YAŞAR UNIVERSITY

GRADUATE SCHOOL

MASTER’S THESIS

CONVOLUTIONAL AUTOENCODER BASED HEART ARRHYTHMIA DETECTION SYSTEM

ÖYKÜ ERAVCI

THESIS ADVISOR: ASSOC. PROF. (PHD) NALAN ÖZKURT

MSC. ELECTRICAL AND ELECTRONICS ENGINEERING

BORNOVA / İZMİR
AUGUST 2024

JURY APPROVAL PAGE

We certify that, as the jury, we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of master's in science.

Jury Members:	**Signature:**
Prof.(PhD) Mehmet ENGİN Ege University	....
Assoc.Prof.(PhD) Burhan GÜLBAHAR Yaşar University	....
Assoc.Prof.(PhD) Nalan ÖZKURT Yaşar University	....

Prof. (PhD) Yucel Ozturkoglu

Director of the Graduate School

ABSTRACT

CONVOLUTIONAL AUTOENCODER BASED

HEART ARRHYTHMIA DETECTION SYSTEM

Eravcı, Öykü
MSc, Electrical and Electronics Engineering
Advisor: Assoc. Prof. (PhD) Nalan Özkurt
August 2024

Remote monitoring of patients is essential for the early diagnosis of diseases and improving quality of life. The rapid development of deep learning techniques has significantly advanced wearable health technologies, making automatic diagnosis increasingly important. This study proposes a deep learning approach for classifying arrhythmias using a customized wavelet-based convolutional autoencoder (WBCAE) model for feature extraction and classification. The autoencoder model ingeniously combines the time-frequency domain examination capability of wavelets with the data-driven feature learning power of autoencoders.

This thesis targets the classification of distinct types of cardiac arrhythmias: normal sinus rhythm (NSR), right bundle branch block (RBBB), left bundle branch block (LBBB), atrial premature contraction (APC), and premature ventricular contraction (PVC), in addition to atrial fibrillation (AF). Two different autoencoder approaches were employed in this thesis. The first one is to use an autoencoder as an anomaly detector, where the autoencoder is trained with only normal samples and abnormal inputs produce higher reconstruction errors. The second method is to use an autoencoder as a feature extractor. In this approach, samples from all classes are used in the training. Then, the compressed representation obtained at the encoder layer output is used with a classifier.

The primary objectives of this thesis are to rigorously evaluate the performance of autoencoder-based deep learning algorithms and automate the classification of various

cardiac arrhythmias. The findings from our experiments underscore the importance of employing deep learning-based models in cardiac disease diagnosis, showcasing the immense potential of integrating wavelet methods with autoencoders in biomedical signal processing systems. This study substantially contributes to medical diagnostics by delivering a dependable tool for early disease detection and patient monitoring, ultimately advancing healthcare outcomes.

Keywords: Deep learning, Atrial Fibrillation (AF), Arrhythmia classification, Wavelet-based autoencoder, WBCAE model, Cardiac arrhythmias, Biomedical signal processing.

ÖZ

EVRİŞİMSEL OTOMATİK KODLAYICI TABANLI KALP ARİTMİSİ TESPİT SİSTEMİ

Eravcı, Öykü
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği
Danışman: Doç. Dr. Nalan Özkurt
Ağustos 2024

Hastaların uzaktan izlenmesi, hastalıkların erken teşhisi ve yaşam kalitesinin iyileştirilmesi açısından çok önemlidir. Derin öğrenme tekniklerinin hızlı gelişimi, giyilebilir sağlık teknolojilerini önemli ölçüde ilerletmiş ve otomatik teşhisi giderek daha önemli hale getirmiştir. Bu çalışmada, özellik çıkarımı ve sınıflandırma için özel bir dalgacık tabanlı konvolüsyonel otomatik kodlayıcı (WBCAE) modeli kullanan yenilikçi bir derin öğrenme yaklaşımı öneriyoruz. Otomatik kodlayıcı model, dalgacıkların zaman-frekans alanı inceleme yeteneğini, otokodlayıcıların veri odaklı özellik öğrenme gücüyle ustaca birleştirir.

Bu çalışma, normal sinüs ritmi (NSR), sağ dal bloğu (RBBB), sol dal bloğu (LBBB), atriyal prematüre kasılma (APC) ve prematüre ventriküler kasılma (PVC) gibi farklı tipte kardiyak aritmilerin yanı sıra atriyal fibrilasyonun (AF) sınıflandırılmasına odaklanmaktadır. Derin öğrenme tabanlı modelleri kullanarak aritmi sınıflandırmasının doğruluğunu ve verimliliğini önemli ölçüde artırmayı amaçlıyoruz, böylece erken hastalık tespiti ve kapsamlı hasta izleme için sağlam bir çerçeve sağlıyoruz.

Bu çalışmanın temel hedefleri, otomatik kodlayıcı tabanlı derin öğrenme algoritmalarının performansını titizlikle değerlendirmek ve çeşitli kardiyak aritmilerin sınıflandırılmasını otomatikleştirmektir. Deneylerimizden elde edilen bulgular, kardiyak hastalık teşhisinde derin öğrenme tabanlı modellerin kullanılmasının önemini vurgulamakta ve dalgacık yöntemlerinin otomatik kodlayıcılarla entegrasyonunun biyomedikal sinyal işleme sistemlerindeki büyük potansiyelini göstermektedir.

Bu çalışma, erken hastalık tespiti ve hasta izleme için güvenilir bir araç sunarak tıbbi teşhisler alanına önemli bir katkı sağlamakta ve nihayetinde sağlık sonuçlarını geliştirmektedir.

Anahtar Kelimeler: Derin öğrenme, Atriyal Fibrilasyon (AF), Aritmi sınıflandırma, Dalgacık tabanlı otomatik kodlayıcı, WBCAE modeli, Kardiyak aritmiler, Biyomedikal sinyal işleme.

ACKNOWLEDGEMENTS

I acknowledge my role as a Master’s researcher in the TÜBİTAK (Scientific and Technological Research Council of Turkey) Project no. 121E119, funded by the 1001 program grant, and This study also supported by the Project Evaluation Commission (PDK) within the scope of the project numbered BAP129, titled "Convolutional Autoencoder Based Cardiac Arrhythmia Detection System and FPGA Implementation."

I would like to express my deepest gratitude to my advisor, Dr. Nalan Ozkurt, for her exceptional guidance, mentorship, and support throughout my Master’s studies. Dr. Ozkurt's extensive knowledge, meticulous approach, and dedication were crucial at every stage of my project. Her insightful feedback and unwavering encouragement were instrumental in overcoming the challenges I faced and in developing my scientific thinking skills. Her patience and understanding significantly contributed to the successful completion of this thesis. I am profoundly grateful for her support and the invaluable impact she has had on my academic and personal growth.

Lastly, I would like to express my deepest gratitude to my family for their unwavering support and encouragement throughout my educational journey. Their patience, encouragement, and inspiration have been a constant source of strength. They stood by me through every challenge, celebrated every achievement, and their boundless love and faith have been crucial in reaching this milestone. I sincerely thank them for their dedication and understanding during this entire process.

Öykü Eravcı

İzmir, 2024

TEXT OF OATH

I declare and honestly confirm that my study, titled “CONVOLUTIONAL AUTOENCODER BASED HEART ARRHYTHMIA DETECTION SYSTEM” and presented as a Master’s Thesis, has been written without applying any assistance inconsistent with scientific ethics and traditions. I declare, to the best of my knowledge and belief, that all content and ideas drawn directly or indirectly from external sources are indicated in the text and listed in the list of references.

Öykü Eravcı

08.08.2024

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

SYMBOLS AND ABBREVIATIONS

ACN	Auto-Encoder Convolutional Network
AF	Atrial Fibrillation
APC	Atrial Premature Beats
AV	Atrioventricular Node
CAE	Convolutional Autoencoders
CNN	Convolutional Neural Networks
CVD	Cardiovascular Diseases
CWT	Continuous Wavelet Transforms
DAE	Denoising Autoencoder
DL	Deep Learning
DWT	Discrete Wavelet Transform
ECG	Electrocardiogram
EEG	Electroencephalogram
IDWT	Inverse Discrete Wavelet Transform
KNN	K-Nearest Neighbors
LBBB	Left Bundle Branch Block
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
ML	Machine Learning
MLP	Multi-Layer Perceptron
NSR	Normal Sinus Rhythm
PCA	Principal Component Analysis
PVC	Premature Ventricular Beats
RBBB	Right Bundle Branch Block
RELU	Rectified Linear Unit

RNN	Recurrent Neural Networks
SA	Sinoatrial Node
SGD	Stochastic Gradient Descent
SVM	Support Vector Machine
WBCAE	Wavelet-Based Convolutional Autoencoder
WHO	World Health Organization

1. CHAPTER: INTRODUCTION

1.1. Motivation and Background

Cardiovascular diseases (CVDs) rank among the top causes of mortality globally. The irregular heartbeat, otherwise known as arrhythmia, is vital because it has a significant effect on cardiovascular health, which makes the victim susceptible to heart attack. Arrhythmias can make the heartbeat too fast, too slow, or irregular, causing poor blood flow, hence severe consequences that may result. They disrupt the normal functioning of the heart, thus leading to various complications. For instance, specially arrhythmia called atrial fibrillation (AF) results in blood pooling in the atria, resulting in clot formation. These clots then get transported elsewhere in the body up to the brain, causing a stroke (American Heart Association, 2023).

Moreover, weak heart muscle due to arrhythmias can bring about heart failure with time, thus impairing the efficiency with which blood is being pumped. Notably, also ventricular tachycardia and ventricular fibrillation are mortal. These root in the lower chambers of the heart (ventricles) and can create a risk for sudden cardiac arrest if not treated urgently. Abrupt termination of heart beating, primarily due to ventricular fibrillation, is one of the most typical causes of myocardial infarction, leading to many deaths associated with it (Mayo Clinic, 2022). Furthermore, cardiac arrest is often induced by arrhythmias for instance ventricular tachycardia and fibrillation that result in the incapacity of the heart to maintain regular blood circulation (McComb, 2006). Therefore, patients with right bundle branch block (RBBB) and ST-segment elevation from V1 through V3 on electrocardiograms (ECG) who don't exhibit any obvious structural heart disease have an elevated risk of sudden death (Brugada et al., 1998).

To ensure the prevention of these diseases among individuals, especially those who are prone to developing arrhythmias, there should be a multidimensional approach. Lifestyle modifications are essential because they can significantly prevent this condition. For individuals at high risk, medical interventions such as medications, pacemakers, or defibrillators may be necessary to manage and prevent arrhythmias (Mayo Clinic, 2023).

In economical aspect, atrial fibrillation is a significant burden considering the substantial cost of hospital stays, treatments, and complications management. Among heart rhythm abnormalities, it is one of the most frequent ones that contribute to substantial healthcare expenses. According to McBride and Mattenklotz (2009), AF has direct costs, including hospitalizations, medications, and procedures, and indirect costs, including lost productivity and long-term care for disabilities or complications like stroke, which escalates as complications increase. In cases where patients have heart failure or stroke, which require costly and intensive interventions, admission periods can be prolonged and recurrent for AF patients. To prevent severe complications that necessitate emergency care and hospital admission. Atrial fibrillation (AF) has become one of the significant contributors to increased medical expenses in recent years. The costs related to AF arise from direct expenditures such as medication, hospitalization, and surgeries. On the other hand, there are indirect costs, such as loss of productivity by employees who have been incapacitated due to disabilities that result from arrhythmias-related strokes, among other causes. Protracted stays in the hospital may ensue, especially if a patient also suffers from congestive heart failure or when their stroke occurs at frequent intervals, thereby requiring expensive diagnostic tests plus therapies respectively.

The usual way of diagnosis of arrhythmias is to consider standard electrocardiograms (ECGs), and event recordings. This method has limited monitoring periods and occasionally misses intermittent arrhythmic events among patients who use them. In the case of one-day-long ECG recordings of Holter devices, Manually interpreting ECG data may require an extensive amount of time and subject to human error, leading to potential misdiagnoses (Chung et al., 2022). Therefore, the topic of automatic diagnosis methods is very current, and a lot of research is being conducted in this area.

Convolutional autoencoders (CAEs) represent a class of deep learning models with significant potential for various applications. Specifically, CAEs have demonstrated great promise in addressing specific challenges in anomaly detection for ECG data. The CAEs are of great advantage for processing and analyzing voluminous ECG data because of their capacity to learn hierarchical feature representations in an automated way. With the help of CAEs with greater computational power, the patient can continuously measure and wear the ECG signal so that extra information cannot be provided longer.

Furthermore, CAEs can induce a severe change in the precision of the arrhythmia detection process. These can be trained to understand signals that are small and not easy to pick up from the ECG data, which can lead to the elimination of minor neglected cases. These models can also be integrated with real-time monitoring systems to inform and alert healthcare providers who can act on the information, thus providing immediate intervention, which is essential for timely intervention. In addition to guaranteeing detection accuracy, CAEs can also automate the interpretation of the procedure, minimizing the workload on the medical professions, reducing human error, and so on, as well as the cost of service. The entailment of CAEs into the already existing arrhythmia detection frameworks can lead to the formation of more dependable and efficient diagnostic systems, ultimately leading to the cure of patients and the better use of the available resources in the healthcare sector.

1.2. Literature Review

Advanced deep learning (DL) and machine learning (ML) approaches have become essential in detecting arrhythmias. These techniques improve the efficiency of classifying arrhythmias from ECG data, which is crucial for early detection and management of cardiovascular diseases.

Historically, several traditional methods have been employed for arrhythmia detection, including:

- **Support Vector Machines (SVM):** SVMs are used to classify data by identifying the hyperplane in the feature space that efficiently classifies the different classes. They have been applied to ECG data to classify normal and abnormal heartbeats. However, this is limited by their performance because of the complexity of data and the extensive requirement for feature engineering.
- **Decision Trees:** Intuitively, decision trees divide data into branches to generate predictions. Overfitting can hinder their accuracy when applied to ECG data for arrhythmia detection, especially with high-dimensional data.
- **K Nearest Neighbors (KNN):** This algorithm classifies samples based on the majority class among its k-nearest neighbors in the feature space. Nevertheless,

despite being straightforward to implement, KNN faces challenges with large datasets and high-dimensional data to typical ECG signals.

