9.9 Challenges and Limitations

This section briefly overviews the challenges faced during the investigation and implementation of the research questions. The challenges consist of two broader categories: The ones that are inherent to the field and the others faced by the author in carrying out the thesis work. Although most of the challenges have been discussed with each chapter, here we discuss only the challenges that commonly envelop the research questions.

- The first challenge is associated with the availability and quality of the datasets both in industrial and academic setup. For industrial data sets, usage and implementation require strict nondisclosure agreements in place before any analysis or experiments can be done. This hinders and prolongs the research publication process. On the other hand, the open-access datasets are limited in their capabilities and quality. Many of them are far too altered and engineered in terms of processing, cleaning, and replacing empty fields.
- Another challenge in the field of deep learning is the lack of software engineering practices. This issue trickles straight from common implementation to testing to documentation. This is a well-documented problem in the field. According to a survey conducted by [218], the biggest challenge faced by software engineers in incorporating ML components with traditional software components is the testing and quality evaluation of those components. This is because the testing mediums and practices are not present as the ML components behave non-deterministically, or test coverage is hard to define.
- Imbalanced datasets are another very common problem found in the area of machine learning and deep learning. Particular challenges occur with multi-class datasets where heavy imbalance occurs between each class. Although methods exist to balance the datasets, however experimentally they are not always guaranteed to work and they take away the primary focus and effort from the actual classification issue one is trying to cater to. Another related issue is that no formal definition exists for the imbalance problem and therefore no formal threshold exists to define what imbalance means.
- The criteria to evaluate the deep learning models for classification varies across different application fields. This makes it challenging to objectively evaluate the

models for the performance without depending on the application field. One way to take it forward is to use the most commonly used criteria in the literature. Another method to strengthen your results obtained from the DL models is to use statistical verification methods as were used and explained in Chapter 6. Implementing these methods and making the results comparable requires reporting the data instances used for testing and training but also reporting the exact hyperparameters used for training the model. Unfortunately, this practice is still not widespread in the DL and ML community rather it gives researchers who implement such methods a competitive disadvantage as more than often it requires a significant amount of effort put into recording and evaluating the results and not relying on a single KPI. Although a straightforward solution does not exist, as this is something that has to be solved by the field or pioneers in favor of others by standardizing the practices as much as possible.

- Another vital challenge which is related to the previous point is the difficulty of reproducing the results obtained by different studies due to the lack of sharing exact datasets and codes. This situation has already improved in the past 2-3 years but is still in the initial stages. More transparency should be enforced in favour of reproducing the results.

Chapter 10

Conclusion & Future Work

> "Success in creating AI would be the biggest event in human history. Unfortunately, it might also be the last." – (Stephen Hawking)

In this concluding chapter, the key findings of this study are summarized. After drawing useful conclusions, the foundation for future work and future research questions that can be addressed as part of further investigation is also being laid.

10.1 Conclusion

The primary goal of this dissertation has been to improve existing feature extraction techniques for time series analysis specifically for usage with deep learning. To achieve this, the concept of time series analysis was presented objectively in Chapter 1. The state of affairs for time series analysis for industrial data science was presented in detail in Chapter 2. The most prominent data mining process CRISP-DM along with its extension called DMME for industrial production dataset were used. The data set was obtained as part of an industrial research collaboration between the research group and the industrial partners. A detailed review of the literature related to application of data science to industrial use-cases was done and the flaws and short comings of not only the used process models but also that of the industry practises were highlighted.

As the industrial datasets were not found adequate for research purposes, a shift towards ECG datasets was made. It was shown for the first time that different human activities including falls can be identified from the ECG signals (See Chapter 5). The obtained ECG signals were filtered and then converted into wavelet transforms which were then converted into 3-D scalograms. Two pre-trained neural networks called AlexNet and GoogLeNet were retrained to classify scalograms into fall, resting, and daily activities.

To reduce the significant manual pre-processing effort, two algorithms for automatic feature extraction were proposed and showed experimentally that the algorithms extract relevant features at the pre-processing stage (See Chapter 6). The first algorithm extracts the information with the help of feature maps and then a simple LSTM is shown to perform better than other state-of-the-art algorithms on the classification task. The same performance was shown when wavelet transformation was used as a feature extractor followed by a plain transformer encoder for the classification function. So the performance of existing deep learning models was elevated both computationally and functionally by deploying appropriate feature extractors before them.

To investigate further the impact of wavelet transformation as dimensional embeddings, it was proposed in Chapter 7 that wavelet transforms would form adequate feature extractors to replace dimensional embeddings in transformers and would not require positional encoding for the classification task. This thesis for the first time, studies in detail the usage of the dimensional embeddings of the transformers with time series classification. An ablation study was designed to observe the effects of positional encoding with wavelets, and different dimensional embeddings with transformers for time series. It was shown experimentally by using different time series datasets in Chapter 8, that wavelets and feature maps broaden the performance of the plain transformers.

At the final stage of the thesis, we have a deep learning classifier that classifies different human activities. We also get algorithms that can take automatic feature extraction further ahead for time series and we get a better overview using wavelet transformations as dimensional embeddings. But all of this knowledge acts as a base for many interesting research questions for the future. Some of them have already been considered but due to the limited duration of Ph.D. studies, they have to be assigned as future perspectives.

10.2 Future Work

As stated in the previous section, many interesting aspects and research questions arise from the current work upon which future work can be based. Some of them are listed below:

- Extending the existing process models like CRISP-DM and DMME as explained in Chapter 2 for data science applications to better adapt to the latest developing technologies in the realm of deep learning and handle the expectations of the industrial partners. The extended process models can focus on the personnel involvement and the roles at every phase of the process model. It can also highlight on more objective data quality assurance in the initial stages of the process.

- The work from Chapter 5 can considered to be extended in the following ways:
  - Design further experiments to collect more data from the experiments for fall detection and human activity recognition from ECG signals.
  - Outline the characteristics in the ECG signals indicating different activities and different falls using explainable AI techniques like activation maps. XAI can also be implemented to the algorithms to assist bioengineers and physicians in interpreting results obtained from the model.
- Chapter 8 can be extended in the following way:
  - Chapter 6 and chapter 8 discuss the use of wavelets and feature maps as feature extractors for deep learning classification. The exact extracted features and their importance for the classification can be explored using proper Explainable AI. Investigation is underway by other Ph.D. students in our research group on relevant features for classification using attention maps as XAI techniques.
  - This thesis has explored some of the automatic feature extraction techniques using prominent mathematical operations for deep learning models. There are still many possibilities that can be explored with dimensional embeddings with transformers like Laplace transformation etc. The difference between the properties of different kinds of wavelet transformations is also not well studied with respect to the feature extraction capability of time series. This has to be explored in detail to assign particular techniques to relevant domain time series.
  - More mathematical foundation needs to be laid out for the experimental evidence collected in the thesis for transformer in general and attention in particular. Some literature can already be found which is looking into the theoretical foundation of transformers and deep learning neural networks.

Appendix A

Mathematical Notations

In this appendix, the notation, and related mathematical facts and conventions used throughout the dissertation are collected for the convenience of the reader.

Mathematical parameters are written in italics, but not units, numbers and mathematical functions like logarithms which are written in bold.

A.1 Vectors

Vectors are denoted by lower case bold Roman letters such as $\mathbf{x}$ and all vectors are assumed to be column vectors. A superscript T denotes the transpose of a matrix or vector, so that $\mathbf{x^T}$ will be row vector.

A.2 Hilbert Space

A Hilbert space is a vector space $\mathbf{H}$ with an inner product $\langle f, g\rangle$ such that the norm defined by

$$|f| = \sqrt{\langle f, f\rangle} \tag{A.1}$$

turns $\mathbf{H}$ into a complete metric space. Examples of finite-dimensional Hilbert spaces include

1. The real numbers $\mathbb{R}^n$ with $\langle v, u\rangle$ the vector dot product of v and u.

2. The complex numbers $\mathbb{C}^n$ with $\langle v, u\rangle$ the vector dot product of v and the complex conjugate of u.

A Hilbert space is always a Banach space, but the converse need not hold.

A.3 Tensor Product

Tensor products are used to extend spaces of one-dimensional signals into spaces of multidimensional signals. A tensor product $f_1 \otimes f_2$ between vectors of two Hilbert spaces $\mathbf{H}_1$ and $\mathbf{H}_2$ satisfies the properties of linearity and distributively as follows:

$$Linearity : \forall \lambda \in \mathbb{C}, \lambda(f_1 \otimes f_2) = (\lambda f_1) \otimes f_2 = f_1 \otimes (\lambda f_2) \tag{A.2}$$

$$Distributivity : (f_1 + g_1) \otimes (f_1 + g_1) = (f_1 \otimes f_2) + (f_1 \otimes g_2) + (g_1 \otimes f_2) + (g_1 \otimes g_2) \tag{A.3}$$

This tensor product yields a new Hilbert space $\mathbf{H} = \mathbf{H}_1 \otimes \mathbf{H}_2$ that includes all vectors of the form $f_1 \otimes f_2$ where $f_1 \in \mathbf{H}_1$ and $f_1 \in \mathbf{H}_2$, as well as linear combinations of such vectors. An inner product in $\mathbf{H}$ is derived from inner products in $\mathbf{H}_1$ and $\mathbf{H}_2$ by

$$\langle f_1 \otimes f_2, g_1 \otimes g_2 \rangle_{\mathbf{H}} = \langle f_1, g_1 \rangle_{\mathbf{H}_1} \langle f_2, g_2 \rangle_{\mathbf{H}_2} \tag{A.4}$$

A.4 Sets

- $\mathbb{N}$: Positive integers including 0
- $\mathbb{Z}$: Integers
- $\mathbb{R}$: Real numbers
- $\mathbb{R}^+$: Positive real numbers

A.5 Signals

- $f(t)$: Continuous time Signal
- $f[n]$: Discrete Signal

A.6 Probability

- X: Random variable
- $E[X]$: Expected Value
- $Cov(X_1, X_2)$: Covariance

Appendix B

Code and Data Availability

In this appendix, the datasets used in this study are listed below. The data sets used are publicly available and present in the corresponding repositories:

- ECG HAR data set : [32]
- PTB DB data set : [27]
- PTB XL : [251]
- The UCR Time Series Classification Archive, ECG5000 [60]
- The UCR Time Series Classification Archive, UCR ECG200 [60]
- The UCR Time Series Classification Archive, [60]
- CSI Dataset. [214].

The code is available on GitHub at https://github.com/buttfatimasajid/Towards-Automated-Feature-Extraction-For-Deep-Learning-Classification-of-Electrocardiogram-Signals.