- **Feedforward Neural Network (FNN):** A feedforward neural network (FNN) is a class of artificial neural network (ANN) distinguished by the direction of data flow between its layers. Its flow is unidirectional, which means that information in the model only flows forward from the input nodes, via the hidden nodes, and to the output nodes. AF detection was successfully done using a feed-forward neural network proposed by Chen et al. (2021) with an area under the receiver operating characteristic curve of 89.40%, a sensitivity of 84.26%, a specificity of 93.23%, and an accuracy of 84.00%. In addition, Cheng et al. (2020) proposed a direct detection method of AF from compressed ECGs with a roughly varying accuracy of 91%. 63% to 98. typically is at a rate of 40%, with the compression ratio ranging between 10% and 90%.

Although successful initially, these traditional methods often have to improve their precision and scalability when dealing with the vast volumes of data produced by contemporary ECG monitoring systems. This deficiency has resulted in the development of more sophisticated approaches.

Arrhythmia detection has recently been transformed by deep learning. Deep learning types such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs) have discovered complex patterns from ECG data.

- **Convolutional Neural Networks (CNNs):** CNNs continuously use input data to determine the spatial hierarchies of features, making them very effective in processing grid-like data such as images and time series data such as ECG signals. Acharya et al. (2017) achieved high precision in detecting arrhythmias, using a nine-layer CNN model to automatically identify five distinct heartbeat classes in original and noise-attenuated sets of ECG readings. The model correctly classified heartbeats in original and noise-free ECGs with an accuracy of 94.03% and 93.47%, respectively. Therefore, because they capture detailed patterns without manual feature extraction, CNNs are very well-suited for arrhythmia detection.

  Numerous studies are being conducted to enhance the efficacy and applicability of deep learning models to detect arrhythmia. Scholars are researching different architectures and training approaches to make these models more robust and generalizable. In addition, many patents have been filed for this kind of use in medical devices or applications. Substantial work has been done on innovation and patentable issues related to the application of CNNs in ECG arrhythmia detection. Kumar (2023) and Rajkumar (2019) developed CNN models for classifying ECG signals into different arrhythmia categories, explicitly mentioning developing a user-friendly web application. Ochiai (2018) suggested using a combination of CNN and denoising autoencoders to detect arrhythmia. The studies show prospects regarding patentable inventions by applying CNNs to ECG arrhythmia detection. There has been a focus on how accurate various atrial fibrillation (AF) classification methods are, including autoencoder convolutional neural networks (CNN), among other deep learning models. Hu et al. (2020) argued that AF classification is challenging because some features must be more discernable to avoid misclassification, thereby impeding poor results.

- **Recurrent Neural Networks (RNNs)**: Variants of recurrent neural networks (RNNs), particularly long short-term memory (LSTM) networks, are ideal for analyzing sequential data. They can incorporate temporal dependencies in the data as they are present inherently in time-series ECG signals. These models were applied to heartbeats modeling, which is usually a sequential signal, and the inclusion of temporal context seems fundamental when handling arrhythmia detection. This is a step forward in arrhythmia detection and an important shift from common Non-DL approaches. As a result, CNNs and RNNs are accurately analyzing complex ECG data and integrating it into wearable and remote monitoring systems, which are beginning to transform cardiac care. Ongoing research and technical improvements improve these models, promising significantly more reliable and efficient arrhythmia detection in the future.

This synthesized study aims to collectively discuss recent developments in applying deep learning-based models for arrhythmia and atrial fibrillation (AF) classification, referencing essential publications regarding methods and results. This study is among the first to leverage an autoencoder pipeline in combination with wavelets. Therefore, the success of wavelets in grasping the time-frequency domain distribution of the signals was integrated into the learning capability of autoencoders. Also, a large dataset collected from diverse databases was used for train and test to show the performance under various conditions.

Integrating these deep learning techniques into medical devices and applications is actively being pursued. Numerous projects and patents are focused on developing real-time arrhythmia detection systems that can be embedded in wearable devices and remote monitoring systems. These systems aim to provide continuous monitoring and timely alerts to healthcare providers, improving patient outcomes.

Developing convolutional neural networks (CNN) and autoencoders has opened up a new path for analyzing medical data, particularly regarding the arrhythmia detection field. Initially designed for image recognition, CNNs were applied to extract characteristics from time series, such as ECG signals. On the other hand, the application of autoencoders is realized through their reputation in data compression and extraction of features, which can be extended to anomaly detection. For the implementation, we aim to give health professionals a more reliable and efficient early arrhythmia detection tool based on CNN, autoencoders, and wavelet integration for robustness in the augmentation of accuracy through the reduction of false positives, potentially saving lives through timely interventions.

1.3. Aim of Study

The primary objective of this study is to examine and assess the efficacy of deep learning-based models in the automatic classification of cardiac arrhythmias and the diagnosis of atrial fibrillation (AF). This study seeks to address the growing need for accurate and efficient diagnostic tools in cardiology by leveraging advanced machine-learning techniques. The specific objectives of this study are outlined as follows:

- **Development of A Customized Wavelet-Based Autoencoder Model:** Unlike traditional methods that may rely solely on either time-domain or frequency-domain features, this study introduces a customized wavelet-based autoencoder (WBCAE) model. Therefore, a significant potential was offered for more detailed and relevant information extraction in the time-frequency domain.
- **Utilizing WBCAE for Feature Extraction:** The study offers a solution approach for deep feature extraction from ECG data sets utilizing wavelet-

based time series modeling. Following feature extraction, The MLP model is used for ECG beat classification, such that the results can fall into one of the different classes: normal heartbeats (NSR), right/left bundle branch block (RBBB/LBBB), atrial premature beats (APC), or premature ventricular beats (PVC). The proposed approach is evaluated on publicly available datasets to achieve high metrics across precision, accuracy, recall, and F1 score.

- **Using WBCAE in Anomaly Detection Mode:** We present a customized wavelet-based convolutional autoencoder (WBCAE) model as an anomaly detector for AF classification. The WBCAE model uses the combined capabilities of wavelet transformations and convolutional autoencoders to extract significant features from ECG signals. The WBCAE model is trained and validated on a comprehensive selection of publicly available datasets. The aim is to achieve robust diagnostic performance. Furthermore, the same model was tested for general-purpose anomaly detection for ECG arrhythmias. The WBCAE is trained with normal sinus rhythm beats and tested with different types of arrhythmia data to observe the performance of the anomaly detector.

By implementing the objectives listed above, we aim to achieve the following general objectives:

- **Enhancing Diagnostic Accuracy**: To achieve the ultimate goal of increasing accuracy rates in the medical area by deep learning algorithms. In these studies, we break current diagnostic limitations by employing wavelet-based feature selection with a contemporary neural network design.
- **Clinical Relevance**: These are relevant to the development of new, non-invasive diagnostic approaches for cardiology that may enable real-time monitoring and diagnosis of heart health.
- **Biomedical Engineering**: These studies contribute to the wider field of biomedical engineering by demonstrating the practical applications of deep learning models in medical diagnostics, delivering a framework for future research and development in automated disease identification and classification.

1.4. Thesis Outline

The contents of the chapters of the thesis are as follows:

The Introduction chapter presents the motivation and engineering problem, summarizes the literature on arrhythmia detection, and states the study's aims. The Overview chapter explains the theoretical background, including ECG, types of arrhythmias, and common techniques for arrhythmia detection and classification. The Experiments and Results section starts by introducing the WBCAE model, detailing wavelet transformation, convolutional autoencoders, and their application in arrhythmia detection. A summary of the main findings, a discussion of the consequences, and recommendations for further research are provided in the section titled Conclusion and Future Work.

2. CHAPTER: OVERVIEW

2.1. ECG

The heart is a critical organ whose major function is blood circulation throughout the body. It functions as a muscle pump, keeping blood flowing by recurring rhythmic contractions. The heart is made up of four chambers: the right and left atriums, as well as the right and left ventricles. In turn, the deoxygenated blood received from the body is passed on into the right ventricle for eventual transmission to the lungs for oxygenation. On the contrary, the left atrium receives blood that is already oxygenated from the lungs and passes it into the left ventricle, which further sends it around the body (Mohrman & Heller, 2023).

An intrinsic electrical conduction system controls the heart's rhythmic contractions. This system is essential for the coordinated contraction of the heart chambers and includes the following components and Figure 2.1. below illustrates the heart's conduction system:

- **Sinoatrial (SA) Node**: The SA node in the right atrium is commonly called the heart's natural pacemaker. It starts the electrical impulses that regulate the heart's contraction pace. The atrial walls contract in response to impulses sent by the SA node, forcing blood into the ventricles (Widmaier, Raff, & Strang, 2019).
- **Atrioventricular (AV) Node**: To allow the atria to complete their contraction before the ventricles are, it slightly delays the impulse. Because it assures that the ventricles are full of blood before they pump blood out, this delay is important (Lilly, 2016).
- **Bundle of His**: The Bundle of His allows the signal from the AV node to enter the ventricles. The electrical signals are subsequently transmitted down the heart's septum by these branches to the right and left bundle into the His bundle (Klabunde, 2021).

- **Right and Left Bundle Branches**: To ensure that the electrical activity reaches both sides of the heart at the same time, these branches deliver the electrical impulse to the appropriate ventricles (Klabunde, 2021).
- **Purkinje Fibers**: Purkinje fibers are a representation of the ventricular myocardium's spreading and splitting off from the bundle branches. Leading to a synchronized contraction that pumps blood from the heart to the lungs and the rest of the body, they rapidly conduct the electrical impulse to the ventricular muscle cells (Klabunde, 2021).

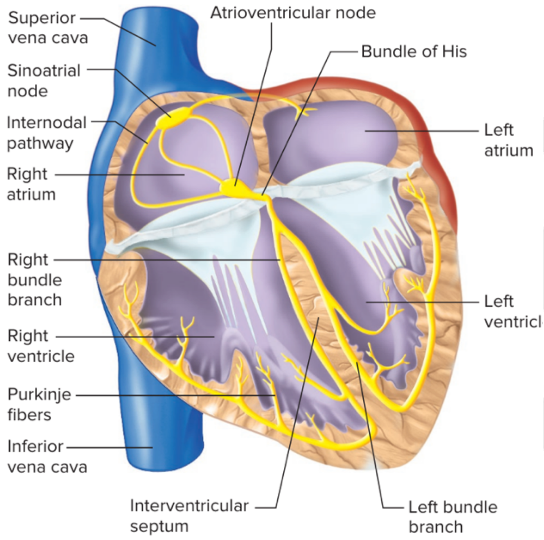


Figure 2.1. The heart's conduction system.

Source: Widmaier, Raff, & Strang, 2019

Electrocardiography (ECG) is the process of constantly recording the heart's electrical activity. This is accomplished by a noninvasive procedure in which electrodes are implanted in the skin to detect electrical impulses generated by the heart. The impulses are plotted on an ECG, which depicts the timing and amount of electrical signals as they travel through various areas of the heart. The ECG is critical for identifying cardiovascular illnesses such as arrhythmias, myocardial infarctions, and other cardiac irregularities.

The ECG trace consists of several characteristic waves and segments, each representing a distinct heart cycle phase:

- **P Wave**: On the ECG, atrial depolarization is shown by the P wave. It starts in the right atrium and then moves to the left atrium. Right atrial depolarization can be observed in the first half of the P wave, while left atrial depolarization is noticed in the second. The P wave’s duration is three small squares wide and 2.5 small squares high. Normal sinus rhythm is always positive in leads I and II, negative in lead aVR, and often biphasic in lead V1 (Sattar et al., 2020).
- **QRS Complex**: It represents ventricle depolarization because the impulse travels through the AV node. Usually, the duration of the QRS complex should be less than three small squares, or approximately 120 ms (usually between 60 and 100 ms). A widened QRS duration is a hallmark of hyperkalemia or a bundle branch block. Conversely, a broad QRS can indicate ventricular rhythm or a premature ventricular contraction.
- **Q Wave**: The initial negative deflection that might not always be present.
- **R Wave**: The initial positive deflection after the P wave.
- **S Wave**: The negative deflection following the R wave.

The QRS complex is significantly larger than the P wave due to the ventricles' greater muscle mass than the atria (Sattar et al., 2020).

- **T Wave**: It shows ventricular repolarization and is sensitive to various cardiac and non-cardiac factors, including hormonal and neurological influences. Usually, the T wave is positive in leads with tall R-waves (upward deflection). The usual criteria for normal T waves are considered to have a height of less than 10 mm and should be between one-eighth and two-thirds the size of R waves (Sattar et al., 2020).
- **PR Interval:** The duration between the beginning of the P wave and the beginning of the QRS complex is measured by this interval. It reflects the time the electrical impulse travels from the SA node through the atria, AV node, and His-Purkinje system to the ventricles (Goldberger, 2024).
- **ST Segment**: This segment is the flat section of the ECG trace between the end of the S wave and the beginning of the T wave. It stands for the period when the ventricles are depolarized and contracting but not yet repolarized. The ST segment is vital in diagnosing ischemia and myocardial infarction (Goldberger, 2024).

2.2. Arrhythmias

2.2.1. Left Bundle Branch Block

LBBB is a cardiac conduction disorder causing delayed or blocked electrical impulses in the left bundle branch, leading to altered left ventricular activation, contraction, and mechanics, and affecting patient diagnosis, treatment, and prognosis.

2.2.2. Right Bundle Branch Block

RBBB is a condition that affects the ventricular activation sequence, causing QRS to extend and modifying the orientation of R- and S-wave vectors in an ECG.

2.2.3. Atrial Fibrillation

AF is an arrhythmia caused by a variety of pathophysiological processes in the atria, which culminate in reduced atrial refractoriness and loss of atrial contractility. This syndrome is distinguished by rapid and irregular activity in various parts of the upper chambers of the heart, which contributes to high cardiac morbidity and mortality.