Appendix C

Code and Note to the Technology

This appendix sheds lights on the Python code and libraries used to implement the experiments mentioned in the dissertation.

C.1 Experimental Set up

The experimental set up has been explained along with each experiment in the corresponding chapters. For all the experiments, mainly python Jupyter notebooks were used. All the code files are available on the GitHub repository. The Jupyter notebooks were deployed both locally and on remote server which is located at fb2, Frankfurt university of applied Sciences, Frankfurt. For local installation, DataSpell 2023 with Python Version 3.8.8 was used mainly for industrial data analysis.

The Python environment was implemented following the standard installation procedures provided. Additionally, custom scripts were developed to automate certain tasks and enhance the efficiency of the analysis and development process.

Examples of output and results generated using the code are included in the chapters for reference.

C.2 Libraries

Following major libraries were used to implement the experiments:

- Matplotlib played a crucial role in facilitating data analysis and interpretation, contributing significantly to the findings presented in this thesis.

- **Numpy**: NumPy is a powerful library in Python for numerical computing, providing support for multi-dimensional arrays and a wide range of mathematical functions. It was mainly used for data handling. Note that NumPy itself does not directly support GPU computations, so the required computations were done on CPUs and then shifted back to GPU.

- **Scikit-learn**: commonly abbreviated as sklearn, is a popular machine learning library in Python that provides efficient tools for data mining, data analysis, and machine learning tasks. It was implemented for data analysis for industrial datasets. It was also used to device testing of the trained models using metrics like, accuracy, F1, confusion matrix etc.

- **Torch**: PyTorch is an open-source machine learning framework developed by Facebook's AI Research lab (FAIR). It provides a flexible and dynamic approach to building and training deep neural networks. It was extensively for deep learning neural network implementations like transformers, CNNs and LSTMs. At its core, PyTorch provides multi-dimensional arrays called Tensors which are used to implement additional capabilities than Numpy array for GPU acceleration and automatic differentiation.

  PyTorch includes a built-in automatic differentiation engine called autograd, which automatically computes gradients for tensors during forward and backward passes of neural network training. This simplifies the implementation of custom loss functions and optimization algorithms.

  PyTorch offers a rich collection of neural network building blocks, including modules for defining layers (e.g., linear layers, convolutional layers, recurrent layers), activation functions, loss functions, and optimizers. These building blocks were used to design the algorithms.

  With PyTorch, you can define neural networks dynamically using Python control flow constructs like loops and conditionals. This allows for more complex and adaptive network architectures.

  PyTorch also provides seamless integration with NVIDIA GPUs for accelerated training and inference. Tensors can be easily moved between CPU and GPU devices, and most operations automatically utilize GPU resources when available.

- **PyWavelets (pywt)**: PyWavelets (pywt) is an open-source Python library that provides wavelet transforms and related signal processing functions. This library was mainly used for obtaining the wavelet transformation of time series signals. It implements various wavelet transforms, including discrete wavelet transform (DWT), continuous wavelet transform (CWT), and stationary wavelet transform (SWT).

With PyWavelets, you can perform multiresolution analysis (MRA) using wavelet transforms. This allows for the decomposition of signals into different scales or levels, enabling efficient representation and processing of signals at different resolutions.

PyWavelets provides a collection of predefined wavelet filter banks, including popular families such as Daubechies, Symlets, and Haar wavelets. Additionally, custom wavelets or filter banks using the library's API can also be used.

PyWavelets offers functions for signal denoising and compression based on wavelet thresholding techniques. These methods exploit the sparsity of wavelet coefficients to remove noise or reduce signal redundancy, leading to improved signal quality or reduced data size.

PyWavelets can be used for feature extraction in various applications, including pattern recognition, classification, and time-series analysis. Wavelet coefficients can serve as informative features for characterizing signals or images in machine learning tasks.

- **Kymatio**: Kymatio is an open-source Python library for wavelet scattering transforms, primarily focused on deep learning and signal processing tasks. It was implemented to obtain the scattering wavelet transformation of the signals as feature extractors for transfomers.

  Kymatio implements wavelet scattering transforms, which, unlike traditional wavelet transforms, applies multiple layers of wavelet transforms and nonlinear operations iteratively to create a hierarchical representation of the input data. These transforms provide a powerful method for analyzing and extracting features from signals and images.

  Kymatio is designed for use in deep learning applications, particularly in tasks such as image classification, object recognition, and signal denoising. It offers efficient implementations of scattering transforms that can be integrated seamlessly into deep neural network architectures.

  One of the key features of scattering transforms is their ability to produce invariant representations of input signals with respect to translations, rotations, and deformations. This property makes them well-suited for tasks requiring robust and invariant feature extraction.

  Kymatio offers flexibility and configurability in terms of the choice of wavelet families, filter bank parameters, and scattering transform settings. Users can customize the transforms to suit their specific application requirements and preferences.

Kymatio comes with comprehensive documentation, including tutorials, examples, and API references. This helps users get started with the library quickly and understand its functionalities and usage.

Bibliography

[1] A. A. Abdullah, M. M. Hassan, and Y. T. Mustafa. A Review on Bayesian Deep Learning in Healthcare: Applications and Challenges. *IEEE Access*, 10:36538–36562, 2022. doi:10.1109/ACCESS.2022.3163384.

[2] U. Acharya, H. Fujita, S. Oh, Y. Hagiwara, J. Tan, and M. Adam. Application of Deep Convolutional Neural Network for Automated Detection of Myocardial Infarction Using ECG Signals. *Information Sciences*, 415-416:190–198, 2017. doi:10.1016/j.ins.2017.06.014. URL https://www.sciencedirect.com/science/article/pii/S0020025517308009.

[3] U. R. Acharya, H. Fujita, O. S. Lih, M. Adam, J. H. Tan, and C. K. Chua. Automated Detection of Coronary Artery Disease Using Different Durations of ECG Segments with Convolutional Neural Network. *KNOWLEDGE-BASED SYSTEMS*, 132:62–71, SEP 15 2017. ISSN 0950-7051. doi:10.1016/j.knosys.2017.06.003.

[4] P. S. Addison. Wavelet transforms and the ECG: A review. *Physiological Measurement*, 26(5):R155–R199, aug 2005. doi:10.1088/0967-3334/26/5/r01.

[5] S. Ahmed, I. E. Nielsen, A. Tripathi, S. Siddiqui, R. P. Ramachandran, and G. Rasool. Transformers in Time-Series Analysis: A Tutorial. *Circuits, Systems, and Signal Processing*, 42(12):7433–7466, 2023. doi:10.1007/s00034-023-02454-8. URL https://doi.org/10.1007/s00034-023-02454-8.

[6] S. Albawi, T. Mohammed, and S. Al-Zawi. Understanding of a Convolutional Neural Network. In *2017 International Conference On Engineering And Technology (ICET)*, pages 1–6, 2017.

[7] Allcock, LM and O'Shea, D. Diagnostic Yield and Development of a Neuro cardiovascular Investigation Unit for Older Adults in a District Hospital. *Journal of Gerontology: MEDICAL SCIENCES 2000*, 2000.

[8] J. Alman and V. V. Williams. A Refined Laser Method and Faster Matrix Multiplication. In *32nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2021)*, 2020. arXiv:2010.05846.

[9] A. Altmann, L. Toloşi, O. Sander, and T. Lengauer. Permutation Importance: A Corrected Feature Importance Measure. *Bioinformatics*, 26(10):1340–1347, 2010. doi:10.1093/bioinformatics/btq134.

[10] D. Alvarez-Melis and T. S. Jaakkola. Towards Robust Interpretability with Self-Explaining Neural Networks. In *Advances in Neural Information Processing Systems*, 2018.

[11] J. Andén and S. Mallat. Deep Scattering Spectrum. *IEEE Transactions on Signal Processing*, 62(16):4114–4128, 2014.

[12] I. Ara, M. N. Hossain, and S. Y. Mahbub. Baseline Drift Removal and De-noising of the ECG signal Using Wavelet Transform. *International Journal of Computer Applications*, June, 2014.

[13] S. Ö. Arik and T. Pfister. Tabnet: Attentive Interpretable Tabular Learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 6679–6687, 2021.

[14] J. F. Arinez, Q. Chang, R. X. Gao, C. Xu, and J. Zhang. Artificial Intelligence in Advanced Manufacturing: Current Status and Future Outlook. *Journal of Manufacturing Science and Engineering*, 142(11):110804, 08 2020. ISSN 1087-1357. doi:10.1115/1.4047855. URL https://doi.org/10.1115/1.4047855.

[15] Arthur C. Guyton, John E. Hall. *Text Book of Medical Physiology*. Elsevier, 2006.

[16] W. M. Association. World Medical Association Declaration of Helsinki: Ethical Principles for Medical Research Involving Human Subjects. *JAMA*, 310(20):2191–2194, November 2013. doi:10.1001/jama.2013.281053.

[17] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek. On Pixel-wise Explanations for Non-linear Classifier Decisions by Layer-wise Relevance Propagation. *PloS one*, 10(7):e0130140, 2015.

[18] A. Baevski, H. Zhou, A. Mohamed, and M. Auli. wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

[19] U. B. Baloglu, M. Talo, O. Yildirim, R. S. Tan, and U. R. Acharya. Classification of Myocardial Infarction With Multi-lead ECG Signals and Deep CNN. *Pattern Recognition Letters*, 122:23–30, 2019. ISSN 0167-8655. doi:https://doi.org/10.1016/j.patrec.2019.02.016. URL https://www.sciencedirect.com/science/article/pii/S016786551930056X.

[20] Bendigo Health. Basic ECG Interpretation Learning Package, 2016. URL http://www.bendigohealth.org.au/Content/Docs/ECG%20learning%20package.pdf. Retrieved on 2019-01-12.

[21] Betts, Desaix, Johnsons, Jody E. Johnson, Korol, Kruse, Poe, Wise, Womble,Young. *Anatomy and Physiology*. OpenStax, 2017.

[22] A. M. Binder de Serdio, D. Stegelmeyer, and F. S. Butt. Early Indicators of Project Abandonment in Industry-Academia Collaborations: Developing an Assessment Framework for Industrial Data Science Projects. In *10th Spanish-German Symposium on Applied Computer Science (SGSOACS 2023)*, Cádiz (Spain), 2024.

[23] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006. ISBN 0387310738.

[24] S. Bitrus, I. Velkavrh, and E. Rigger. Applying an Adapted Data Mining Methodology (DMME) to a Tribological Optimisation Problem. In *Data Science–Analytics and Applications: Proceedings of the 3rd International Data Science Conference–iDSC2020*, pages 38–43. Springer, 2021.

[25] L. L. Blunda, L. Gutiérrez-Madroñal, M. F. Wagner, and I. Medina-Bulo. A Wearable Fall Detection System Based on Body Area Networks. *IEEE Access*, 8:193060–193074, 2020. doi:10.1109/ACCESS.2020.3032497.

[26] M. Bodini, M. W. Rivolta, and R. Sassi. Opening the Black Box: Interpretability of Machine Learning Algorithms in Electrocardiography. *Philosophical Transactions of the Royal Society A*, 379(2212):20200253, 2021.

[27] R. Bousseljot, D. Kreiseler, and A. Schnabel. Nutzung der ekg-signaldatenbank cardiodat der ptb über das internet. *Biomedizinische Technik*, 40(Ergänzungsband 1):S 317, 1995. URL https://doi.org/10.13026/C28C71.

[28] M. Bownik and J. W. Iverson. Multiplication-Invariant Operators and the Classification of LCA Group Frames. *Journal of Functional Analysis*, 280(2):108780, 2021.

[29] A. P. Bradley. Shift-Invariance in the Discrete Wavelet Transform. *Proceedings of VIIth Digital Image Computing: Techniques and Applications. Sydney*, 2003.

[30] L. Breiman. Statistical Modeling: The Two Cultures (with comments and a rejoinder by the author). *Statistical science*, 16(3):199–231, 2001.

[31] A. Brennen. What Do People Really Want When They Say They Want" Explainable AI?" We Asked 60 Stakeholders. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–7, 2020.

[32] F. Butt, L. La Blunda, M. Wagner, J. Schäfer, I. Medina-Bulo, and D. Oteiza. ECG Data for Deep Transfer Learning, 2020. URL https://dx.doi.org/10.3390/info12020063.

[33] F. S. Butt. Fall detection using machine learning techniques on ecg signals. Master's thesis, Frankfurt University of Applied Sciences, Frankfurt, Frankfurt Germany, March 2019.

[34] F. S. Butt, L. La Blunda, M. F. Wagner, J. Schäfer, I. Medina-Bulo, and D. Gómez-Ullate. Fall Detection from Electrocardiogram (ECG) Signals and Classification by Deep Transfer Learning. *Information*, 12(2):63, 2021. doi:10.3390/info12020063.

[35] F. S. Butt, J. Schäfer, M. F. Wagner, and D. G.-U. Oteiza. Time series analysis using machine learning techniques: Medical and industrial applications. In *proceedings of II Jornadas de Investigación Predoctoral en Ingeniería Informática (JIPII 2022)*, Cádiz (Spain), 2022. Department of Computer Science and Engineering, UCA.

[36] F. S. Butt, J. Schäfer, M. F. Wagner, and D. G.-U. Oteiza. Towards Automated Feature Extraction For Deep Learning Classification of Electrocardiogram Signals. In *8th Spanish-German Symposium on Applied Computer Science (SGSOACS 2022)*, Toledo (Spain), 2022.

[37] F. S. Butt, M. F. Wagner, J. Schäfer, and D. G. Ullate. Toward automated feature extraction for deep learning classification of electrocardiogram signals. *IEEE Access*, 10:118601–118616, 2022. doi:10.1109/ACCESS.2022.3220670.

[38] F. S. Butt, J. Schäfer, M. F. Wagner, and D. G.-U. Oteiza. Explainable AI for time series classification - An Overview and future directions. In *9th Spanish-German Symposium on Applied Computer Science (SGSOACS 2023)*, Tutzing (Germany), 2023.

[39] F. S. Butt, J. Schäfer, M. F. Wagner, and D. G.-U. Oteiza. Automatic Feature extraction for time series analysis. In *Workshop: Statistics, Machine Learning and Applications*, Kaub (Germany), 2023.

[40] F. S. Butt, J. Schäfer, M. F. Wagner, D. Stegelmeyer, and D. G.-U. Oteiza. Application of crisp-dm and dmme to a case study of condition monitoring of lens coating machines. In *Proceedings of the 2023 IEEE International Workshop on Metrology for Industry 4.0 & IoT (MetroInd4.0&IoT)*, Brescia (Italy), 2023. IEEE. doi:10.33965/ac2019_201912c027.

[41] F. S. Butt, M. F. Wagner, J. Schäfer, and D. G.-U. Oteiza. In *Adopting Dimensional Embedding For Time Series Classification In Transformer Architecture*, 2024. Submitted.

[42] L. Cai, K. Janowicz, G. Mai, B. Yan, and R. Zhu. Traffic transformer: Capturing the continuity and periodicity of time series for traffic forecasting. *Transactions in GIS*, 24(3):736–755, 2020.

[43] F. Castells, P. Laguna, L. Sörnmo, A. Bollmann, and J. Roig. Principal component analysis in ecg signal processing. *EURASIP Journal on Advances in Signal Processing*, 2007(1):074580, 2007. doi:10.1155/2007/74580. URL https://doi.org/10.1155/2007/74580.

[44] H. Castro, F. Costa, L. Ferreira, P. Ávila, G. D. Putnik, and M. Cruz-Cunha. Data Science for Industry 4.0: A Literature Review on Open Design Approach. *Procedia Computer Science*, 204:877–884, 2022. ISSN 1877-0509. doi:https://doi.org/10.1016/j.procs.2022.08.106. International Conference on Industry Sciences and Computer Science Innovation.

[45] C. Catley, K. Smith, C. McGregor, and M. Tracy. Extending CRISP-DM to Incorporate Temporal Data Mining of Multidimensional Medical Data Streams: A Neonatal Intensive Care Unit Case Study. In *Proceedings of the 22nd IEEE International Symposium on Computer-Based Medical Systems*, pages 1–5, Sao Carlos, SP, Brazil, 22–25 June 2009.

[46] S. Celin and K. Vasanth. Ecg signal classification using various machine learning techniques. *Journal of Medical Systems*, 42(12):241, 2018. doi:10.1007/s10916-018-1083-6. URL https://doi.org/10.1007/s10916-018-1083-6.

[47] Centers for Disease Control and Prevention. Underlying Cause of Death, 1999-2020 Request. https://wonder.cdc.gov/ucd-icd10.html, 2020.

[48] C. Che, P. Zhang, M. Zhu, Y. Qu, and B. Jin. Constrained Transformer Network for ECG Signal Processing and Arrhythmia Classification. *BMC MEDICAL INFORMATICS AND DECISION MAKING*, 21(1), JUN 9 2021. doi:10.1186/s12911-021-01546-2.

[49] Z. Chen, Y. Bei, and C. Rudin. Concept Whitening for Interpretable Image Recognition. *Nature Machine Intelligence*, 2(12):772–782, 2020.

[50] E. Choi, M. T. Bahadori, J. Sun, J. Kulas, A. Schuetz, and W. Stewart. Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/231141b34c82aa95e48810a9d1b33a79-Paper.pdf.

[51] W. J. Clancey. Heuristic classification. *Artificial Intelligence*, 27(3):289–350, 1985. ISSN 0004-3702. doi:https://doi.org/10.1016/0004-3702(85)90016-5. URL https://www.sciencedirect.com/science/article/pii/0004370285900165.

[52] G. D. Clifford, F. Azuaje, P. McSharry, et al. *Advanced methods and tools for ECG data analysis*, volume 10. Artech house Boston, 2006.

[53] E. Commission. Digital Transformation Monitor, Germany Industrie 4.0, January 2017. URL 'https://ati.ec.europa.eu/sites/default/files/2020-06/DTM_Industrie%204.0_DE.pdf'. This is an official document prepared by the European Commission to provide an overview and definitions of current trends in digital transformations with respect to Industry 4.0.

[54] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20:273–297, 1995.

[55] Z. Cui, W. Chen, and Y. Chen. Multi-scale convolutional neural networks for time series classification, 2016.

[56] D. Cvetkovic, E. D. Übeyli, and I. Cosic. Wavelet Transform Feature Extraction From Human PPG, ECG, and EEG Signal Responses to ELF PEMF Exposures: A Pilot Study. *Digital Signal Processing*, 18(5):861–874, 2008. ISSN 1051-2004. doi:https://doi.org/10.1016/j.dsp.2007.05.009. URL https://www.sciencedirect.com/science/article/pii/S1051200407000978.

[57] A. Dallali, A. Kachouri, and M. Samet. A Classification of Cardiac Arrhythmia Using WT, HRV, and Fuzzy C-Means Clustering. *Signal Processing: An International Journal (SPJI)*, Volume (5):101–108, 2011,1.

[58] N. Damodaran, E. Haruni, M. Kokhkharova, and J. Schäfer. Device Free Human Activity and Fall Recognition using WiFi Channel State Information (CSI). *CCF Transactions on Pervasive Computing and Interaction*, 2:1–17, January 2020. doi:10.1007/s42486-020-00027-1.

[59] N. Das and M. Chakraborty. Performance Analysis of FIR and IIR filters for ECG Signal De-noising based on SNR. *2017 Third International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)*, 2017.

[60] H. A. Dau, A. Bagnall, K. Kamgar, C.-C. M. Yeh, Y. Zhu, S. Gharghabi, C. A. Ratanamahatana, and E. Keogh. The ucr time series archive, 2019.

[61] I. Daubechies. *Ten Lectures on Wavelets*. Society for Industrial, 1992.

[62] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[63] D. F. Dickinson. The Normal ECG in Childhood and Adolescence. *Heart*, 91(12): 1626–1630, 2005.

[64] Diego Castro, William Coral, Camilo Rodriguez,Jose Cabra and Julian Colorado. Wearable-Based Human Activity Recognition Using an IoT Approach. *Sensors and Actuators in Smart Cities)*, 2017.

[65] T. G. Dietterich. Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. *Neural Computation*, 10:1895–1923, October 1998. doi:10.1162/089976698300017197.

[66] Donghui Zhang. Wavelet Approach for ECG Baseline Wander Correction and Noise Reduction. *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, pages 1105–1108, 2006.

[67] R. Donida Labati, E. Muñoz, V. Piuri, R. Sassi, and F. Scotti. Deep-ecg: Convolutional neural networks for ecg biometric recognition. *Pattern Recognition Letters*, 126:78–85, 2019. ISSN 0167-8655. doi:https://doi.org/10.1016/j.patrec.2018.03.028. Robustness, Security and Regulation Aspects in Current Biometric Systems.

[68] Dr. Lewis Potter. Understanding an ECG, 2011. URL "https://geekymedics.com/understanding-an-ecg/". Retrieved on 2019-01-12.

[69] L. Drowatzky, H. Wiemer, and S. Ihlenfeldt. Data Mining Suitable Digitization of Production Systems–A Methodological Extension to the DMME. In *Congress of the German Academic Association for Production Technology*, pages 524–534. Springer, 2022.