2.2.4. Ventricular Premature Contraction

PVCs are a kind of cardiac arrhythmias caused by ectopic heartbeats originating in the ventricles which are the heart's lower chambers, which can occur in healthy individuals or indicate serious conditions like structural heart disease, and cardiomyopathy.

2.2.5. Premature Atrial Contraction

Premature atrial contractions (PACs) are common irregular heartbeats originating in the atria, and upper chambers of the heart, associated with atrial fibrillation, stroke, and cardiovascular mortality, but are often considered benign.

2.3. ECG Arrhythmia Detection and Classification

2.3.1. Wavelet Transform

The wavelet transform is an effective mathematical tool used for signal and data analysis, compression, and reconstruction. This approach decomposes signals into different scales or frequency bands while also providing time and frequency information. Unlike the Fourier transform, wavelets can identify frequency components that change over time, giving them substantial advantages for dealing with non-stationary signals (Addison, 2017). Therefore, the wavelet transform is used for signal and image compression, reduction in noise levels, feature extraction, and time-frequency analysis. The wavelet transform is used to reduce the space occupied by data. It also reduces noise in signals like medical images or time series data. Wavelet transform can also be applied to detect significant signal features, such as EEG or ECG biomedical signal abnormalities. An essential requirement in identifying transient events is the simultaneous processing of time and frequency components. The potential applications of the wavelet transform are in signal processing to analyze biomedical signals, including the ECG and EEG. Image processing is used in medical and satellite pictures, recognizing faces, and image compression. Its applications in engineering include vibration analysis, signal assessment, and structural health monitoring. Critical wavelet transform properties, such as multiresolution analysis, allow researchers to study a signal's overall structure and details. Time-frequency localization allows for the identification of frequency components with time. Compactly supported wavelet functions obtain signal analysis, and orthogonal wavelets minimize the loss of information when the signals are decomposed into components. The various kinds of wavelet transforms are Continuous Wavelet Transforms (CWT), representing signals at continuous scales and shifts, thereby enabling a high-resolution time-frequency spectrum. Discrete Wavelet Transform (DWT) decomposes a signal at discrete scales and shifts and is less computationally demanding; therefore, it is more suitable for compression applications. Haar wavelets are relatively simple and fast for signal analysis and compression. Daubechies wavelets are more complex and smoother, hence providing better frequency resolution. Morlet wavelets are suitable for studying frequency, Meyer wavelets are smoothing wavelets with frequency domain definitions, and the wavelet transform is a comprehensive and powerful analysis technique that has proven useful in present signal and image processing. As a result, it was widely used in academic research and industrial applications.

2.3.1.1 Continuous Wavelet Transform

The continuous wavelet transform decomposes random processes into localized orthogonal basis functions for analysis, modeling, and simulation of non-stationary processes in a variety of applications, analyzing and reconstructing images, and signals. For example, an equivalent representation might be more understandable or shorter than the original data. Generally, a wavelet transform is performed on a function f() using a wavelet ψ(). The wavelets are defined as a waveform localized in time and space; more formally, one requires that it be an element of, i.e., ψ(t) satisfies only some necessary conditions. These functions are manipulated through shifting along the time axis and dilation (expanding or compressing the wavelet) to transform the signal into a different form, effectively 'unfolding' it in scale and time.

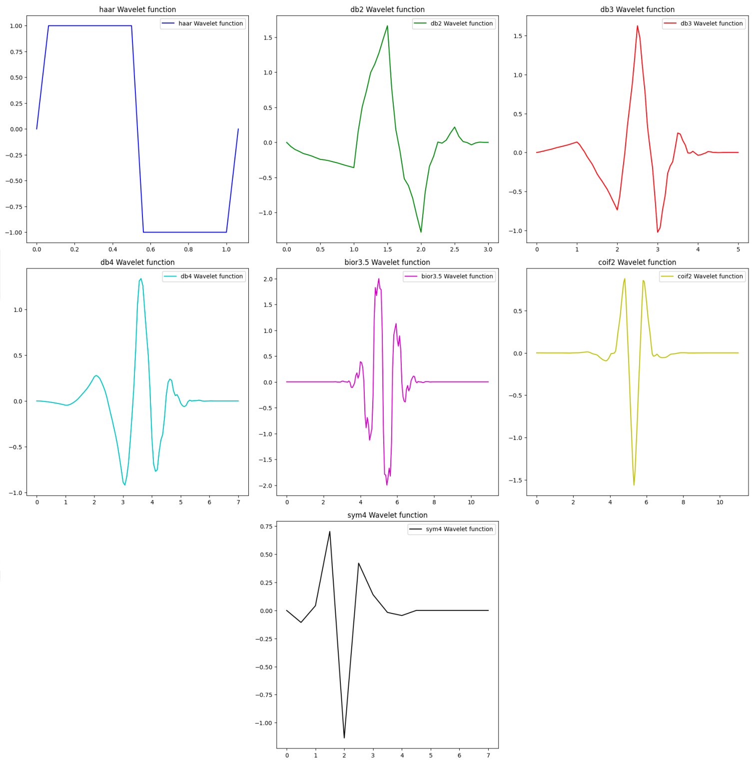


Figure 2.2. Wavelet functions, 4th level of resolution.

Source: Author

The 4th level of resolution plots of several wavelet functions used in the thesis are as in Figure 2.2. The definition of a continuous signal's wavelet transform for the wavelet function is,

$$T(a,b) = w(a) \int_{-\infty}^{\infty} x(t)\psi^{*}\left(\frac{1-b}{a}\right) dt \tag{1}$$

w(a) is chosen as $1/\sqrt{a}$ to ensure energy conservation across scales, with the asterisk notation implying the use of the wavelet function's complex conjugate during the transformation process.

From this point forward, we will use w(a) = 1/ $\sqrt{a}$ Therefore, the wavelet transform has the following expression:

$$T(a,b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{\infty} x(t)\, \psi^{*} \left(\frac{1-b}{a}\right) dt \tag{2}$$

This is called the continuous wavelet transform (CWT). Here, we integrate the product of the wavelet function and the signal over the entire signal range. In simple terms, this integration process is known as convolution. The normalized wavelet function is often written more simply as,

$$\psi_{a,b}(t) = \frac{1}{\sqrt{a}}\, \psi\left(\frac{1-b}{a}\right) \tag{3}$$

Therefore, the normalization ensures that the energy of the wavelet function is considered. As a result, the integral of the transform can be expressed as,

$$T(a,b) = \int_{-\infty}^{\infty} x(t)\ \psi_{a,b}^{*}(t)\, dt \tag{4}$$

An inverse wavelet transform is defined as follows:

$$x(t) = \frac{1}{C_g} \int_{-\infty}^{\infty} \int_{0}^{\infty} T(a,b)\ \psi_{a,b}(t)\, \frac{da\, db}{a^2} \tag{5}$$

Through integration over all scales and locations, b, this approach enables the reconstruction of the original signal from its wavelet transform. It's important to note that we use the original wavelet function in the inverse transform, not its conjugate as used in the forward transformation. If we restrict the integration to a specific variety of scales, we can perform a basic filtering operation on the original signal rather than considering all possible scales.

2.3.1.2 Discrete Wavelet Transform

It is feasible to restore the original signal entirely by utilizing finite sums of discrete wavelet coefficients instead of continuous integrals, which are necessary for the Continuous Wavelet Transform (CWT). Its ability to handle the Fast Wavelet Transform enables the ability to compute the discrete wavelet transform and its inverse rapidly (Addison, 2017).

The wavelet transform of a continuous-time signal, x(t), is examined using discrete values for the dilation (scale) and translation parameters, a and b. One approach to discretizing these parameters is using a logarithmic scale for a, which governs the scale, and linking this to discrete steps in b, which determine the translation locations. This wavelet discretization method follows a structured format where b moves in discrete steps proportionate to the scale a.

$$\psi_{m,n}(t) = \frac{1}{\sqrt{a_0^m}}\psi\left(\frac{t-nb_0a_0^m}{a_0^m}\right) \tag{6}$$

The parameters m and n determine the dilation and translation of the wavelet, respectively. a_0 is a predefined fixed parameter for dilation, set to a value greater than 1, while b_0 is a location parameter that must be greater than zero. The parameters m and n can take any integer value, including positive and negative ones.

Typical selections for the discrete wavelet parameters a_0 and b_0 are 2 and 1, correspondingly. This logarithmic scaling, where both dilation and translation steps follow powers of two, is referred to as the dyadic grid arrangement. The dyadic grid is regarded as the most basic and economical discretization approach for practical applications, allowing the generation of an orthonormal wavelet basis. By substituting a_0= 2 and b_0= 1 into equation (6), we observe that the dyadic grid wavelet can be expressed as,

$$\psi_{m,n}(t) = \frac{1}{\sqrt{2^m}}\psi\left(\frac{t-n2^m}{2^m}\right) \tag{7}$$

Employing the dyadic grid wavelet from equation (7), The discrete wavelet transform (DWT) has the following expression:

$$T_{m,n} = \int_{-\infty}^{\infty} x(t)\,\psi_{m,n}(t)\,dt \tag{8}$$

Selecting an orthonormal wavelet basis, denoted as $\psi_{m,n}(t)$, allows us to reconstruct the original signal in relation to the wavelet coefficients, $T_{m,n}$ using the inverse discrete wavelet transform in the following manner:

$$x(t) = T_{m,n} \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} T_{m,n}\ \psi_{m,n}(t) \tag{9}$$

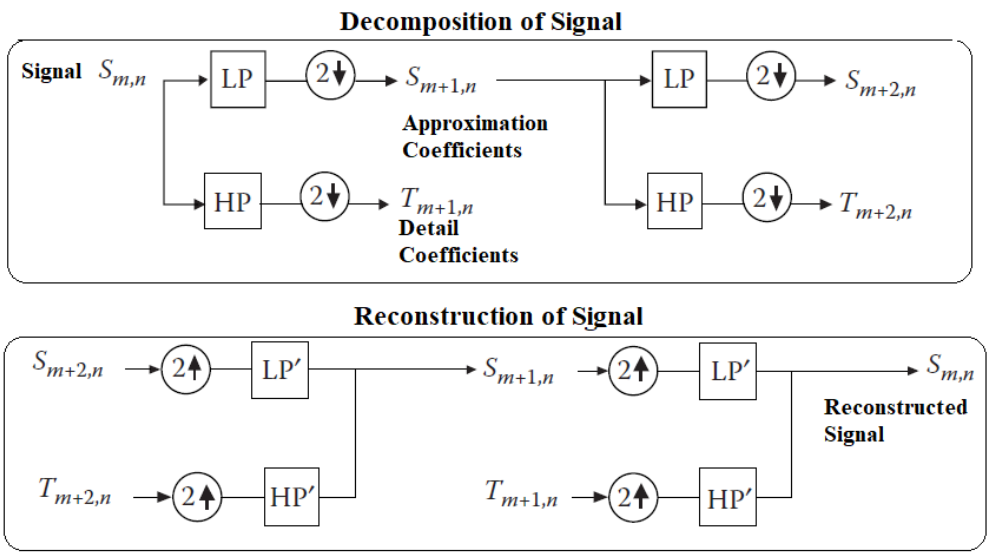
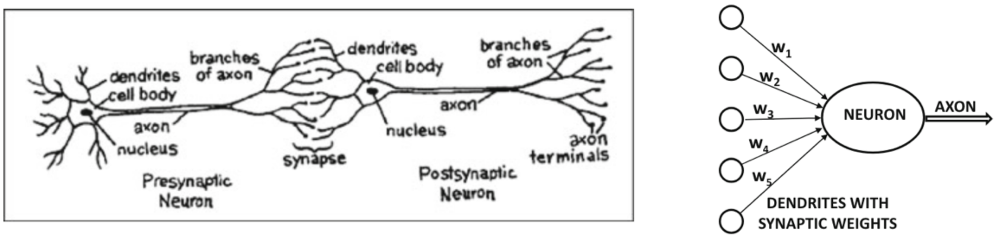


Figure 2.3. Two level decomposition and reconstruction from coefficients by low-pass and high-pass filters.

Source: Addison, 2017

2.3.2. Neural Networks

One of the most popular methods for mimicking how biological organisms learn is the use of artificial neural networks. An illustration of such a network in humans is the nervous system, where the main unit is the neuron that is connected and communicates with others using axons and dendrites at synapses, as shown in Figure 2.4(a). These synaptic connections change with experience and learning. To do this, artificial neural networks use computational units, termed neurons. In contrast to the biological neural network, these artificial counterparts depend on weights rather than synaptic strengths. Inputs to a neuron are influenced by these weights, which scale the neuron's calculation to produce an output. Figure 2.4 (b) An artificial neural network computes an output to give some inputs. It does this through the conduction of values from input to output neurons via weights acting as intermediary factors. Learning is achieved through changes to these weights connecting the neurons. (Aggarwal, 2018).



(a) Biological neural network (b) Artificial neural network

Figure 2.4. The link between neurons at the synapses.

Source: Aggarwal, 2018

2.3.2.1 The Perceptron

The perceptron is the most straightforward class of neural networks. A perceptron has a single layer for input and one output node. A simple diagram of a perceptron is shown in Figure 2.5. The single-layer perceptron is one of the building blocks for artificial neural networks and was developed by Frank Rosenblatt. Perceptron is a basic model of an artificial neural network used for solving linearly separable tasks, pattern recognition, and classification, with applications in machine learning and computational models.

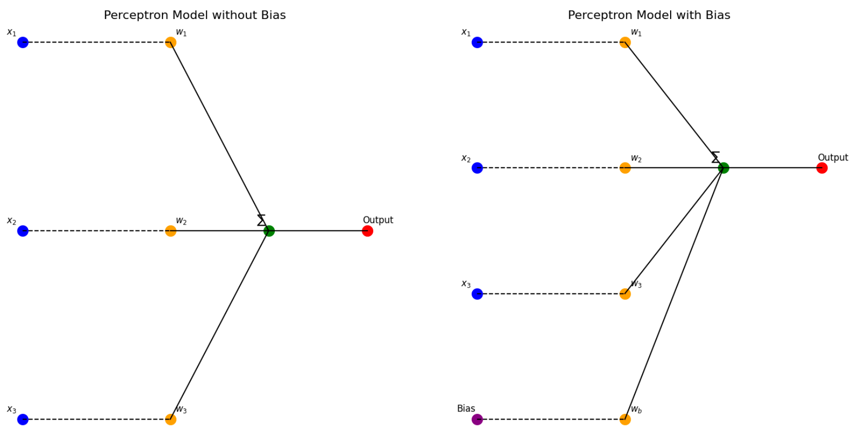


Figure 2.5. The basic architecture of the perceptron.