[70] F. Duarte. Amount of data created daily (2023), 2023. URL https://explodingtopics.com/blog/data-generated-per-day.

[71] Z. Ebrahimi, M. Loni, M. Daneshtalab, and A. Gharehbaghi. A Review on Deep Learning Methods for ECG Arrhythmia Classification. *Expert Systems With Applications: X*, 7:100033, 2020. doi:10.1016/j.eswax.2020.100033. URL https://www.sciencedirect.com/science/article/pii/S2590188520300123.

[72] B. Efron. Prediction, Estimation, and Attribution. *International Statistical Review*, 88:S28–S59, 2020.

[73] F. A. Elhaj, N. Salim, A. R. Harris, T. T. Swee, and T. Ahmed. Arrhythmia recognition and classification using combined linear and nonlinear features of ecg signals. *COMPUTER METHODS AND PROGRAMS IN BIOMEDICINE*, 127: 52–63, APR 2016. ISSN 0169-2607. doi:10.1016/j.cmpb.2015.12.024.

[74] P. Esling and C. Agon. Time-series Data Mining. *ACM Computing Surveys (CSUR)*, 45(1):1–34, 2012.

[75] K. Feng, X. Pi, H. Liu, and K. Sun. Myocardial Infarction Classification Based on Convolutional Neural Network and Recurrent Neural Network. *Applied Sciences*, 9:1879, 2019,5.

[76] S. Fong, K. Lan, P. Sun, S. Mohammed, J. Fiaidhi, and S. Mohammed. A Time-Series Pre-processing Methodology for Biosignal Classification Using Statistical Feature Extraction. In *Proceedings of the 10th IASTED international conference on biomedical engineering (Biomed'13)*, pages 207–214, 2013.

[77] N. M. Foumani, C. W. Tan, G. I. Webb, and M. Salehi. Improving Position Encoding of Transformers for Multivariate Time Series classification. *Data Mining and Knowledge Discovery*, 38(1):22–48, 2024. doi:10.1007/s10618-023-00948-2. URL https://doi.org/10.1007/s10618-023-00948-2.

[78] J. E. Fowler. The Redundant Discrete Wavelet Transform and Additive Noise. *IEEE Signal Processing Letters*, 12(9):629–632, 2005.

[79] H. Führ. Admissible Vectors for the Regular Representation. *Proceedings of the American Mathematical Society*, 130(10):2959–2970, 2002.

[80] W. Fuller. *Introduction to Statistical Time Series*. Probability and Statistics Series. Wiley, 1976. ISBN 9780471287155. URL https://books.google.de/books?id=pYwpAQAAMAAJ.

[81] Z. Gao, X. Wang, S. Sun, D. Wu, J. Bai, Y. Yin, X. Liu, H. Zhang, and V. De Albuquerque. Learning Physical Properties in Complex Visual Scenes: An Intelligent Machine for Perceiving Blood Flow Dynamics from Static CT Angiography Imaging. *Neural Networks*, 123:82–93, 2020. URL https://www.sciencedirect.com/science/article/pii/S0893608019303764.

[82] D. Genoud, V. Cuendet, and J. Torrent. Soft Fall Detection Using Machine Learning in Wearable Devices. In *Conference Technoark 2018: Quantified Self*, pages 501–505, 03 2016. doi:10.1109/AINA.2016.124.

[83] N. S. Geoffrey Hinton and K. Swersky. 'neural networks for machine learning nline course', 2016.

[84] A. L. Goldberger, L. A. N. 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 (June 13). Circulation Electronic Pages: http://circ.ahajournals.org/content/101/23/e215.full PMID:1085218; doi: 10.1161/01.CIR.101.23.e215.

[85] B. Goodman and S. Flaxman. European Union Regulations on Algorithmic Decision Making and a "Right to Explanation". *AI Magazine*, 38(3):50–57, Sept. 2017. ISSN 2371-9621. doi:10.1609/aimag.v38i3.2741. URL http://dx.doi.org/10.1609/aimag.v38i3.2741.

[86] R. Grossman, C. Kamath, P. Kegelmeyer, V. Kumar, and R. Namburu, editors. *Data Mining for Scientific and Engineering Applications*. Springer, New York, NY, USA, 2001. ISBN 978-1-4615-1733-7.

[87] J. Guan, W. Wang, P. Feng, X. Wang, and W. Wang. Low-Dimensional Denoising Embedding Transformer for ECG Classification. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1285–1289, 2021.

[88] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi. A Survey of Methods for Explaining Black Box Models. *ACM computing surveys (CSUR)*, 51(5):1–42, 2018.

[89] A. Guillaume, C. Vrain, and W. Elloumi. Random dilated shapelet transform: A new approach for time series shapelets. In *International Conference on Pattern Recognition and Artificial Intelligence*, pages 653–664. Springer, 2022.

[90] M. Guillemé, V. Masson, L. Rozé, and A. Termier. Agnostic Local Explanation for Time Series Classification. In *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 432–439, 2019. doi:10.1109/ICTAI.2019.00067.

[91] A. Gupta, E. Huerta, Z. Zhao, and I. Moussa. Deep Learning for Cardiologist-Level Myocardial Infarction Detection in Electrocardiograms. In *8th European Medical And Biological Engineering Conference*, pages 341–355, 2021.

[92] I. Guyon and A. Elisseeff. An introduction to Variable and Feature Selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.

[93] I. Guyon, S. Gunn, M. Nikravesh, and L. A. Zadeh. *Feature extraction: Foundations and Applications*, volume 207. Springer, 2008.

[94] A. Haar. Zur Theorie der orthogonalen Funktionensysteme. *Mathematische Annalen*, 69(3):331–371, 1910. doi:10.1007/BF01456326. URL https://doi.org/10.1007/BF01456326.

[95] T. Hall. The Role of Data in Industry 4.0, 2020. URL https://industrytoday.com/the-role-of-data-in-industry-4-0/.

[96] A. Y. Hannun, P. Rajpurkar, M. Haghpanahi, G. H. Tison, C. Bourn, M. P. Turakhia, and A. Ng. Cardiologist-level Arrhythmia Detection and Classification in Ambulatory Electrocardiograms Using a Deep Neural Network. *Nature Medicine*, 25:65–69, 2019.

[97] M. A. Hasnul, N. A. A. Aziz, S. Alelyani, M. Mohana, and A. A. Aziz. Electrocardiogram-Based Emotion Recognition Systems and Their Applications in Healthcare – A Review. *Sensors*, 21(15), 2021. ISSN 1424-8220. doi:10.3390/s21155015. URL https://www.mdpi.com/1424-8220/21/15/5015.

[98] J. Heath and C. McGregor. CRISP-DM0: A Method to Extend CRISP-DM to Support Null Hypothesis Driven Confirmatory Data Mining. In *Proceedings of the 1st Advances in Health Informatics Conference*, pages 96–101, Kitchener, Ontario, Canada, 28–30 April 2010.

[99] M. Henke, editor. *Instandhaltungsforum, InFo 2019*, 2019. Fraunhofer IML. https://publica.fraunhofer.de/handle/publica/404355.

[100] H. V. Hoang and M. Tran. Deepsense-inception: Gait identification from inertial sensors with inception-like architecture and recurrent network. In *Conference: 2017 13th International Conference on Computational Intelligence and Security (CIS)*, pages 594–598, Dec 2017. doi:10.1109/CIS.2017.00138.

[101] S. Hochreiter and J. Schmidhuber. Long Short-Term Memory. *Neural Comput.*, 9 (8):1735–1780, 1997. URL https://doi.org/10.1162/neco.1997.9.8.1735.

[102] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417–441 and 498–520, 1933.

[103] B. Hou, J. Yang, P. Wang, and R. Yan. Lstm-based auto-encoder model for ecg arrhythmias classification. *IEEE TRANSACTIONS ON INSTRUMENTATION AND MEASUREMENT*, 69(4, 1):1232–1240, APR 2020. ISSN 0018-9456. doi:10.1109/TIM.2019.2910342.

[104] P. O. Hoyer and A. Hyvärinen. Independent component analysis applied to feature extraction from colour and stereo images. *Network: computation in neural systems*, 11(3):191–210, 2000.

[105] R. Hu, J. Chen, and L. Zhou. A Transformer-based Deep Neural Network for Arrhythmia Detection Using Continuous ECG Signals. *COMPUTERS IN BIOLOGY AND MEDICINE*, 144, MAY 2022. ISSN 0010-4825. doi:10.1016/j.compbiomed.2022.105325.

[106] R. Hu, J. Chen, and L. Zhou. A Transformer-based Deep Neural Network for Arrhythmia Detection Using Continuous ECG Signals. *Computers in Biology and Medicine*, 144:105325, 2022. ISSN 0010-4825. doi:https://doi.org/10.1016/j.compbiomed.2022.105325. URL https://www.sciencedirect.com/science/article/pii/S0010482522001172.

[107] J. Huang, B. Chen, B. Yao, and W. He. Ecg arrhythmia classification using stft-based spectrogram and convolutional neural network. *IEEE ACCESS*, 7:92871–92880, 2019. ISSN 2169-3536. doi:10.1109/ACCESS.2019.2928017.

[108] S. Huber, H. Wiemer, D. Schneider, and S. Ihlenfeldt. Dmme: Data mining methodology for engineering applications – a holistic extension to the crisp-dm model. *Procedia CIRP*, 79:403–408, 2019.

[109] A. Hyvärinen. Independent component analysis: recent advances. *Philos Trans A Math Phys Eng Sci*, 371(1984):20110534, Feb 2013. ISSN 1364-503X (Print); 1471-2962 (Electronic); 1364-503X (Linking). doi:10.1098/rsta.2011.0534.

[110] N. Insights. Artificial Intelligence: Localization Winners, Losers, Heroes, Spectators, and You, 2019. URL https://www.nimdzi.com/wp-content/uploads/2019/06/Nimdzi-AI-whitepaper.pdf. Accessed on 11 April 2023.

[111] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller. Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, 33(4):917–963, 2019. doi:10.1007/s10618-019-00619-1. URL https://doi.org/10.1007/s10618-019-00619-1.

[112] B. K. Iwana and S. Uchida. An empirical survey of data augmentation for time series classification with neural networks. *Plos one*, 16(7):e0254841, 2021.

[113] S. Jambukia, V. Dabhi, and H. Prajapati. Classification of ECG Signals Using Machine Learning Techniques: A Survey. In *2015 International Conference On Advances In Computer Engineering And Applications*, pages 714–721, 2015.

[114] M. Jayasanthi, G. Rajendran, and R. B. Vidhyakar. Independent component analysis with learning algorithm for electrocardiogram feature extraction and classification. *Signal, Image and Video Processing*, 15(2):391–399, 2021. doi:10.1007/s11760-020-01813-1. URL https://doi.org/10.1007/s11760-020-01813-1.

[115] C. K. Jha and M. H. Kolekar. Cardiac Arrhythmia Classification Using Tunable Q-Wavelet Transform Based Features and Support Vector Machine Classifier. *Biomedical Signal Processing and Control*, 59:101875, 2020.

[116] R. Jia and B. Liu. Human daily activity recognition by fusing accelerometer and multi-lead ecg data. In *2013 IEEE International Conference on Signal Processing, Communication and Computing (ICSPCC 2013)*, pages 1–4, 2013. doi:10.1109/ICSPCC.2013.6664056.

[117] M. Joseph. *Modern Time Series Forecasting with Python : Explore Industry-Ready Time Series Forecasting Using Modern Machine Learning and Deep Learning*. Packt Publishing, Limited, Birmingham, UNITED KINGDOM, 2022. ISBN 9781803232041. URL http://ebookcentral.proquest.com/lib/frankfurtmain/detail.action?docID=30259471.

[118] M. Kachuee, S. Fazeli, and M. Sarrafzadeh. Ecg heartbeat classification: A deep transferable representation. In *2018 IEEE International Conference on Healthcare Informatics (ICHI)*, pages 443–444, 2018. doi:10.1109/ICHI.2018.00092.

[119] M. W. Kadous and C. Sammut. Classification of multivariate time series and structured data using constructive induction. *Machine Learning*, 58(2):179–216, 2005. doi:10.1007/s10994-005-5826-5. URL https://doi.org/10.1007/s10994-005-5826-5.

[120] S. Kaplan Berkaya, A. K. Uysal, E. Sora Gunal, S. Ergin, S. Gunal, and M. B. Gulmezoglu. A survey on ecg analysis. *Biomedical Signal Processing and Control*, 43:216–235, 2018. ISSN 1746-8094. doi:https://doi.org/10.1016/j.bspc.2018.03.003. URL https://www.sciencedirect.com/science/article/pii/S1746809418300636.

[121] S. Karpagachelvi, M. Arthanari, and M. Sivakumar. ECG Feature Extraction Techniques - A Survey Approach. *arXiv:1005.0957*, 2010. URL https://arxiv.org/abs/1005.0957.

[122] A. Khan, A. Sohail, U. Zahoora, and A. Qureshi. A Survey of the Recent Architectures of Deep Convolutional Neural Networks. *Artificial Intelligence Review*, 53:5455–5516, 2020. doi:10.1007/s10462-020-09825-6. URL https://doi.org/10.1007/s10462-020-09825-6.

[123] D. R. Kher, T. Pawar, and D. Thakar. *Impact Analysis of Body Movements on Wearable Ambulatory Electrocardiogram*. OMICS International, 03 2015. ISBN 978-1-63278-048-5. doi:10.4172/978-1-63278-048-5-049.

[124] R. Kher. Wearable ambulatory electrocardiogram (ecg) and eeg dataset, 2020. URL https://dx.doi.org/10.21227/ysnc-gc65.

[125] H. Khorrami and M. Moavenian. A comparative study of DWT, CWT and DCT Transformations in ECG Arrhythmias Classification. *Expert Syst. Appl.*, 37:5751–5757, 2010,8.

[126] B. Kim, M. Wattenberg, J. Gilmer, C. Cai, J. Wexler, F. Viegas, and et al. Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV). In *International Conference on Machine Learning*, page 2668–2677. PMLR, 2018.

[127] Y.-G. Kim, D. Shin, M. Y. Park, S. Lee, M. S. Jeon, D. Yoon, and R. W. Park. Ecg-view ii, a freely accessible electrocardiogram database. *PloS one*, 12 (4):e0176222, 2017. ISSN 1932-6203. doi:10.1371/journal.pone.0176222. URL https://europepmc.org/articles/PMC5402933.

[128] D. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. *the 3rd International Conference for Learning Representations, San Diego*, 2015. URL http://arxiv.org/abs/1412.6980.

[129] S. Kiranyaz, T. Ince, and M. Gabbouj. Real-time patient-specific ecg classification by 1-d convolutional neural networks. *IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING*, 63(3):664–675, MAR 2016. ISSN 0018-9294. doi:10.1109/TBME.2015.2468589.

[130] G. Klosowski, T. Rymarczyk, D. Wojcik, S. Skowron, T. Cieplak, and P. Adamkiewicz. The use of time-frequency moments as inputs of lstm network for ecg signal classification. *ELECTRONICS*, 9(9), SEP 2020. doi:10.3390/electronics9091452.

[131] P. W. Koh, T. Nguyen, Y. S. Tang, S. Mussmann, E. Pierson, B. Kim, and P. Liang. Concept Bottleneck Models. In *International Conference on Machine Learning*, page 5338–5348. PMLR, 2020.

[132] J. Kojuri, R. Boostani, P. Dehghani, F. Nowroozipour, and N. Saki. Prediction of Acute Myocardial Infarction with Artificial Neural Networks in Patients with Non Diagnostic Electrocardiogram. *Journal Of Cardiovascular Disease Research*, 6:51–59, 2015,5.

[133] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25, pages 1097–1105. Curran Associates, Inc., 2012.

[134] J. Kuzilek, V. Kremen, F. Soucek, and L. Lhotska. Independent component analysis and decision trees for ecg holter recording de-noising. *PLoS One*, 9(6):e98450, 2014. ISSN 1932-6203 (Electronic); 1932-6203 (Linking). doi:10.1371/journal.pone.0098450.

[135] L. La Blunda, D. Corral-Plaza, M. Wagner, G. Ortiz, and I. Medina-Bulo. Distributed real-time based human activity analysis system. In *6th International Conference on Applied Computing in Cagliari (ITALY)*, volume 6, November, 2019.

[136] G. Lai, W. Chang, Y. Yang, and H. Liu. Modeling Long- and Short-Term Temporal Patterns with Deep Neural Networks. *CoRR*, abs/1703.07015, 2017. URL http://arxiv.org/abs/1703.07015.

[137] H. Lakkaraju, E. Kamar, R. Caruana, and J. Leskovec. Interpretable & Explorable Approximations of Black Box Models, 2017.

[138] G. Latif, F. Y. Al Anezi, M. Zikria, and J. Alghazo. EEG-ECG Signals Classification for Arrhythmia Detection using Decision Trees. In *2020 Fourth International Conference on Inventive Systems and Control (ICISC)*, pages 192–196, 2020. doi:10.1109/ICISC47916.2020.9171084.

[139] G. Lenis, N. Pilia, A. Loewe, W. H. W. Schulze, and O. Dössel. Comparison of Baseline Wander Removal Techniques considering the preservation of ST changes in the Ischemic ECG: A simulation Study. *Computational and Mathematical Methods in Medicine*, 2017.

[140] H.-Y. Lin, S.-Y. Liang, Y.-L. Ho, Y.-H. Lin, and H.-P. Ma. Discrete-wavelet-transform-based noise removal and feature extraction for ecg signals. *Irbm*, 35(6): 351–361, 2014.

[141] Z. C. Lipton. The Mythos of Model Interpretability: In Machine Learning, the Concept of Interpretability is Both Important and Slippery. *Queue*, 16(3):31–57, 2018.

[142] G. Litjens, F. Ciompi, J. M. Wolterink, B. D. de Vos, T. Leiner, J. Teuwen, and I. Išgum. State-of-the-Art Deep Learning in Cardiovascular Image Analysis. *JACC Cardiovasc Imaging*, 12(8 Pt 1):1549–1565, August 2019. doi:10.1016/j.jcmg.2019.06.009.

[143] B. Liu, J. Liu, G. Wang, K. Huang, F. Li, Y. Zheng, Y. Luo, and F. Zhou. A Novel Electrocardiogram Parameterization Algorithm and Its Application in Myocardial Infarction Detection. *Computers In Biology And Medicine*, 61, 2014,8.

[144] M. Liu, S. Ren, S. Ma, J. Jiao, Y. Chen, Z. Wang, and W. Song. Gated Transformer Networks for Multivariate Time Series Classification. *arXiv preprint*, 2021.

[145] N. Liu, L. Wang, Q. Chang, Y. Xing, and X. Zhou. A Simple and Effective Method for Detecting Myocardial Infarction Based on Deep Convolutional Neural Network. *Journal Of Medical Imaging And Health Informatics*, 8:1508–1512, 2018,9.

[146] N. Lu, Y. Wu, L. Feng, and J. Song. Deep Learning for Fall Detection: Three-Dimensional CNN Combined With LSTM on Video Kinematic Data. *IEEE Journal of Biomedical and Health Informatics*, 23(1):314–323, 2019. doi:10.1109/JBHI.2018.2808281.

[147] A. S. Lundervold and A. Lundervold. An overview of deep learning in medical imaging focusing on mri. *Zeitschrift für Medizinische Physik*, 29(2):102 – 127, 20199. ISSN 0939-3889. doi:https://doi.org/10.1016/j.zemedi.2018.11.002. Special Issue: Deep Learning in Medical Physics.

[148] H. M. Lynn, S. B. Pan, and P. Kim. A deep bidirectional gru network model for biometric electrocardiogram classification based on recurrent neural networks. *IEEE Access*, 7:145395–145405, 2019. doi:10.1109/ACCESS.2019.2939947.

[149] S. Mallat. Geometrical Grouplets. *Applied and Computational Harmonic Analysis*, 26(2):161–180, 2009. ISSN 1063-5203. doi:https://doi.org/10.1016/j.acha.2008.03.004. URL https://www.sciencedirect.com/science/article/pii/S1063520308000444.

[150] S. Mallat and S. Zhong. Characterization of Signals from Multiscale Edges. *IEEE Transactions On Pattern Analysis And Machine Intelligence*, 14:710–732, 1992.

[151] S. G. Mallat. Multiresolution Approximations and Aavelet orthonormal Bases of $L^2(R)$. *Transactions of the American mathematical society*, 315(1):69–87, 1989.

[152] Y. Mallet, D. Coomans, J. Kautsky, and O. De Vel. Classification Using Adaptive Wavelets for Feature Extraction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(10):1058–1066, 1997. doi:10.1109/34.625106.

[153] K. Manivel and R. S. Ravindran. Noise Removal for Baseline wander and power line in Electrocardiograph Signals. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, February, 2015.

[154] H. Martin, U. Morar, W. Izquierdo, M. Cabrerizo, A. Cabrera, and M. Adjouadi. Real-time Frequency-Independent Single-Lead and Single-Beat Myocardial Infarction Detection. *Artificial Intelligence In Medicine*, 121:102179, 2021. URL https://www.sciencedirect.com/science/article/pii/S093336572100172X.