Source: Author

A single-layer perceptron operates with an input vector x = $[x_1, x_2, \ldots, x_n]$ and weight vector w = $[w_1, w_2, \ldots, w_n]$ The goal is to check whether the product of the input vector and the weight vector exceeds a threshold value, thereby producing an output. Steps as follows,

1. Dot Product of Input and Weight Vectors:

$$z = w \cdot x = w_1 x_1 + w_2 x_2 + \cdots + w_n x_n \tag{10}$$

2. Activation Function: The resulting z value is passed through an activation function, typically a step function for single layer perceptrons, θ is the threshold value.

$$y = \begin{cases} 1, & if\ z \geq \theta \\ 0, & if\ z < \theta \end{cases} \tag{11}$$

3. Weight Update: During learning, weights are updated based on the difference between the predicted output and the actual output. The weight update rule is as follows:

$$w_i = w_i + \Delta w_i \tag{12}$$

   Δw_i is the product of the learning rate (η) and the error (t−y), multiplied by the input value (x_i):

$$\Delta w_i = \eta(t - y)\, x_i \tag{13}$$

   Repeat steps of (Equations 10 - 13) until convergence or a predetermined number of iterations.

2.3.2.2 Convolutional neural networks

CNNs are a kind of deep neural network that has revolutionized the task areas in computer vision and image processing. In particular, CNNs are designed to process structured grid data, especially images, to solve classification, image segmentation, and object detection tasks. CNNs have found their inspiration through the biological processes of the visual cortex. Spatial hierarchies of features can be automatically and adaptively learned by a CNN architecture from input images. This property of hierarchical learning makes CNNs strong in managing the complexity and high dimensionality of visual data.

CNNs essentially involve convolutions, which are the actions of passing a filter or kernel through the input data such that feature maps are created. The process is similar to how the human brain processes visual information: it detects, in a hierarchical manner, various edges and textures, along with other more complex structures. Some main layers in a usual CNN architecture are the convolutional, pooling, and fully connected (dense) layers. Each of the layers has a specific role in the processing pipeline.

- **Convolutional Layer:** This is the basic module of a CNN. Convolutional layers apply an ensemble of learnable filters to the incoming data. When the filter moves across the input image, it does dot product operation and creates a feature map representing the presence of particular characteristics in various locations.

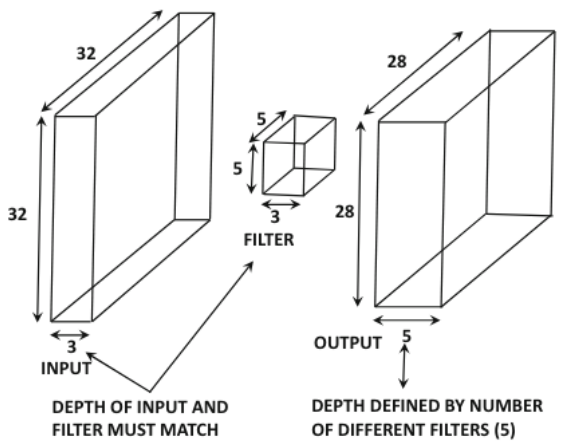


Figure 2.6. The Convolution between an input layer and filter.

Source: Aggarwal, 2018

- **Pooling Layer:** After the convolutional layer, pooling layers are applied to down-sample the spatial dimensions of the feature maps. This reduces the computational load and also provides spatial invariance that makes the network robust to the variations in the input. The operation of max pooling is shown in Figure 2.7. with 7 × 7 map strides of 1 and 2.

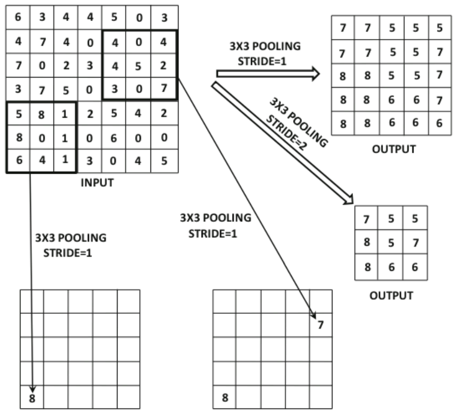


Figure 2.7. Max-pooling applied to a single activation map.

Source: Aggarwal, 2018

- **Fully Connected Layer:** After some convolutional and pooling layers, the final classification is done via fully connected layers. These flatten the previous layer into a one-dimensional tensor to feed it into standard feed-forward neural networks to classify the images.

Training a CNN is, in other words, finding out the most optimized weights for the filters in convolutional layers and connections in fully connected layers. Gradient descent is applied, most generally, through backpropagation. The process of training includes:

- **Forward Propagation:** The input is fed to the model. At each convolutional layer, feature maps are generated. After that, the pooling layers down-sample those characteristics, and the features obtained from it are passed to fully connected layers to give the final output, such as class scores.
- **Loss Calculation:** The loss function compares the network output with the actual ground truth labels, trying to quantify the difference or loss between actual and predicted labels.
- **Backward Propagation:** It is a neural network learning algorithm that adjusts weights to minimize error and is useful in a variety of fields such as image compression, pattern recognition, and music analysis. It can learn complex nonlinearity and adaptive control, but it has limitations such as slow training.
- **Parameter Update:** The network parameters are updated using an optimization algorithm, such as stochastic gradient descent (SGD), which

adjusts the weights in the direction that minimizes the loss (Goodfellow et al., 2016).

CNNs have been successfully applied to various applications, demonstrating their versatility and power in handling visual data.

Batch Normalization

Batch normalization is the newer technique devised to alleviate the vanishing and exploding gradients problems, in which gradients of activations in successive layers diminish or amplify in magnitude. Another major issue in training deep networks is internal covariate shift. Such a problem arises due to the variation of network parameters during the training process, leading to shifts in hidden variable activations (Aggarwal, 2018). In other words, the inputs from previous layers to later layers fluctuate back and forth, slowing down convergence when the later layers hit unstable data in the training. Batch normalization solves both problems by inserting "normalization layers" between hidden layers to change the features with more stable variance and stabilize training dynamics.

Activation Functions

Activation functions are applied in artificial neural networks to compute the output of neurons. Here, the activation functions transform the weighted sum of inputs to an output in a specified range, allowing the neural network to understand and identify complex patterns. Activation functions that are used alongside their formulas will be explained below.

- **Sigmoid Activation Function:**
  The output is restricted by the sigmoid function to a range of 0 to 1. The formula is as follows:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{14}$$

  The main advantage of this function is that it keeps outputs within a bounded range. However, it can slow down the learning process due to small derivatives for large or small inputs.

- **ReLU (Rectified Linear Unit) Activation Function**

The ReLU function leaves positive inputs unchanged and sets negative inputs to zero. The formula is as follows:

$$\mathrm{ReLU}(x) = max(0, x) \tag{15}$$

ReLU is computationally simple and fast. It also performs well with large datasets.

- **Leaky ReLU Activation Function**

Leaky ReLU is a variation of the ReLU function that provides a small slope for negative inputs. The formula is as follows:

$$\text{Leaky ReLU}(x) = \begin{cases} x & x > 0 \\ \alpha x & x \leq 0 \end{cases} \tag{16}$$

Here, α is typically a small value (e.g., 0.01) in Equation 16. This allows some activation for negative inputs, reducing the "dead neuron" problem.

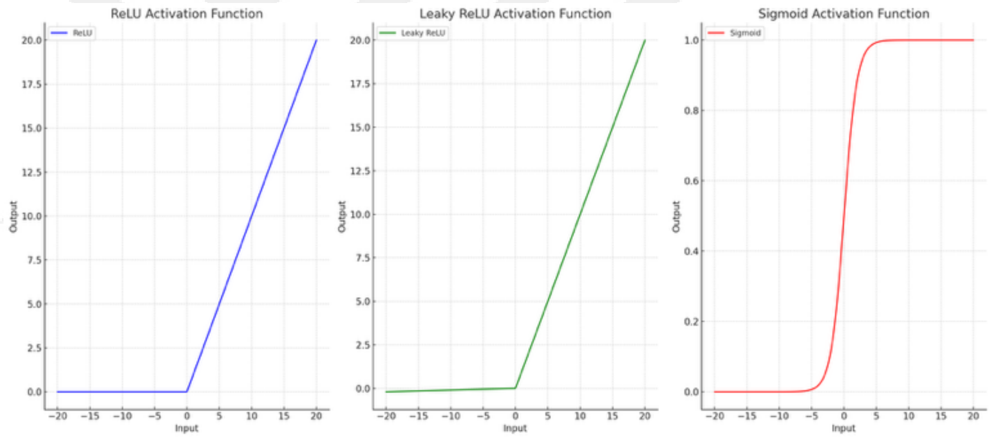


Figure 2.8. Activation functions.

Source: Author

Loss Functions

Loss functions define the error between target values in the train data and the prediction generated by a model. This needs to be minimized to increase the models' predictive power. This study measures the model's performance using mean absolute error (MAE).

Mean Absolute Error

The mean absolute error, commonly used in loss functions, is the average magnitude of errors without considering direction.

$$MAE = \frac{1}{n}\sum_{i=1}^{n} | y_i - \hat{y_i}| \tag{17}$$

Where in Equation 17:

- n is the number of data points.
- y_i is the actual value for the i-th data point.
- $\hat{y_i}$ is the predicted value for the i-th data point.

2.3.3. Autoencoder

Autoencoders are artificial neural networks used to learn effective codings in an unsupervised manner. Learning a representation for a set of data is the fundamental concept of an autoencoder, usually by trying to find good ways to compress and decompress the data. Besides its basic compression and decompression functionality, an autoencoder can learn a highly vital feature of the data, which can be used to reconstruct the input from this compressed representation. This study describes the architecture, types, applications, and challenges of autoencoders as part of machine learning.

The encoder and the decoder are the two primary components of autoencoders. The input is compressed by the encoder into a latent-space representation, which the decoder then uses to recreate the original input. Mathematically, if x is the input data, the encoder function f maps x to z, the latent representation, and the decoder function g maps z back to x', the reconstructed input. The goal is to minimize the reconstruction error, which can be expressed as:

$$L(x, x') = ||x - x'||^2 \tag{18}$$

where L is the loss function, typically the mean squared error or mean absolute error.

The encoder part of AE transforms the input data x into a latent representation z. It reduces the dimensionality of the data, capturing the most salient features. The encoder can be represented as:

$$z = f(x) = \sigma(Wx + b) \tag{19}$$

Where W and b are the weight matrix and bias vector of the encoder, respectively, and σ is the activation function in Equation 19.

The decoder part of AE reconstructs the original data from the latent representation. It maps the latent space back to the input space and can be represented as:

$$x' = g(z) = \sigma(W'z + b')\ x' \tag{20}$$

Where W' and b' are the decoder's weight matrix and bias vector, respectively, and σ is the activation function.

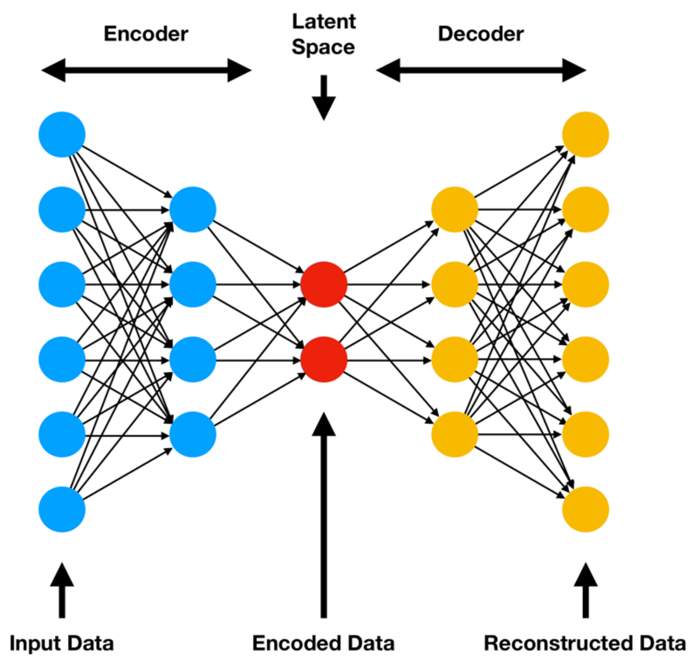


Figure 2.9. General architecture of an autoencoder.

Source: CompThree, n.d.

The architectures and the type of constraints applied to the autoencoders put them into different categories. Some of the most commonly used are the undercomplete, sparse, denoising, variational, and contractive autoencoders. An under-complete autoencoder is one in which the dimensionality of the latent space is lower than that of the input space, hence forcing the network to learn essential features of the data—in essence, with its primary objective as that of a dimensionality reduction method. The hidden representation of sparse autoencoders is then enforced with a sparsity constraint, meaning that at any time, only a few neurons will be active, rendering their features useful for tasks like image recognition and anomaly detection. Denoising autoencoders (DAEs) are specialized autoencoder neural networks trained to reconstruct the original input from a corrupted or noisy version. This training process helps in learning robust representations that can effectively handle noise and corruption in the data (Vincent, 2011). A probabilistic approach for auto-encoders introduces variational autoencoders, which map inputs to a distribution in place of a single point in the latent space and apply a probabilistic encoder and decoder; this term includes a loss that enforces learned distribution to be near a prior distribution. Lastly, contractive autoencoders add a penalty term to the loss function that penalizes large derivatives of the encoder function, ensuring that the encoder is less sensitive to minor variations in the input, which leads to more robust feature learning.

Due to their capacity to learn compact and meaningful data representations, autoencoders can be applied to various domains. Autoencoders can be considered a dimensionality reduction technique, much like Principal Component Analysis (PCA), which reduces the features in data while maintaining vital information to be useful for visualization and data processing. Additionally, autoencoders can serve as anomaly detectors since they learn normal patterns from the data; this is because anomalies tend to give rise to high reconstruction errors, meaning that unusual patterns within the dataset can be found. Furthermore, denoising autoencoders can be instrumental in image enhancement and preprocessing by learning to reconstruct noisy images into their original versions. Moreover, variational autoencoders allow for the generation of new samples via sampling through learned latent space, thus facilitating applications like image generation, data augmentation, and semi-supervised learning. Lastly, these learned features often yield better performance than raw data because an Autoencoder can extract valuable features from the data applicable to other tasks, such as classification or clustering.

Autoencoders are powerful, unsupervised learning tools capable of learning compact and meaningful data representations. They have many uses, from dimensionality reduction and anomaly detection to image denoising and data generation. However, they also come with challenges such as overfitting, interpretability, and computational complexity. Despite these challenges, ongoing research continues to improve the robustness and efficiency of autoencoders, making them an essential component of modern machine-learning techniques.

3. CHAPTER: EXPERIMENTS AND RESULTS

The convolutional autoencoder model designed custom and used in the following studies will be mentioned in this section. Over the past few years, deep learning and signal processing have joined forces to create new ways to analyze data and extract features. A Wavelet-Based Convolutional Autoencoder (WBCAE) model is proposed in this thesis for arrhythmia detection and classification. Unlike Fourier transforms, which give frequency info but lose track of time, wavelets keep both time and frequency details. This makes them great for looking at signals that change over time. The WBCAE combines wavelet transforms and convolutional neural networks (CNNs) into one model. Therefore, this model uses the advantages of both methods to improve feature extraction and data representation. This section looks at how Wavelet-based CAEs are built, how they work, and the results of the experiments.

A WBCAE typically consists of the following components:

- **Wavelet Transform**: The wavelet transform divides incoming data into frequency components. This method generates wavelet coefficients, which display the data at various scales and resolutions. The wavelet transform is useful for analyzing signals with temporary characteristics that change properties. It permits multi-level analysis of signals ranging from wide to detailed, which is crucial for detecting localized signal properties that vary over time (Li et al. 2023).
- **Convolutional Encoder**: A convolutional encoder receives wavelet coefficients. This encoder contains several convolutional layers that apply filters to the wavelet coefficients. These layers capture geographical hierarchies and reduce the data size. The encoder's primary function is to compress the data into a smaller latent space while retaining important properties.
- **Convolutional Decoder**: The squeezed version from the encoder is passed through a convolutional decoder, which attempts to recreate wavelet coefficients. The decoder utilizes deconvolutional layers (also known as transposed convolutional layers), which enhance the latent representation to match the original size of the wavelet coefficients.

- **Inverse Wavelet Transform**: To return the reconstructed wavelet coefficients to the original data domain, an inverse wavelet transform is applied. This phase guarantees that the output data looks identical to the original data, reducing reconstruction errors.

3.1. Determination of Atrial Fibrillation with WBCAE Anomaly Detection

Remote patient monitoring is crucial for early disease detection and improving quality of life. Advances in deep learning have propelled wearable health technologies forward, enhancing automatic diagnosis capabilities. The deep learning approach for classifying atrial fibrillation (AF) arrhythmia is presented in this study using a customized wavelet-based convolutional autoencoder (WBCAE). The WBCAE combines the time-frequency analysis of wavelets with the feature-learning ability of convolutional autoencoders, serving as an anomaly detector.

Atrial Fibrillation (AF) is the most studied heart rhythm disorder, characterized by irregular and rapid atrial rhythm (300-500 beats per minute). Unlike the regular impulses in Normal Sinus Rhythm (NSR), AF results from abnormalities in impulse generation or cellular connections, leading to chaotic impulses. Despite advancements, accurately classifying AF remains challenging, complicating treatment plans and prognosis.

This study aims to develop a Wavelet-based Convolutional Autoencoder (WBCAE) for efficient AF detection. The study's contributions include:

1. Enhancing AF detector performance using convolution filters on a single ECG signal channel.
2. Combining wavelet signal analysis with deep learning via the WBCAE structure.
3. Addressing data imbalance by training the network with a single signal type using anomaly detection.

Anomaly detection entails identifying uncommon or atypical data points that significantly deviate from the majority of the data. Autoencoders are particularly adept at this task because they are trained to reconstruct normal patterns. During the testing phase, when encountering anomalous data, the reconstruction error tends to be higher, making it feasible to identify these anomalies.

Steps in Anomaly Detection Using Autoencoders

1. **Training on Normal Data**: The autoencoder is trained using normal sinus rhythm data, allowing it to learn the patterns and features of this data.

2. **Reconstruction Error**: During the testing phase normal and anomaly data are used and the autoencoder attempts to reconstruct new data. For normal data, the reconstruction error will be low. For anomalous data, the error will be high.
3. **Threshold Setting**: A threshold for the reconstruction error is set based on the distribution of errors for the normal data. Data points with errors above this threshold are classified as anomalies and below are classified as normal.

This model emphasizes identifying abnormalities over simply categorizing rhythms, offering a novel approach to AF detection. Testing on diverse datasets demonstrates its flexibility and reliability, distinguishing it from other studies.

3.1.1. Dataset

When discovered too late, atrial fibrillation can result in mortality, an embolism, or even a stroke. The most used method for timely detection of this severe condition is the examination of ECG records. In ECG recordings, three specific signs of atrial fibrillation are mainly considered: the absence of the P wave, irregular RR intervals, and fibrillation on the baseline.

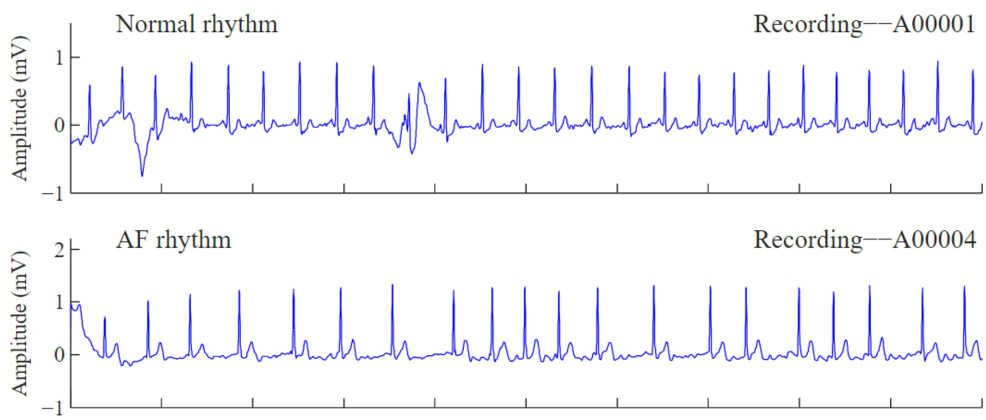


Figure 3.1. Normal Sinus Rhythm and Atrial Fibrillation ECG recordings.

Source: Clifford et al., 2017

Figure 3.1 illustrates ECG recordings of normal sinus rhythm and atrial fibrillation. As can be observed from the upper graph, P waves, QRS complexes, and T waves can be easily identified for each beat. The distance between R peaks is regular. However, the second graph observes beats in irregular time instants. Furthermore, P peaks are absent, and a quivering isoelectric line is shown at the TQ interval. Autoencoder studies were carried out using publicly available ECG databases. NSR data "MIT-BIH Sinus Rhythm Database" (NSRDB) (Goldberger et al., 2000) and Atrial

Fibrillation data “MIT-BIH Atrial Fibrillation Database (AFDB)” (Moody, 1983), “The PhysioNet/Computing in Cardiology Challenge 2017” (AFPC) (Clifford et al., 2017) taken from databases. Table 3.1 is a list of all the database's features.

Table 3.1. shows that NSRDB includes 24-hour data from 18 healthy individuals. In contrast, AFDB includes 10-hour AF and non-AF data from 25 patients and short-term single-channel ECG recordings used in the competition held by Physionet in 2017. Only the AF portion of the Physionet competition data was included in this study. The locations and beat labels of the QRS complexes of ECG signals in the NSRDB and AFDB databases are available.

Table 3.1. Description of the databases.

Data	***Subject***	***Lead***	***Duration of recordings***	***Sampling frequency***
NSRDB	18	2	24 Hours	128 Hz
AFDB	25	2	10 Hours	250 Hz
AFPC	771	1	10-60 seconds	300 Hz

AF signals in the AFPC database were separated with the Pan Tompkins algorithm and labeled by expert authors of this study. The sampling frequency was converted to 250 Hz for data at different sampling frequencies. Before the data was applied to the autoencoder, the signal was divided into 256 sample windows, which can be seen in Figure 3.2.

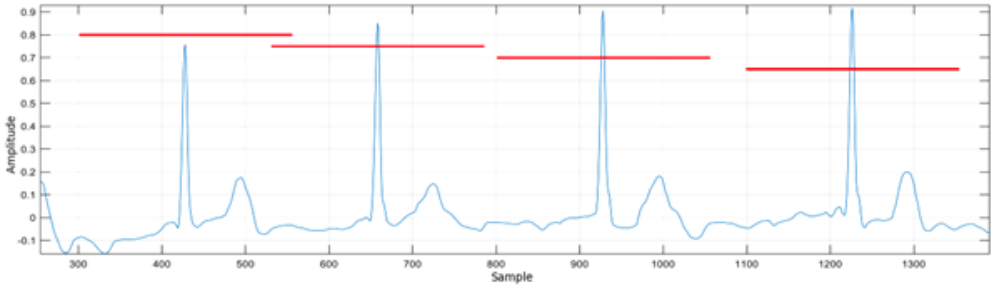


Figure 3.2. Illustration of ECG Signal Windowing.

Source: Author

In Figure 3.2., each red line shows the interval of a signal window and Table 3.2. lists the number of data windows resulting from the process.

Table 3.2. Number of Data Windows Used in Training and Testing

Data	***NSRDB***	***AFDB***	***AFPC***
Number of frames used for training	800,000	-	-
Number of frames used in testing (Test 1)	395,455	395,455	-
Number of frames used in testing (Test 2)	32,010	-	32,010

3.1.2. Evaluation methods

This study uses an autoencoder for anomaly detection to recognize AF and NSR ECG signals. In anomaly detection, AE is trained with only a single class data, so the model is optimized to represent this data. In this study, the autoencoder was trained with NSR data to minimize the reconstruction error, as depicted in Figure 3.3. In the testing phase, NSR and AF signals are applied to the AE, and the reconstruction error is calculated, as seen in Figure 3.3. This approach allowed for the evaluation of how well the autoencoder could reconstruct both NSR and AF signals, providing insights into its performance in distinguishing between the two rhythm types. If the error is less than the given threshold value, it is labeled as NSR; if it is greater than the given threshold value, it is labeled AF.

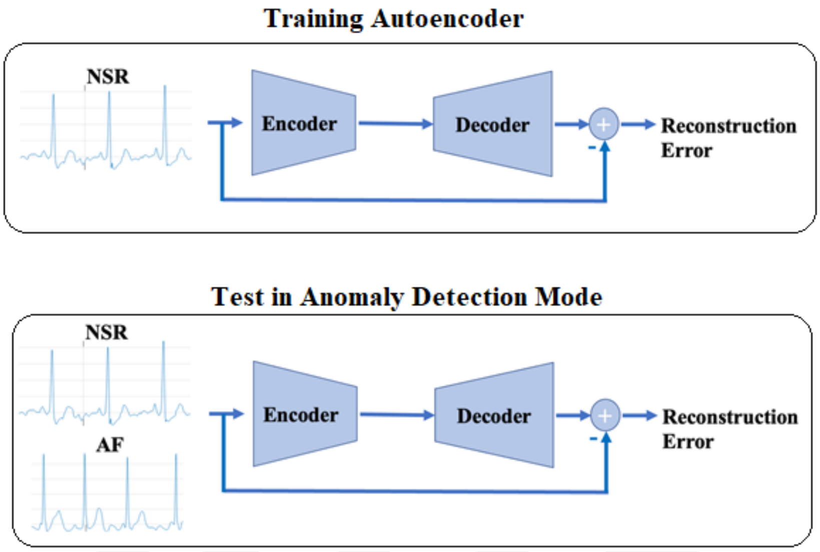


Figure 3.3. Train and test of AE with NSR and AF rhythms in anomaly detection.

Source: Author

The critical issue in anomaly detection is to select the threshold. In this study, the following steps are applied to obtain an acceptable threshold value that leads to successful detection:

1. Use the model to calculate reconstruction loss on normal data.
2. Use the model to calculate reconstruction loss for anomalous data.
3. Generate a range of threshold values between the minimum and maximum reconstruction loss values observed in the normal data.
    - Iterate over different threshold values to find the best F1 score. Compute the precision, recall, and F1 scores for each threshold value.
    - Update the best F1 score and corresponding threshold if the current F1 score exceeds the previous one.
4. Return the best threshold and corresponding F1 score as the optimal threshold.

The identified optimal threshold is applied to the mixed test data, consisting of normal and atrial fibrillation samples. The performance of the chosen threshold is assessed using several metrics, including F1 score, precision, and recall, to assess the effectiveness of the anomaly detection system. After detecting normal and anomaly with threshold, the results of the models were examined with the evaluation metric.

3.1.2.1 Evaluation Metrics

Accuracy

The percentage of correctly identified cases out of all the instances is what accuracy measures. It is a fundamental metric for evaluating the overall performance of the classification model.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{21}$$

Where in Equation 21:

TP = True Positives

TN = True Negatives

FP = False Positives

FN = False Negatives

Precision

The number of true positive predictions made out of all positive predictions is what precision quantifies. It shows how many predicted positives are positive.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{22}$$

Recall

The number of true positive predictions created out of all actual positive cases is known as recall or sensitivity. It indicates how many of the actual positive instances the model correctly identified.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{23}$$

F1-Score

The F1-score combines both precision and recall into one measure by calculating their harmonic mean. This metric offers a balanced view of the model's performance, considering both how often it correctly identifies positive cases and how accurately it excludes negative ones.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{24}$$

These metrics collectively give a detailed insight into the model's performance, highlighting its strengths and areas needing improvement.