[155] R. J. Martis, U. R. Acharya, and L. C. Min. Ecg beat classification using pca, lda, ica and discrete wavelet transform. *Biomedical Signal Processing and Control*, 8(5):437–448, 2013. ISSN 1746-8094. doi:https://doi.org/10.1016/j.bspc.2013.01.005. URL https://www.sciencedirect.com/science/article/pii/S1746809413000062.

[156] F. Martínez-Plumed, L. Contreras-Ochando, C. Ferri, J. Hernández-Orallo, M. Kull, N. Lachiche, M. Ramírez-Quintana, and P. Flach. Crisp-dm twenty years later: From data mining processes to data science trajectories. *IEEE Transactions On Knowledge And Data Engineering*, 33:3048–3061, 2021.

[157] S. M. Mathews. *Dictionary and Deep Learning Algorithms with Applications to Remote Health Monitoring Systems*. PhD thesis, University of Delaware, 2017. URL http://udspace.udel.edu/handle/19716/21241. PhD thesis.

[158] S. M. Mathews, C. Kambhamettu, and K. E. Barner. A Novel Application of Deep Learning for Single-lead ECG Classification. *Comput Biol Med*, 99:53–62, August 1 2018. doi:10.1016/j.compbiomed.2018.05.013.

[159] MathWorks. Classify Time Series Using Wavelet Analysis and Deep Learning. https://de.mathworks.com/help/wavelet/examples/signal-classification-with-wavelet-analysis-and-convolutional-neural-networks.html, 2018. Retrieved on 2018-08-20.

[160] MathWorks. Visualize activations of a convolutional neural network. https://de.mathworks.com/help/deeplearning/ug/visualize-activations-of-a-convolutional-neural-network.html, 2018. Retrieved on 2018-12-10.

[161] Md Shahiduzzaman. Fall detection by Accelerometer and Heart Rate Variability Measurement. *Global Journal of Computer Science and Technology: G Interdisciplinary*, 2015.

[162] P. Melillo, R. Castaldo, G. Sannino, A. Orrico, G. de Pietro, and L. Pecchia. Wearable Technology and ECG Processing for Fall Risk Assessment, Prevention and Detection. *Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual International Conference*, 2015:7740—7743, 2015. ISSN 2375-7477. doi:10.1109/embc.2015.7320186. URL https://doi.org/10.1109/EMBC.2015.7320186.

[163] L. J. Mena, V. G. Félix, A. Ochoa, R. Ostos, E. González, J. Aspuru, P. Velarde, and G. E. Maestre. Mobile personal health monitoring for automated classification of electrocardiogram signals in elderly. *Comput Math Methods Med*, 2018: 9128054, 2018. ISSN 1748-6718 (Electronic); 1748-670X (Print); 1748-670X (Linking). doi:10.1155/2018/9128054.

[164] L. Meng, W. Tan, J. Ma, R. Wang, X. Yin, and Y. Zhang. Enhancing Dynamic ECG Heartbeat Classification with Lightweight Transformer Model. *ARTIFICIAL INTELLIGENCE IN MEDICINE*, 124, FEB 2022. ISSN 0933-3657. doi:10.1016/j.artmed.2022.102236.

[165] E. Merdjanovska and A. Rashkovska. Comprehensive survey of computational ecg analysis: Databases, methods and applications. *Expert Systems with Applications*, 203:117206, 2022. ISSN 0957-4174. doi:https://doi.org/10.1016/j.eswa.2022.117206. URL https://www.sciencedirect.com/science/article/pii/S0957417422005917.

[166] D. Michie, D. J. Spiegelhalter, and C. C. Taylor. Machine Learning, Neural and Statistical Classification. *Ellis Horwood Series in Artificial Intelligence*, 1994.

[167] A. Milchevski and M. Guse. Performance Evaluation of FIR and IIR Filtering of ECG Signals. *Computational and Mathematical Methods in Medicine*, 2016.

[168] Mitra, Sanjit K. *Digital Signal Processing: A Computer-Based Approach*. McGraw-Hill School Education Group, 2nd edition, 2001. ISBN 0072522615.

[169] M. Moavenian and H. Khorrami. A qualitative comparison of artificial neural networks and support vector machines in ecg arrhythmias classification. *Expert Systems with Applications*, 37(4):3088–3093, 2010. ISSN 0957-4174. doi:https://doi.org/10.1016/j.eswa.2009.09.021. URL https://www.sciencedirect.com/science/article/pii/S0957417409008021.

[170] Mohebbanaaz, L. V. R. Kumari, and Y. P. Sai. Classification of ecg beats using optimized decision tree and adaptive boosted optimized decision tree. *Signal, Image*

and Video Processing, 16(3):695–703, 2022. doi:10.1007/s11760-021-02009-x. URL https://doi.org/10.1007/s11760-021-02009-x.

[171] A. Natarajan, Y. Chang, S. Mariani, A. Rahman, G. Boverman, S. Vij, and J. Rubin. A Wide and Deep Transformer Neural Network for 12-Lead ECG Classification. In *2020 Computing in Cardiology*, pages 1–4, 2020. doi:10.22489/CinC.2020.107.

[172] Neena Damodaran and Jörg Schäfer. Device free human activity recognition using wifi channel state information. In *2019 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computing, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, pages 1069–1074. IEEE, August 2019. doi:10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00205.

[173] L. T. Nemirko A.P. Biometric Human Identification Based on Electrocardiogram. In *Proc. XII-th Russian Conference on Mathematical Methods of Pattern Recognition, Moscow*, pages 387–390, Moscow, 2005. MAKS Press.

[174] I. Neves, D. Folgado, S. Santos, M. Barandas, A. Campagner, L. Ronzio, F. Cabitza, and H. Gamboa. Interpretable Heartbeat Classification Using Local Model-Agnostic Explanations on ECGs. *Computers in Biology and Medicine*, 133:104393, 2021. ISSN 0010-4825. doi:https://doi.org/10.1016/j.compbiomed.2021.104393. URL https://www.sciencedirect.com/science/article/pii/S0010482521001876.

[175] O. Niaksu. Crisp data mining methodology extension for medical domain. *Balt. J. Mod. Comput.*, 3:92–109, 2015.

[176] R. T. Olszewski. *Generalized Feature Extraction for Structural Pattern Recognition in Time-Series Data*. PhD thesis, Carnegie Mellon University Pittsburgh, PA, 2001. URL https://api.semanticscholar.org/CorpusID:17004764.

[177] F. Ordóñez and D. Roggen. Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition. *Sensors*, 16:115, 2016. doi:10.3390/s16010115. URL https://doi.org/10.3390/s16010115.

[178] S. J. Pan and Q. Yang. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010. doi:10.1109/TKDE.2009.191.

[179] E. B. Panganiban, A. C. Paglinawan, W. Y. Chung, and G. L. S. Paa. Ecg diagnostic support system (edss): A deep learning neural network based

classification system for detecting ecg abnormal rhythms from a low-powered wearable biosensors. *Sensing and Bio-Sensing Research*, 31:100398, 2021. ISSN 2214-1804. doi:https://doi.org/10.1016/j.sbsr.2021.100398. URL https://www.sciencedirect.com/science/article/pii/S2214180421000039.

[180] K. Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 6(2):559–572, 1901.

[181] J. "Pennington, R. Socher, and C. Manning. "GloVe: Global vectors for word representation". In *"Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)"*, pages "1532–1543", "Doha, Qatar", oct "2014". "Association for Computational Linguistics". doi:"10.3115/v1/D14-1162".

[182] J. Pereira and M. Silveira. Learning Representations from Healthcare Time Series Data for Unsupervised Anomaly Detection. In *2019 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 1–7, 2019. doi:10.1109/BIGCOMP.2019.8679157.

[183] R. S. Peres, X. Jia, J. Lee, K. Sun, A. W. Colombo, and J. Barata. Industrial Artificial Intelligence in Industry 4.0 - Systematic Review, Challenges and Outlook. *IEEE Access*, 8:220121–220139, 2020. doi:10.1109/ACCESS.2020.3042874.

[184] M. Phuong and M. Hutter. Formal Algorithms for Transformers. *arXiv preprint arXiv:2207.09238*, 2022.

[185] V. Plotnikova, M. Dumas, and F. Milani. Adaptations of data mining methodologies: A systematic literature review. *PeerJ Comput Sci*, 6:e267, 2021. doi:10.7717/peerj-cs.267. Published on May 25.

[186] O. J. Prieto, C. J. Alonso-González, and J. J. Rodríguez. Stacking for multivariate time series classification. *Pattern Analysis and Applications*, 18(2):297–312, 2015. doi:10.1007/s10044-013-0351-9. URL https://doi.org/10.1007/s10044-013-0351-9.

[187] A. Purbasari, F. Rinawan, A. Zulianto, A. Susanti, and H. Komara. Crisp-dm for data quality improvement to support machine learning of stunting prediction in infants and toddlers. In *8th International Conference on Advanced Informatics: Concepts, Theory and Applications (ICAICTA)*, pages 1–6, 2021. doi:10.1109/ICAICTA53211.2021.9640294.

[188] K. Pusch. ECG Classification Using Different Machine Learning Models for Human Activity Recognition. Bachelor thesis, Frankfurt University of Applied Sciences, 2021.

[189] N. Qian. On the momentum term in gradient descent learning algorithms. *Neural Networks*, 12(1):145 – 151, 1999. ISSN 0893-6080. doi:https://doi.org/10.1016/S0893-6080(98)00116-6. URL http://www.sciencedirect.com/science/article/pii/S0893608098001166.

[190] M. N. Rabe and C. Staats. Self-attention Does Not Need $O(n^2)$ Memory. *arXiv preprint*, 2022.

[191] H. Rai and K. Chatterjee. Hybrid CNN-LSTM Deep Learning Model and Ensemble Technique for Automatic Detection of Myocardial Infarction Using Big ECG Data. *Applied Intelligence*, 52:5366–5384, 2022. URL https://doi.org/10.1007/s10489-021-02696-6.

[192] J. J. Rajan. *Time Series Classfication*. PhD thesis, University of Cambridge, 1994. Ph.D. thesis.

[193] K. N. Rajesh and R. Dhuli. Classification of ECG Heartbeats Using Non-linear Decomposition Methods and Support Vector Machine. *Computers in Biology and Medicine*, 87:271–284, 2017. ISSN 0010-4825. doi:https://doi.org/10.1016/j.compbiomed.2017.06.006. URL https://www.sciencedirect.com/science/article/pii/S0010482517301701.