3.1.3. Wavelet Based Convolutional Autoencoder Design

In this study, An EncoderMiniBlock and DecoderMiniBlock, which are Wavelet-based Convolutional AutoEncoder (CAE) parts, are custom designed. When considering the feature space, the likelihood of overfitting increases with the complexity of the model during training. This study preferred a simple architecture to avoid overfitting and simultaneously reduce computational complexity. The proposed model is given in Figure 3.4.

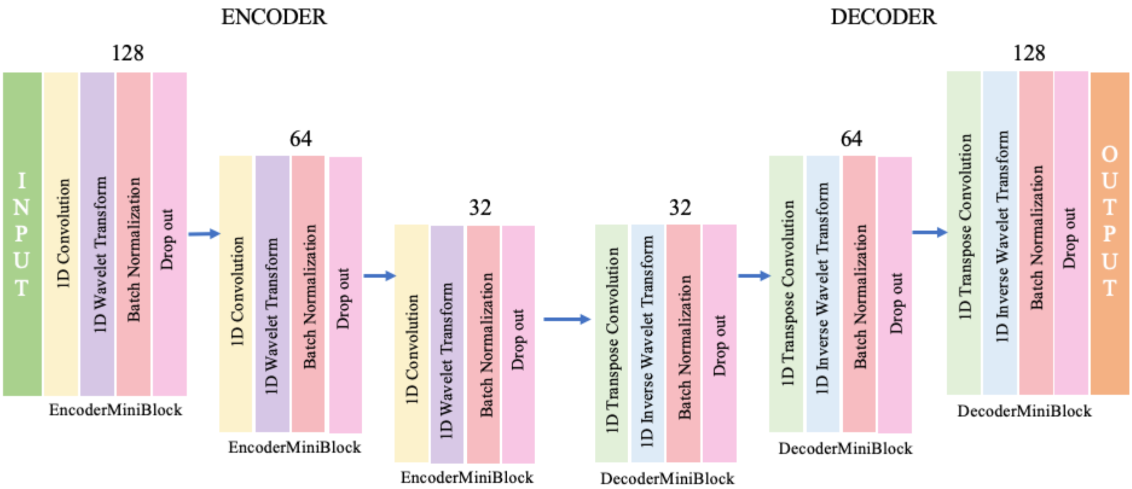


Figure 3.4. Proposed Wavelet-Based Convolutional Autoencoder Model.

Source: Author

The optimal model was discovered by experimenting with different models, changing architectures, layer counts, and other configurations. We found that the proposed model performed the best after trying various options. Figure 3.4 shows three EncoderMiniBlocks containing 128, 64, and 32-dimensional filters used in the encoder. Similarly, 32, 64, and 128-dimensional decoding MiniBlocks are included in the decoding section. The last layer contains a single-unit fully connected layer (dense layer) and a Rectified Linear Unit (ReLu). Within the EncoderMiniBlock are convolutional layers or 1D convolution layer, discrete wavelet transform (DWT) layer, batch normalization layer, and dropout layer, respectively. WaveTF library was used for wavelet function implementation (Versaci, 2020).

WaveTF is a TensorFlow library that implements 1D and 2D wavelets, transforms them, and exposes them as Keras layers to easily add them to machine learning workflows. The library implements the most used Haar and DB2 wavelet kernels. Anti-symmetric reflection filling is applied to handle boundary effects, which broadens the signal while preserving its first-order finite difference at the boundary. WaveTF transparently supports both 32- and 64-bit floating point at runtime.

Table 3.3. Reconstruction of low pass filter coefficients of the wavelet functions

	Db2	**Db3**	**Db4**	**Sym4**	**Coif2**	**Bior 3.5**
$g_0[0]$	0.48296	0.03223	-0.230378	0.032223	0.016387	0.0
$g_0[1]$	0.836516	0.08544	0.714847	-0.012604	-0.041465	0.0
$g_0[2]$	0.2241439	-0.13501	0.630881	-0.099219	-0.067373	0.0
$g_0[3]$	-0.1294095	-0.45988	-0.027984	0.2978578	0.3861101	0.0
$g_0[4]$		0.80689	0.187035	0.8037388	0.8127236	0.1767767
$g_0[5]$		-0.33267	0.0308414	0.4976187	0.417005	0.5303301
$g_0[6]$			-0.0328830	-0.029636	-0.0764886	0.5303301
$g_0[7]$			-0.010597	-0.075766	-0.0594344	0.1767767
$g_0[8]$					0.02368017	0.0
$g_0[9]$					0.00561144	0.0

$g_0[10]$	-0.0018232	0.0
$g_0[11]$	-0.0007206	0.0

If wavelet transformation is active in the EncoderMiniBlock, the transformation function is defined for the selected wavelet. In the original version of the library, only Haar and Daubechies 2 wavelets are defined. In this study, wavelets successfully used in literature for arrhythmia detection were also adapted to the library. The DecoderMiniBlock contains a 1D transpose convolution layer, Inverse Wavelet Transform (IDWT) layer, batch normalization layer, and dropout layer. This study conducted autoencoder experiments with Haar and DB2 wavelets and wavelets that generally give successful results in biomedical signal classification. To implement wavelet transform in WaveTF library, the low-pass reconstruction filter coefficients should be entered as model parameters. The wavelet coefficients used are listed in Table 3.3.

3.1.4. Experiments and Results

This study aims to train the wavelet-based convolutional autoencoder with a single class of data, optimize it according to this signal, and obtain an efficient system separating the signal type from others in the testing phase. Model in Figure 3.4. was trained with NSR data from the NSRDB database. At the end of the training, the tests were performed with data from the NSRDB database, which the model did not see in training, and data taken from two different databases, AFDB and AFPC. Experiments were conducted in the TensorFlow 2 environment in Python 3. If there is no improvement in the validation error for ten epochs, early stopping is applied to prevent overfitting. Adagrad optimization algorithm was used with 128-dimensional batches. The initial learning rate was chosen as 10-3. The train is set to continue for a maximum of 50 epochs. In Test 1, 395,455 entries from the NSRDB database and 395,455 from the AFDB database were used. In Test 2, 32,010 entries from the NSRDB database and 32,010 from the AFPC database were used. The model was trained with 800,000 NSR entries from the NSRDB database for both tests. The data is divided into separate sets for train and test purposes. During training, the model learns to reconstruct the input data without exposure to the patient's data in the test set. This ensures that the test set consists of unseen examples, allowing for a rigorous evaluation of the model's generalization performance. Therefore, when training an autoencoder, the test data remains entirely independent, ensuring an unbiased assessment of the model's ability to reconstruct unseen instances.

3.1.4.1 Experiment 1: Effect of wavelet family on performance

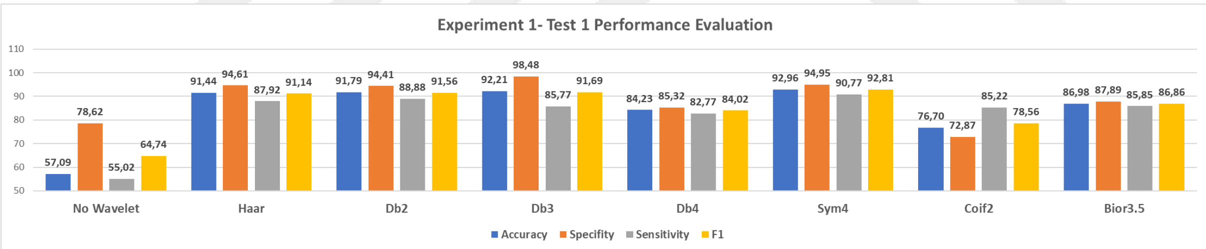
This experiment assesses how different wavelet families impact the performance of convolutional autoencoder models in anomaly detection tasks using ECG signals. By training multiple models with various wavelet families (e.g., Daubechies, Symlet, and Coiflet), the study aims to identify the optimal wavelet family that enhances the model's ability to extract relevant features and accurately detect anomalies. The WBCAE Model (Figure 3.4.) structure was established without a wavelet layer and also with the various wavelets. The system was optimized, and the loss function MAE, which gave the best results, was selected. The results of the experiments are listed in Table 3.4 and Figure 3.5. (a) and (b) separate performance graphs for both experiments are given according to wavelet type.

Table 3.4. Experiment 1 Results: Analysis and Findings

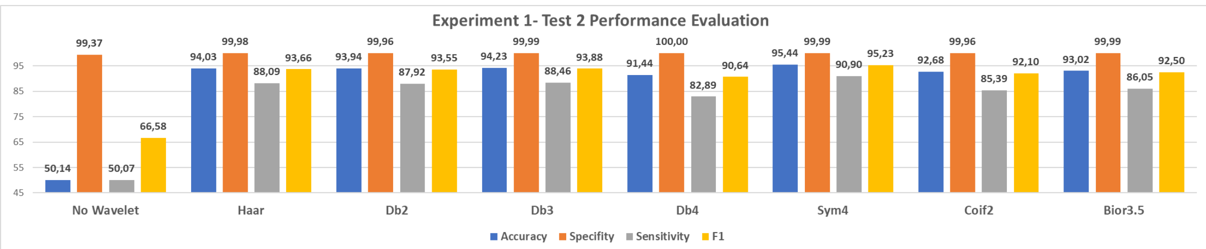
Wavelet		*Accuracy (%)*	*Precision (%)*	*Recall (%)*	*F1 Score (%)*
***No* Wavelet**	**Test 1**	57.09	78.62	55.02	64.74
	Test 2	50.14	99.37	50.07	66.58
Haar	**Test 1**	91.44	94.61	87.92	91.14
	Test 2	94.03	99.98	88.09	93.66
Db2	**Test 1**	91.79	94.41	88.88	91.56
	Test 2	93.94	99.96	87.92	93.55
Db3	**Test 1**	92.21	98.48	85.77	91.69
	Test 2	94.23	99.99	88.46	93.88
Db4	**Test 1**	84.23	85.32	82.77	84.02
	Test 2	91.44	100.00	82.89	90.64
Sym4	**Test 1**	92.96	94.95	90.77	92.81

	Test 2	95.44	99.99	90.90	95.23
Coif2	**Test 1**	76.70	72.87	85.22	78.56
	Test 2	92.68	99.96	85.39	92.10
Bior3.5	**Test 1**	86.98	87.89	85.85	86.86
	Test 2	93.02	99.99	86.05	92.50

When Table 3.4. and Figure 3.5. are examined, it is observed that the addition of a wavelet layer improves the classification performance noticeably. Among all wavelet families, Symlet 4 produced the best accuracy, and all the scores are balanced for this wavelet. In Test 2, all wavelets achieved better results compared to Test 1. The downloaded site provided the labels of the AFDB database used in Test 1. However, upon visual inspection by the experts, it was determined that the labeling was done in blocks, and some AF beats had more normal sinus rhythm characteristics than AF. Our cardiologist authors relabeled all the beats in the AFPC dataset, and all the beats used in Test 2 were correctly identified. This may explain the difference between the classification performance. Symlet 4 wavelet is evenly ahead in all performance scores for both sets.



(a)



(b)

Figure 3.5. Performance Comparison of WBCAE with Various Wavelets.

Source: Author

3.1.4.2 Experiment 2: Effect of input window size on performance

The performance of anomaly detection models trained on ECG data is the main focus of this experiment, which investigates the impact of changes in input window size. By varying the window size and evaluating model performance metrics, the experiment aims to determine the optimal window size for effectively capturing temporal dependencies and detecting anomalies in ECG data. The Sym4 wavelet and AFPC database were used in the tests. The results are given in Table 3.5. and Figure 3.6. The highest success was achieved for length 256.

Table 3.5. The effect of different window size on the performance metrics

Window Size	*Accuracy (%)*	*Precision (%)*	*Recall (%)*	*F1 Score (%)*
256	**95.44**	**99.99**	90.9	**95.23**
512	92.35	91.44	**93.13**	92.28
1024	90.36	96.36	86.03	90.90

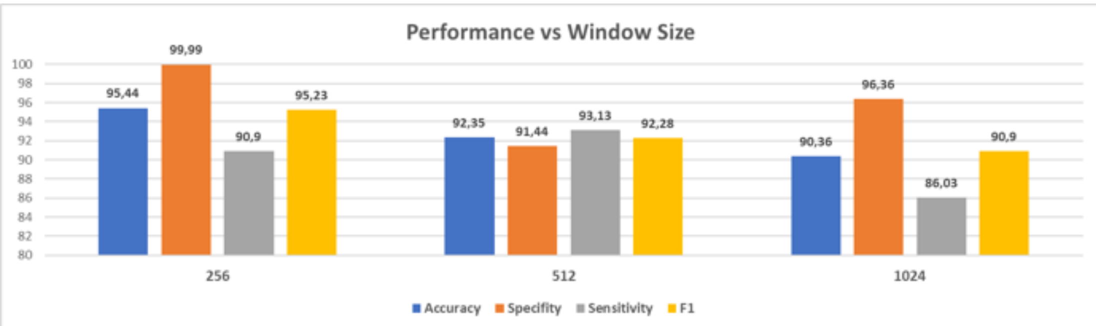


Figure 3.6. Impact of Window Size on Performance Metrics.

Source: Author

3.1.5. Summary

In experiment 1, different wavelet families are tested. We observed that a notable enhancement in the detection performance was obtained with the addition of a wavelet layer. Furthermore, it is observed that the Symlet 4 wavelet produces the best results.

The results validate our intuition that the wave closely resembling the analyzed normal sinus rhythm waveform will be deemed successful. During the model design phase, the impact of altering the structures by varying the number and positions of the layers was conducted. Our findings revealed that the proposed model, as depicted in Figure 3.4., outperforms the other models tested. Thus, other models were not included in the study. In another experiment, the input window size of the system is changed for the proposed model with Sym4 wavelet. The window size 256 is observed to perform better in accuracy, precision, and F1 scores. The AF detection performance of the state-of-the-art machine learning models in the literature is also considered.

In this study, we developed a robust autoencoder structure based on wavelets, which proved highly effective even for a short window of approximately 1 second. We conducted various tests using essential wavelets and analyzed key performance metrics such as accuracy, recall, precision, and f1 score. These evaluations helped us compare the effectiveness of different wavelets. We also examined factors like input length and loss function across various models. The Sym4 wavelet emerged as the most promising and successful among the tested methodologies. The wavelet layer is shown to enhance the performance of the AE structure in anomaly detection mode. Thus, the proposed model can be employed in different signal-processing applications, even for unbalanced datasets. The selection of wavelets plays an essential role in the network performance.

3.2. Arrhythmia Classification with WBCAE as Feature Extractor

Autoencoders have a role, in extracting features and turning data into analyzable elements for tasks like classification and clustering. In the autoencoder setup, the encoder compresses input data into a space known as the space or bottleneck layer. This condensed representation captures features while filtering out noise and details. The compressed form can then serve as a feature vector, for machine learning tasks.

Steps in Feature Extraction Using Autoencoders

1. **Data Preprocessing**: Before feeding data into the autoencoder it is commonly prepared. This involves steps like scaling data and converting ECG signals into heartbeats.

2. **Training the Autoencoder**: The autoencoder undergoes training with the preprocessed data. It learns how to convert the incoming data into a latent space and then reconstruct it during this process.
3. **Extracting Features**: Once trained the encoder component of the autoencoder can be utilized to convert input data into the space effectively extracting features from the data.
4. **Classification Using the Encoded Features**: These features that have been extracted can serve as inputs for machine learning models, such, as classifiers or clustering algorithms.

Autoencoders help with feature extraction. Offer benefits making them valuable, in numerous situations. Unlike PCA and other conventional approaches that focus on data analysis autoencoders leverage neural network structures to detect nonlinear connections, within their datasets. This makes it possible to reduce dimensions without deleting essential aspects of the data for simplicity in processing and analyzing. For instance, the compression process may involve filters that make the features less noisy. Likewise, they are highly adaptive; hence, their versatility in feature extraction procedures is demonstrated by the fact that they may be used with a variety of data kinds, including text, time series, and pictures.

This study proposes an efficient wavelet-based convolutional autoencoder model for the feature extraction of the arrhythmia types, such as right bundle branch block (RBBB), left bundle branch block (LBBB), premature ventricular contractions (PVC), atrial premature contractions (APC) as shown in Figure 3.8. One-lead ECG signals are then classified with a Multilayer Perceptron (MLP), as illustrated in Figure 3.7. The combination of the wavelets with autoencoder structure is one of the main contributions of this study. Therefore, the success of wavelets in grasping the time-frequency domain distribution of the signals was integrated into the learning capability of autoencoders. Also, a large dataset collected from different databases was used for training and testing to show the performance under various conditions (Eravcı & Ozkurt, 2023).

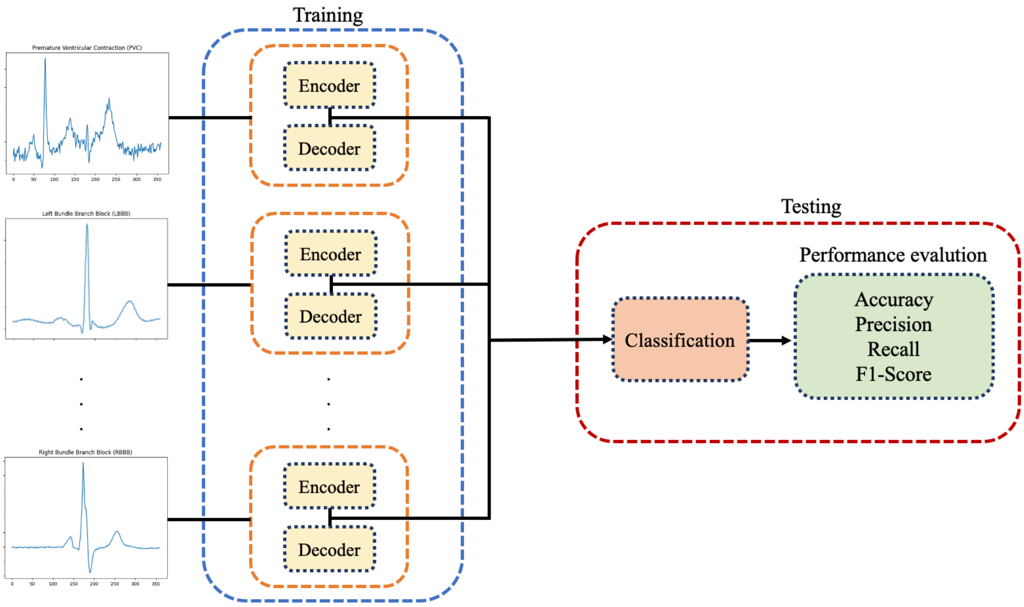


Figure 3.7. Diagram of proposed method.

Source: Author

3.2.1. Dataset

Three databases, including the St. Petersburg INCART 12-Lead Arrhythmia Database, the MIT-BIH Arrhythmia Database, and the MIT-BIH Supraventricular Arrhythmia Database, were integrated as indicated in Table 3.6. We preserved 180 samples before and after each R-peak, obtained 360 sampling points for each beat, and then resampled all data to 180 Hz. The ECG signal was min-max normalized after each beat was divided into beat segments, as shown in Figure 3.8.

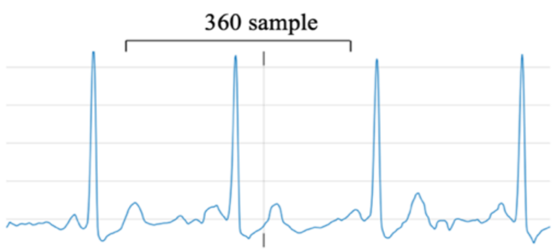


Figure 3.8. Each heartbeat 360 Sampled RR-Interval lengths.

Source: Author

The MIT-BIH Arrhythmia Database is a commonly used test data for evaluating arrhythmia detectors (Moody & Mark, 2001). It's mainly composed of 48 two-channel ambulatory ECG recordings sampled at 360Hz. Each recording contains 48 half-hour ECG excerpts. The data was gathered from 47 people. The ages of the participants, who comprised 25 men and 22 women, ranged from 32 to 89. This dataset consists of recordings with various degrees of arrhythmias and recordings with normal sinus rhythm. The MIT-BIH Supraventricular Arrhythmia Database is a publicly available database commonly used in cardiac arrhythmia research and comprises 78 half-hour-long excerpts of two channel ECG. It includes a selection of ECG recordings with annotations concentrated on supraventricular arrhythmias (Chen et al., 2017). St. Petersburg Institute of Cardiological Technics 12-lead Arrhythmia Database has 75 annotated recordings taken from 32 Holter records, which are included in this database. Each record has 12 standard leads sampled at 257 Hz and lasts 30 minutes. The recordings were taken from patients with various arrhythmias, including sinus rhythm and ventricular and supraventricular arrhythmias (Tihonenko et al., 2008). In this study, we concentrated on four types of arrhythmia and normal sinus rhythm listed in Table 3.6.

Table 3.6. Data Distribution of Arrhythmia Types Used

Arrhythmia Types	**Beat Size**
Left Bundle Branch Block (LBBB)	13,322
Right Bundle Branch Block (RBBB)	12,068
Normal Sinus Rhythm (NSR)	75,011
Premature Ventricular Contraction (PVC)	33,632
Atrial Premature Contraction (APC)	4,441
Total	138,474

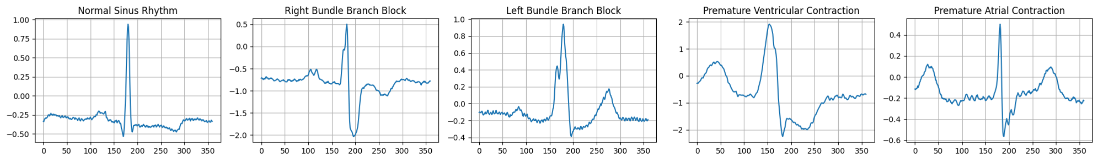


Figure 3.9. Five types of arrhythmia heartbeats.

Source: Author

3.2.2. Evaluation Methods and Experiments

The designed convolutional autoencoder consists of an encoder and a decoder, illustrated in Figure 3.10., which are defined as sequential models using 1D convolutional, max pooling, Batch normalization, and upsampling layers. The number in the convolutional layer denotes the number of filters in each layer. The encoder takes as input a 1D signal of shape (360, 1) and applies a series of convolutional filters, along with activation functions like ReLU and padding. The input is downsampled using Max pooling layers, and the activations are normalized using Batch normalization layers. Similar definitions are applied to the decoder. However, convolutional layers are used for upsampling rather than downsampling. The final convolutional layer has one filter and a ReLu activation, and upsampling layers are employed to expand the input's dimensionality.

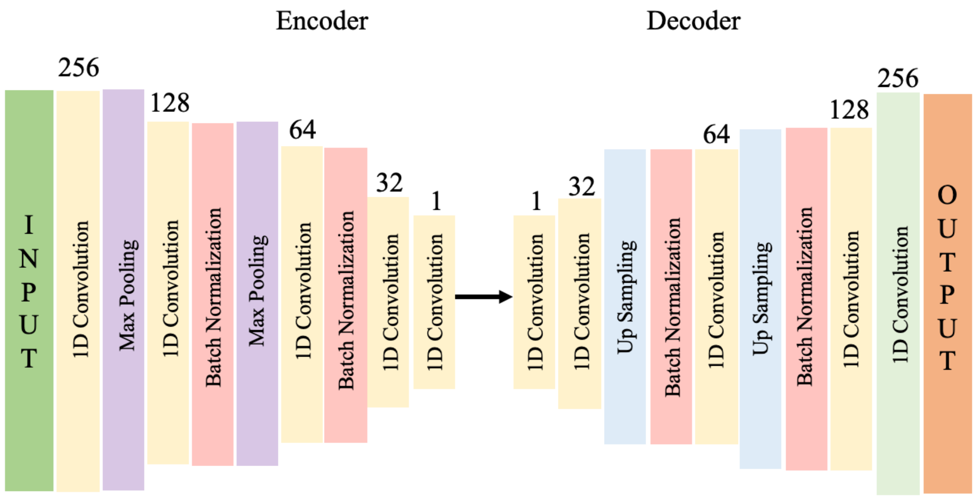


Figure 3.10. Proposed convolutional autoencoder architecture.

Source: Author

3.2.2.1 Arrhythmia detection with custom-designed convolutional AE

The wavelet convolutional autoencoder class consists of an encoder and a decoder, defined as sequential models using convolutional layers, wavelet transformation layers, and other layers like batch normalization, as depicted in Figure 3.11. The encoder takes as input a 1D signal of shape (360, 1) and incorporates several convolutional layers with different configurations, including activation functions like ReLU, padding, and wavelet transformations using the WaveTF library. The decoder is defined similarly to the Convolutional autoencoder model.

Table 3.7. The Reconstruction Coefficients of The Wavelet Functions Used.

	Daubechies 2	***Coif 2***	***Bior 3.5***
$g_0[0]$	0.48296	0.016387	0.0
$g_0[1]$	0.836516	-0.041465	0.0
$g_0[2]$	0.2241439	-0.067373	0.0
$g_0[3]$	-0.1294095	0.3861101	0.0
$g_0[4]$		0.8127236	0.1767767

$g_0[5]$	0.417005	0.5303301
$g_0[6]$	-0.0764886	0.5303301
$g_0[7]$	-0.0594344	0.1767767
$g_0[8]$	0.02368017	0.0
$g_0[9]$	0.00561144	0.0
$g_0[10]$	-0.0018232	0.0
$g_0[11]$	-0.0007206	0.0

The wavelet transform block is in Figure 3.11. involves the calculation of one-dimensional Discrete Wavelet Transform of the given signal. WaveTF library conducts DWT by convolving inputs with low-pass and high-pass filter coefficients. These are matrix operations in the Tensorflow environment (Versaci, 2020). Initially, two mother wavelet options existed in the WaveTF library: Haar wavelet and Daubechies 2. In this study, different mother wavelets reported to have high performance in biomedical signal analysis were adapted to the algorithm. The used wavelet coefficients are listed in Table 3.7. the models were trained with different parameters to reach the optimal performance heuristically. We used wavelet transform as a layer in our autoencoder model because of its unique capacity to capture both time and frequency domain characteristics, allowing for excellent representation of transient patterns and variable frequency components.

3.2.2.2 Classification with MLP

The classifier model architecture includes several layers, including dense layers with ReLU activation functions, batch normalization layers for normalization, dropout layers to prevent overfitting, and a used softmax function for activation layer for multi-class classification. The model is compiled with binary cross-entropy loss and the SGD optimizer. In this study, we used the Multi-Layer Perceptron (MLP) architecture for classification, taking advantage of its capacity to understand subtle patterns and non-linear correlations within the data. The MLP's adaptability in modeling complicated decision boundaries and its effectiveness in handling varied classification tasks made it an attractive and powerful choice for our study, resulting in correct predictions.

3.2.3. Results

After the model structures were created, the ECG beats in each class are shown in Table 3.6. were trained with the autoencoder model, each beat taken from the autoencoder model separated into train and test with a 0.20 ratio. The following procedure was the classification stage with the Multilayer Perceptron (MLP) model, and accuracy metrics were examined by feeding encoded data into the designed classification model. In this study, we applied the Adam as an optimizer, a well-known gradient-based optimization technique for neural network training. The learning rate of the optimizer is set to 1e-3, which determines how much the model weights are updated during training. Then, by defining the optimizer and loss function for training, the model is compiled. The loss function is set to mean absolute error (MAE), a typical loss function for problems involving classification. A comparison of the autoencoder model using different wavelets is shown in Table 3.8.