[194] H. Ramsauer, B. Schäfl, J. Lehner, P. Seidl, M. Widrich, T. Adler, L. Gruber, M. Holzleitner, M. Pavlović, G. Sandve, V. Greiff, D. Kreil, M. Kopp, G. Klambauer, J. Brandstetter, and S. Hochreiter. Hopfield Networks is All You Need. *arXiv preprint*, 2021.

[195] T. Reasat and C. Shahnaz. Detection of Inferior Myocardial Infarction Using Shallow Convolutional Neural Networks. In *2017 IEEE region 10 humanitarian technology conference (R10-HTC)*, pages 718–721. IEEE, 2017.

[196] R. Remya, K. Indiradevi, and K. Babu. Classification of Myocardial Infarction Using Multi Resolution Wavelet Analysis of ECG. *Procedia Technology*, 24:949–956, 2016,12.

[197] M. T. Ribeiro, S. Singh, and C. Guestrin. Why should I Trust You?Explaining the Predictions of any Classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

[198] H. Robbins and S. Monro. A stochastic approximation method. *Annals of Mathematical Statistics*, 22:400–407, 1951.

[199] J. Rosa-Bilbao, F. S. Butt, D. Merkl, M. F. Wagner, J. Schäfer, and J. Boubeta-Puig. In *IoT-based Indoor Air Quality Management System for Intelligent Education Environments*, 2024. Submitted.

[200] C. S. and R. E. A Novel Deep Learning based Gated Recurrent Unit with Extreme Learning Machine for Electrocardiogram (ECG) Signal Recognition. *Biomedical Signal Processing And Control*, 68:102779, 2021. URL https://www.sciencedirect.com/science/article/pii/S1746809421003761.

[201] S. Saadatnejad, M. Oveisi, and M. Hashemi. Lstm-based ecg classification for continuous monitoring on personal wearable devices. *IEEE JOURNAL OF BIOMEDICAL AND HEALTH INFORMATICS*, 24(2):515–523, FEB 2020. ISSN 2168-2194. doi:10.1109/JBHI.2019.2911367.

[202] S. Saadatnejad, M. Oveisi, and M. Hashemi. LSTM-Based ECG Classification for Continuous Monitoring on Personal Wearable Devices. *IEEE Journal of Biomedical and Health Informatics*, 24(2):515–523, 2020. doi:10.1109/JBHI.2019.2911367.

[203] Y. Saeys, I. Inza, and P. Larrañaga. A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507–2517, 08 2007. ISSN 1367-4803. doi:10.1093/bioinformatics/btm344. URL https://doi.org/10.1093/bioinformatics/btm344.

[204] N. Safdarian, N. Dabanloo, and G. Attarodi. A New Pattern Recognition Method for Detection and Localization of Myocardial Infarction Using T-Wave Integral and Total Integral as Extracted Features from One Cycle of ECG Signal. *Journal of Biomedical Science and Engineering*, 7:818–824, 2014.

[205] S. Sahoo, A. Subudhi, M. Dash, and S. Sabut. Automatic classification of cardiac arrhythmias based on hybrid features and decision tree algorithm. *International Journal of Automation and Computing*, 17(4):551–561, 2020. doi:10.1007/s11633-019-1219-2. URL https://doi.org/10.1007/s11633-019-1219-2.

[206] M. Sanjit K. *Digital Signal Processing*. McGraw-Hill, 1998.

[207] M. Sarfraz, A. A. Khan, and F. F. Li. Using independent component analysis to obtain feature space for reliable ecg arrhythmia classification. In *2014 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 62–67, 2014. doi:10.1109/BIBM.2014.6999249.

[208] L. Sathyapriya, L. Murali, and T. Manigandan. Analysis and Detection R-peak Detection Using Modified Pan-Tompkins Algorithm. In *2014 IEEE International Conference On Advanced Communications, Control And Computing Technologies*, pages 483–487, 2014.

[209] Y. Sattar and L. Chhabra. Electrocardiogram. In *StatPearls [Internet]*. StatPearls Publishing, 2022.

[210] J. Schäfer. Human Activity Recognition With CSI Data – Attention Is All You Need. pre-print, 2022.

[211] J. Schäfer. A Short Note on Attention Embeddings. preprint, 2023.

[212] C. Schröer, F. Kruse, and J. M. Gómez. A systematic literature review on applying crisp-dm process model. *Procedia Computer Science*, 181:526–534, 2021. doi:10.1016/j.procs.2021.01.199.

[213] K. Schwab. *The Fourth Industrial Revolution*. Portfolio Penguin, 1st edition, January 5 2017.

[214] J. Schäfer. CSI Human Activity, 2021. URL https://dx.doi.org/https://doi.org/10.3390/app11198860.

[215] J. Schäfer, B. R. Barrsiwal, M. Kokhkharova, H. Adil, and J. Liebehenschel. Human Activity Recognition Using CSI Information with Nexmon. *Applied Sciences*, 11(19), 2021. ISSN 2076-3417. doi:10.3390/app11198860. URL https://www.mdpi.com/2076-3417/11/19/8860.

[216] Seema rani, Amanpreet kaur,J S Ubhi. Comparative Study of FIR and IIR Filters for the Removal of Baseline Noises From ECG Signal. *International Journal of computer Science and Information Technologies*, pages 1105–1108, 2011.

[217] N. Seidle. Analog to Digital Conversion, 2013. URL https://learn.sparkfun.com/tutorials/analog-to-digital-conversion/all. Retrieved on 2018-12-2.

[218] A. Serban, K. van der Blom, H. Hoos, and J. Visser. Software Engineering Practices for Machine Learning — Adoption, Effects, and Team Assessment. *Journal of Systems and Software*, 209:111907, 2024. ISSN 0164-1212. doi:https://doi.org/10.1016/j.jss.2023.111907. URL https://www.sciencedirect.com/science/article/pii/S0164121223003023.

[219] S. M. Shankaranarayana and D. Runje. Attention Augmented Convolutional Transformer for Tabular Time-series. In *2021 International Conference on Data Mining Workshops (ICDMW)*, pages 537–541, 2021. doi:10.1109/ICDMW53433.2021.00071.

[220] L. Sharma and R. Sunkaria. Inferior Myocardial Infarction Detection Using Stationary Wavelet Transform and Machine Learning Approach. *Signal, Image And Video Processing*, 12:199–206, 2018.

[221] L. N. Sharma, S. Dandapat, and A. Mahanta. Multichannel ecg data compression based on multiscale principal component analysis. *IEEE Transactions on Information Technology in Biomedicine*, 16(4):730–736, 2012. doi:10.1109/TITB.2012.2195322.

[222] V. Sharma, A. Stranieri, J. Ugon, P. Vamplew, and L. Martin. An agile group aware process beyond crisp-dm: A hospital data mining case study. In *Proceedings of the International Conference on Compute and Data Analysis (ICCDA '17)*, pages 109–113, 2017. doi:10.1145/3093241.3093273.

[223] S.-Y. Shih, F.-K. Sun, and H. yi Lee. Temporal Pattern Attention for Multivariate Time Series Forecasting. *Machine Learning*, 108:1421 – 1441, 2018. URL https://api.semanticscholar.org/CorpusID:52196634.

[224] H. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. J. Mollura, and R. M. Summers. Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. *CoRR*, abs/1602.03409, 2016. URL http://arxiv.org/abs/1602.03409.

[225] A. K. Singh and S. Krishnan. Ecg signal feature extraction trends in methods and applications. *BioMedical Engineering OnLine*, 22(1):22, 2023. doi:10.1186/s12938-023-01075-1. URL https://doi.org/10.1186/s12938-023-01075-1.

[226] R. Socher, B. Huval, B. Bath, C. Manning, and A. Ng. Convolutional-Recursive Deep Learning for 3D Object Classification. In *Advances In Neural Information Processing Systems*, volume 25, 2012. URL https://proceedings.neurips.cc/paper/2012/file/3eae62bba9ddf64f69d49dc48e2dd214-Paper.pdf.

[227] H. Song, D. Rajan, J. Thiagarajan, and A. Spanias. Attend and Diagnose: Clinical Time Series Analysis Using Attention Models. *32nd AAAI Conference On Artificial Intelligence, AAAI 2018*, pages 4091–4098, 2018.

[228] H. Song, D. Rajan, J. Thiagarajan, and A. Spanias. Attend and diagnose: Clinical time series analysis using attention models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

[229] K. Stapor, P. Ksieniewicz, S. García, and M. Woźniak. How to design the fair experimental classifier evaluation. *Applied Soft Computing*, 104:107219, 2021. ISSN 1568-4946. doi:https://doi.org/10.1016/j.asoc.2021.107219. URL https://www.sciencedirect.com/science/article/pii/S1568494621001423.

[230] V. Strassen. Gaussian Elimination is not Optimal. *Numerische Mathematik*, 13 (4):354–356, August 1969. doi:10.1007/BF02165411.

[231] N. Strodthoff and C. Strodthoff. Detecting and Interpreting Myocardial Infarction Using Fully Convolutional Neural Networks. *Physiological Measurement*, 40: 015001, 2019,1. doi:10.1088/1361-6579/aaf34d. URL http://dx.doi.org/10.1088/1361-6579/aaf34d.

[232] N. Strodthoff, P. Wagner, T. Schaeffter, and W. Samek. Deep Learning for ECG Analysis: Benchmarks and Insights from PTB-XL. *IEEE Journal Of Biomedical And Health Informatics*, 25:1519–1528, 2021.

[233] S. Studer, T. Bui, C. Drescher, A. Hanuschkin, L. Winkler, S. Peters, and K.-R. Müller. Towards crisp-ml(q): A machine learning process model with quality assurance methodology. *Machine Learning and Knowledge Extraction*, 3(2):392–413, 2021. doi:10.3390/make3020020.

[234] L. Sun, Y. Lu, K. Yang, and S. Li. ECG Analysis Using Multiple Instance Learning for Myocardial Infarction Detection. *IEEE Transactions On Biomedical Engineering*, 59:3348–3356, 2012.

[235] G. A. Susto, A. Schirru, S. Pampuri, S. McLoone, and A. Beghi. Machine Learning for Predictive Maintenance: A Multiple Classifier Approach. *IEEE Transactions on Industrial Informatics*, 11(3):812–820, June 2015. doi:10.1109/TII.2014.2349359.

[236] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Computer Vision and Pattern Recognition (CVPR)*, 2015. URL http://arxiv.org/abs/1409.4842.

[237] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu. A Survey on Deep Transfer Learning. In V. Kůrková, Y. Manolopoulos, B. Hammer, L. Iliadis, and I. Maglogiannis, editors, *Artificial Neural Networks and Machine Learning –ICANN 2018*, pages 270–279, Cham, 2018. Springer International Publishing. ISBN 978-3-030-01424-7.

[238] M. Tan and R. Kenny. Cardiovascular Assessment of Falls in Older People. *Clinical Interventions In Aging*, 1:57–66, 2006,2.

[239] M. Tan and Q. V. Le. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *ArXiv*, abs/1905.11946, 2019. URL https://api.semanticscholar.org/CorpusID:167217261.

[240] A. Theissler, F. Spinnato, U. Schlegel, and R. Guidotti. Explainable AI for Time Series Classification: A Review, Taxonomy and Research Directions. *IEEE Access*, 10:100700–100724, 2022. doi:10.1109/ACCESS.2022.3207765.

[241] J. Thickstun. Technical Report: The Transformer Model in Equations. Technical report, University of Washington, 2020. URL https://johnthickstun.com/docs/transformers.pdf.

[242] M. E. Tinetti. Factors Associated with Serious Injury During Falls by Ambulatory Nursing Home Residents. *Journal of the American Geriatrics Society*, 35(7):644–648, 1987. doi:10.1111/j.1532-5415.1987.tb04341.x. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1532-5415.1987.tb04341.x.

[243] I. Tobore, J. Li, A. Kandwal, L. Yuhang, Z. Nie, and L. Wang. Statistical and spectral analysis of ecg signal towards achieving non-invasive blood glucose monitoring. *BMC Med Inform Decis Mak*, 19(Suppl 6):266, Dec 2019. ISSN 1472-6947 (Electronic); 1472-6947 (Linking). doi:10.1186/s12911-019-0959-9.

[244] C. Tsai, C. Lai, H. Chao, and A. Vasilakos. Big data analytics: A survey. *Journal of Big Data*, 2015. doi:10.1186/s40537-015-0030-3. SAS Institute Inc. 2017. SAS Enterprise MinerTM 14.3: Reference Help. Cary: SAS Institute Inc. DOI 10.1186/s40537-015-0030-3.

[245] A. Turan, Ö. Barshan, and B. Barshan. Detecting Falls with Wearable Sensors Using Machine Learning Technique. *Sensors 2014, 14(6), 10691-10708*, 2014.

[246] A. N. Uwaechia and D. A. Ramli. A comprehensive survey on ecg signals as new biometric modality for human authentication: Recent advances and future challenges. *IEEE Access*, 9:97760–97802, 2021. doi:10.1109/ACCESS.2021.3095248.

[247] M. Uyar, S. Yildirim, and M. T. Gencoglu. An Effective Wavelet-Based Feature Extraction Method for Classification of Power Quality Disturbance Signals. *Electric Power Systems Research*, 78(10):1747–1755, 2008. ISSN 0378-7796. doi:https://doi.org/10.1016/j.epsr.2008.03.002. URL https://www.sciencedirect.com/science/article/pii/S0378779608000953.

[248] R. Øyvind. *Linear Algebra, Signal Processing, and Wavelets - A Unified Approach: Python Version*. Springer, 2019,1. ISBN 978-3-030-02939-5.

[249] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, and I. Polosukhin. Attention is All you Need. *Advances In Neural Information Processing Systems*, 30, 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[250] J. Venter, A. de Waal, and C. Willers. Specializing crisp-dm for evidence mining. In *Proceedings of the IFIP International Conference on Digital Forensics*, pages 303–315, Orlando, FL, USA, 30 January–1 February 2007.

[251] P. Wagner, N. Strodthoff, R. Bousseljot, W. Samek, and T. Schaeffter. PTB-XL, a Large Publicly Available Electrocardiography Dataset (version 1.0.0). *PhysioNet*, 2020. URL https://doi.org/10.13026/qgmg-0d46.

[252] J. Wang, X. Qiao, C. Liu, X. Wang, Y. Liu, L. Yao, and H. Zhang. Automated ECG Classification Using a Non-Local Convolutional Block Attention Module. *Computer Methods And Programs In Biomedicine*, 203:106006, 2021. URL https://www.sciencedirect.com/science/article/pii/S016926072100081X.

[253] S. Wang, B. Li, M. Khabsa, H. Fang, and H. Ma. Linformer: Self-Attention with Linear Complexity. *arXiv preprint arXiv:2006.04768*, 2020.

[254] T. Wang, C. Lu, Y. Sun, M. Yang, C. Liu, and C. Ou. Automatic ECG Classification Using Continuous Wavelet Transform and Convolutional Neural Network. *Entropy*, 23, 2021. URL https://www.mdpi.com/1099-4300/23/1/119.

[255] Y. Wang, X. Jiang, R. Cao, and X. Wang. Robust indoor human activity recognition using wireless signals. *Sensors (Basel, Switzerland)*, 15(7):17195–17208, 07 2015. doi:10.3390/s150717195. URL https://www.ncbi.nlm.nih.gov/pubmed/26184231.

[256] Z. Wang, V. Ramamoorthy, U. Gal, and A. Guez. Possible Life Saver: A Review on Human Fall Detection Technology. *Robotics*, 9(3):55, Jul 2020. ISSN 2218-6581. doi:10.3390/robotics9030055. URL http://dx.doi.org/10.3390/robotics9030055.

[257] W. Wei. *Time Series Analysis: Univariate and Multivariate Methods*. Pearson, 2006. ISBN 0-321-32216-9.

[258] Q. Wen, T. Zhou, C. Zhang, W. Chen, Z. Ma, J. Yan, and L. Sun. Transformers in Time Series: A Survey. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI 2023, 19th-25th August 2023, Macao, SAR, China*, pages 6778–6786. ijcai.org, 2023. doi:10.24963/IJCAI.2023/759. URL https://doi.org/10.24963/ijcai.2023/759.

[259] D. Wild and B. Isaacs, Nayak. How Dangerous are Falls in Old People at Home? *British Medical Journal, volume 282*, 1981.

[260] R. Wirth and J. Hipp. Crisp-dm: Towards a standard process model for data mining. In *Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining*, volume 1, pages 29–39, 2000.

[261] D. Wolpert and W. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997. doi:10.1109/4235.585893.

[262] World Health Organization. WHO Global Report on Falls Prevention in Older Age, 2007. URL https://www.who.int/ageing/publications/Falls_prevention7March.pdf. Last Checked on 2020-12-01.

[263] K. Xia, J. Huang, and H. Wang. LSTM-CNN Architecture for Human Activity Recognition. *IEEE Access*, 8:56855–56866, 2020. doi:10.1109/ACCESS.2020.2982225.

[264] G. Yan, S. Liang, Y. Zhang, and F. Liu. Fusing Transformer Model with Temporal Features for ECG Heartbeat Classification. In *2019 IEEE International Conference On Bioinformatics And Biomedicine (BIBM)*, pages 898–905, 2019.

[265] F. Yang, G. Wang, C. Luo, and Z. Ding. Improving Automatic Detection of ECG Abnormality with Less Manual Annotations using Siamese Network. In *2021 43rd Annual International Conference Of The IEEE Engineering In Medicine Biology Society (EMBC)*, pages 1120–1123, 2021.

[266] G. Yen and K.-C. Lin. Wavelet Packet Feature Extraction for Vibration Monitoring. *IEEE Transactions on Industrial Electronics*, 47(3):650–667, 2000. doi:10.1109/41.847906.

[267] O. Yildirim, P. Plawiak, R.-S. Tan, and U. R. Acharya. Arrhythmia Detection Using Deep Convolutional Neural Network with Long Duration ECG Signals. *COMPUTERS IN BIOLOGY AND MEDICINE*, 102:411–420, NOV 1 2018. ISSN 0010-4825. doi:10.1016/j.compbiomed.2018.09.009.

[268] O. Yildirim, U. B. Baloglu, R.-S. Tan, E. J. Ciaccio, and U. R. Acharya. A New Approach for Arrhythmia Classification Using Deep Coded Features and LSTM Networks. *COMPUTER METHODS AND PROGRAMS IN BIOMEDICINE*, 176: 121–133, JUL 2019. ISSN 0169-2607. doi:10.1016/j.cmpb.2019.05.004.

[269] Ö. Yildrim, P. P ławiak, R. S. Tan, and U. R. Acharya. Arrhythmia Detection Using Deep Convolutional Neural Network with Long Duration ECG Signals. *Comput Biol Med*, 102:411–420, November 1 2018. doi:10.1016/j.compbiomed.2018.09.009.

[270] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3320–3328. Curran Associates, Inc., 2014. URL http://papers.nips.cc/paper/5347-how-transferable-are-features-in-deep-neural-networks.pdf.

[271] Y. Yuan and L. Lin. Self-Supervised Pretraining of Transformers for Satellite Image Time Series Classification. *IEEE Journal of Selected Topics in*

Applied Earth Observations and Remote Sensing, 14:474–487, 2021. doi:DOI: 10.36227/techrxiv.13025039.v3.

[272] G. Zerveas, S. Jayaraman, D. Patel, A. Bhamidipaty, and C. Eickhoff. A Transformer-based Framework for Multivariate Time Series Representation Learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining (KDD '21)*, pages 2114–2124, New York, NY, USA, 2021. Association for Computing Machinery. doi:10.1145/3447548.3467401.

[273] G. Zewdie and M. Xiong. Fully Automated Myocardial Infarction Classification using Ordinary Differential Equations. *arXiv preprint*, 2014. URL `https://arxiv.org/abs/1410.6984`.

[274] L. Zhang, H. Peng, and C. Yu. An Approach for ECG Classification Based on Wavelet Feature Extraction and Decision Tree. In *2010 International Conference on Wireless Communications & Signal Processing (WCSP)*, pages 1–4, 2010. doi:10.1109/WCSP.2010.5633782.

[275] H. Zhao, J. Liu, H. Chen, J. Chen, Y. Li, J. Xu, and W. Deng. Intelligent diagnosis using continuous wavelet transform and gauss convolutional deep belief network. *IEEE Transactions on Reliability*, 72(2):692–702, 2023. doi:10.1109/TR.2022.3180273.

[276] Q. Zhao and L. Zhang. Ecg feature extraction and classification using wavelet transform and support vector machines. In *2005 International Conference on Neural Networks and Brain*, volume 2, pages 1089–1092, 2005. doi:10.1109/ICNNB.2005.1614807.

[277] Y. Zheng, Q. Liu, E. Chen, Y. Ge, and J. Zhao. Time Series Classification Using Multi-Channels Deep Convolutional Neural Networks. *Web-Age Information Management*, 2014. doi:10.1007/978-3-030-64610-9_33.

[278] S. Śmigiel, K. Pałczyński, and D. Ledziński. Deep Learning Techniques in the Classification of ECG Signals Using R-Peak Detection Based on the PTB-XL Dataset. *Sensors*, 21, 2021. URL `https://www.mdpi.com/1424-8220/21/24/8174`.