Table 3.8. Performance Metrics Results

Wavelet	***Batch Size***	***Accuracy (%)***	***Precision (%)***	***Recall (%)***	***F1 Score (%)***
None	32	99.5	98.9	97.2	98.9
	64	98.0	97.8	97.2	97.5
	128	99.6	99.0	99.0	99.0
Db2	32	99.3	98.4	98.1	98.2
	64	95.1	96.3	87.3	89.3
	128	99.5	98.9	98.8	98.9
Bior 3.5	**32**	**99.8**	**99.8**	**99.7**	**99.7**
	64	99.6	99.5	99.1	99.3
	128	99.5	99.4	99.0	99.2

Coif 2	32	90.6	86.4	74.3	70.6
	64	98.3	97.6	96.5	97.0
	128	99.7	99.2	99.1	99.2

In Table 3.8., the first column shows which mother wavelet was used. In the first experiment, the structure in Figure 3.11. with no wavelet layer was employed. Different batch sizes were used for each experiment, and the performance results were listed. We observed that Bior 3.5, with a batch size of 32, achieves the highest accuracy of 99.8%, followed closely by no wavelet with a batch size of 32. When the other performance measures of the Bior 3.5 model were considered, out of all positive predictions, precision is defined as the percentage of true positive predictions. The highest result of the Bior 3.5 model indicates a high proportion of correct positive predictions. Recall also had the highest rate of true positive predictions out of all actual positive occurrences, suggesting that it can correctly identify many positive examples. The F1 score offers a balanced assessment of both measures and is calculated as the harmonic mean of recall and precision. Bior 3.5, with a batch size of 32, has the highest macro F1 score (99.7%), indicating a good balance between precision and recall. It can be noted that lower precision-recall results were obtained in the no-wavelet case.

It's worth noting that the performance of each wavelet type varies across different batch sizes. For example, Coif 2, with a batch size of 32, shows lower performance compared to other batch sizes, particularly in precision, recall, and F1 scores. The proposed structure was compared with several advanced deep-learning algorithms for classifying arrhythmias. As can be observed from Table 3.9., different feature extraction and classification methods were employed for arrhythmia detection. The most important outcome of this research is achieving higher performance with a custom-designed wavelet-based autoencoder model. Another critical point is the size of the dataset. Our algorithm has been proven to have higher classification rates with more extensive data, which includes beats from different datasets with different properties (Eravcı & Ozkurt, 2023).

Table 3.9. Performance Of the Deep Learning Arrhythmia Detection Algorithms

Authors	***Total Beat***	***Method***	***Accuracy (%)***
(Wu, Lu, Yang, & Wong, 2021)	32,422	CNN	97.41
(Acharya et al., 2017)	109,449	CNN	94.03
(Liu et al., 2022)	97,300	LSTM AE ANN	99.00
(Sahoo, Kanungo, Behera, & Sabut, 2017)	109,494	DWT SVM	98.39
(Mohonta, Motin, & Kumar, 2022)	7,500	CWT 2DCNN	99.65
(Ojha et al., 2022)	97,861	AE SVM	99.53
Proposed Method *Bior 3.5*	138,474	Wavelet-based Convolutional AE MLP	99.80

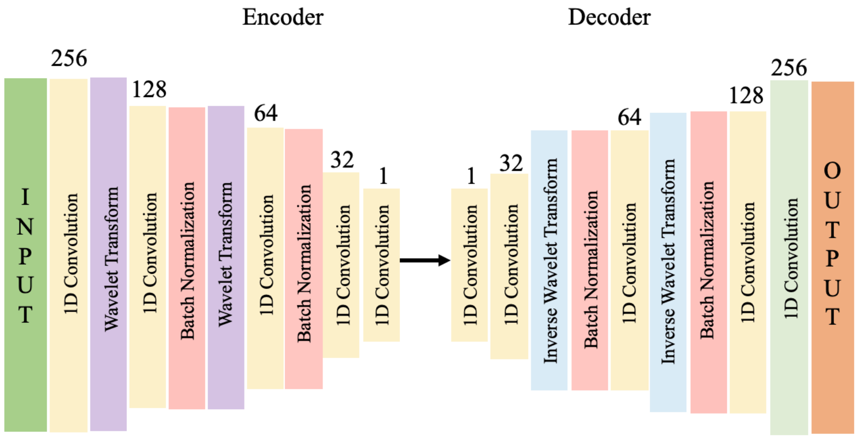


Figure 3.11. Wavelet convolutional autoencoder architecture.

Source: Author

3.2.4. Summary

Effective detection of cardiac arrhythmia types is a significant problem, especially in analyzing long medical records and wearable health tracking devices. In this study, we suggested a convolutional autoencoder model to categorize five distinct kinds of ECG beats. Our model takes advantage of the wavelet transform, which reveals the time-frequency domain characteristics of signals. Thanks to the autoencoder model, the features contained in the data are encoded compactly and can be analyzed with a simple classifier. Furthermore, a high performance was obtained with extensive data collected from different datasets. This is an indication of the robustness of the proposed model.

The selection of the mother wavelet is still an important design parameter. Usually, the best mother wavelet is selected experimentally, as in this work. Bior 3.5 demonstrated a successful performance for the problem at hand. In the upcoming studies, a signal-adaptive wavelet selection algorithm is proposed.

3.3. Arrhythmia Anomaly Detection with WCAE

Arrhythmia classification relies on a deep learning technique that uses a customized wavelet-based convolutional autoencoder (WBCAE) for this study. This anomaly detector integrates time-frequency analysis of wavelets and convolutional autoencoders for feature learning purposes. The analysis reveals the potentiality of deep learning models in arrhythmia detection and wavelet approaches to biomedical signal processing. To conclude, this combination of WBCAE with arrhythmia data presents an important way to improve anomaly detection by taking advantage of both the strengths associated with deep learning as well as advanced signal processing strategies.

3.3.1. Dataset

In this study, the same datasets were used as in 3.2. but different sizes of samples. In addition to normal sinus rhythm, the dataset includes arrhythmia types such as premature ventricular contraction, left bundle branch block, right bundle branch block, and finally atrial premature contractions. The dataset comprises 360 samples, which include 128,399 segments of arrhythmia data, and 75,011 segments of normal sinus rhythm data as can be seen in Table 3.10.

Table 3.10 Number of different data classes used in the study

Arrhythmia Types	**Beat Size**
Left Bundle Branch Block (LBBB)	13,373
Right Bundle Branch Block (RBBB)	12,092
Normal Sinus Rhythm (NSR)	75,011
Premature Ventricular Contraction (PVC)	98,497
Atrial Premature Contraction (APC)	4,437
Total	203,41

3.3.2. Evaluation Methods and Experiments

To thoroughly evaluate the performance of the proposed Wavelet-Based Convolutional Autoencoder (WBCAE) in detecting arrhythmias several evaluation metrics and methods were employed. These measurements offer a thorough comprehension of the model's efficacy and reliability in different scenarios. The anomaly detection method is mentioned in 4.1.2. was used. That is, If the error is less than the given threshold value, it is labeled as normal sinus rhythm; if it is greater than the given threshold value, it is labeled anomaly.

The experiments were carefully planned to assess the performance of the WBCAE in different scenarios and forms. The main objective was to optimize the model for arrhythmia detection as well as comparing its performance of wavelet families. Another aim of this study is to asses the performance of the anomaly detection approach in unbalanced datasets.

For efficient processing of the dataset, WBCAE was implemented and trained on a high-performance computing cluster of Google Colab. The experiments entailed testing various configurations of the model, including choosing wavelets and hyperparameters.

Hyperparameters:

- Number of epochs: 50
- Batch size: 60
- Learning rate: Optimized through grid search and experimentation
- Wavelets: Various wavelets, including Haar, Daubechies (db3, db4), Biorthogonal (bior 3.5), Coiflet (coif2), and Symlet (sym4), were tested.

Model Architecture

The WBCAE architecture consists of an encoder and a decoder. The encoder compresses the input ECG signals into a latent representation, while the decoder reconstructs the signal, emphasizing the detection of anomalies. The model used in Figure 3.4.

Results

The measures listed above were used to assess the WBCAE's performance. The results varied depending on the wavelet used, highlighting the impact of wavelet selection on model performance.

Table 3.11. Experiment Results.

Wavelet	*Arrhythmia*	*Accuracy %*	*Precision %*	*Recall %*	*F1 Score %*
db3	RBBB	98,87	99,92	97,54	98,72
	LBBB	98,29	99,87	96,50	98,16
	APC	99,38	99,11	98,17	98,64
	PVC	99,27	99,99	99,17	99,58
db4	RBBB	99,53	99,73	99,22	99,47
	LBBB	98,15	99,83	96,24	98,00
	APC	98,87	99,09	95,92	97,48
	PVC	99,60	99,89	99,66	99,77
sym4	RBBB	99,54	99,81	99,16	99,49
	LBBB	99,34	99,74	98,86	99,30
	APC	99,80	99,44	99,68	99,56
	PVC	99,75	99,92	99,79	99,86
bior3.5	RBBB	99,42	99,87	98,83	99,34
	LBBB	98,79	99,67	97,75	98,70
	APC	98,13	98,71	93,01	95,78
	PVC	99,62	99,92	99,64	99,78
coif2	RBBB	99,63	99,73	99,44	99,59
	LBBB	99,41	99,47	99,29	99,38
	APC	99,57	99,79	98,31	99,05
	PVC	99,55	99,90	99,58	99,74

3.3.3. Summary

This study was achieved with a dataset containing normal heartbeats and four different types of arrhythmias (e.g. RBBB , APC, LBBB and PVC) with an anomaly detection approach and using the WBCAE method. As a first step, the data obtained from the electrocardiogram (ECG) signals were divided into normal and arrhythmic signals and each pulse signal was processed separately. The anomaly change is displayed in data wavelets and subjected to time-frequency analysis. Five different wavelets were used for this purpose: db3, db4, sym4, bior3.5 and coif2. Each wavelet analyzed the ECG signals from a different perspective and revealed important features.

These features are given as input to the WBCAE model, and the aim is to preserve important features and create an optimized feature space by reducing the course of the model to a low-dimensional representation . This way it is easier to distinguish between normal and arrhythmic. The outputs of the WBCAE model were then further examined with anomaly detection methods, and a specific threshold value was used to distinguish each arrhythmia (RBBB , APC, LBBB and PVC) from normal signals. This method ensures the correct organization of normal and normal (arrhythmic) functioning.

Finally, the performance of the model was evaluated with metrics such as Accuracy , Precision, Recall and F1 Score. These measurements measure the overall accuracy of the model, record positive predictions, and distinguish between types of arrhythmias. The results showed that WBCAE and the abnormality detection approach were highly successful in distinguishing between normal and four different arrhythmias.

The WBCAE introduced in this study represents remarkable progress in arrhythmia detection especially in feature extraction, making use of the strengths of wavelet transforms and deep learning. The thorough evaluation and experimental results demonstrate the model's robustness, flexibility, and high performance. The study highlights the power of wavelet methods combined with convolutional autoencoders in biomedical signal processing, which offers effective means for remote patient monitoring as well as early disease diagnosis.

4. CHAPTER: CONCLUSIONS AND FUTURE WORK

In this thesis, we presented studies on detecting atrial fibrillation (AF) and other heart arrhythmias using wavelet based convolutional autoencoders. Both studies improved arrhythmia diagnosis and classification from ECG signals by utilizing wavelets and convolutional autoencoders' distinct abilities.

In the first study wavelet-based convolutional autoencoder was used in anomaly detection mode designed exclusively for AF detection. Atrial fibrillation is distinguished by the absence of P waves and irregular heartbeats, which make it difficult to diagnose due to the nonstationary nature of ECG signals. To solve these issues, we combined wavelet transformations and convolutional autoencoders. The effect of the selected wavelet and the size of the signal window were analyzed heuristically. It is observed that Symlet 4 wavelet with 256 sample windows creates the best results with an accuracy of 95.44% and an f1 score of 95.23% for public datasets which shows the effectiveness of the algorithm.

In the second study, the wavelet-based convolutional autoencoder was used as a feature extractor for different arrhythmias. This method combined the time-frequency analysis skills of wavelet transformations with the feature extraction power of convolutional autoencoders. The WBCAE retrieved the essential features for classification using MLP. The model's results illustrate its ability to accurately detect and classify numerous types of heartbeats with minimal errors. One of the study's key findings was the significance of wavelet selection. The Bior 3.5 wavelet outperformed the other wavelets tested, emphasizing the significance of selecting the right wavelet for the job.

The third study conducts arrhythmia detection by using anomaly detection for different types of arrhythmias. The anomaly detection method can distinguish normal and abnormal ECG signals. The performance of our anomaly detection model has shown significant improvement, especially in detecting rare arrhythmia types. With this approach, it is possible to detect abnormal functioning earlier and more accurately, and by doing so, it enables the early implementation of potentially life-saving interventions. Future developments, testing this method on larger datasets and different arrhythmia types will contribute to learning the model's capacity to generalize. This study highlights the enhancement of anomaly detection techniques in medical data analysis and provides new research and application opportunities in the healthcare field. The model's resistance to imbalanced datasets and reasonable processing complexity increase its suitability for real-time applications such as wearable health monitoring devices.

While the studies have shown promising results, there are several areas for future study to improve the proposed models further and extend their applicability:

- ***Wavelet Selection Optimization***: Develop adaptive algorithms for selecting the most suitable wavelet based on the characteristics of the input signal to further enhance classification performance.
- ***Reducing Computational Complexity***: Optimize the models to reduce computational demands, making them more suitable for real-time applications and deployment on wearable devices.
- ***Integration with Wearable Devices*: ** Adapt the models for wearable health monitoring devices, focusing on real-time processing and energy efficiency.
- ***Handling Diverse and Noisy Data:*** Enhance the models' robustness to noise and artifacts through advanced preprocessing and data augmentation techniques.
- ***Clinical Validation***: Conduct clinical trials to validate the models' performance in real-world settings, ensuring their reliability and effectiveness for clinical diagnostics.

By addressing these issues, we want to increase the accuracy, efficiency, and applicability of wavelet-based convolutional autoencoders for cardiac arrhythmia detection and other biomedical signal-processing tasks. The ultimate goal is to create complete diagnostic tools that may be extensively used in clinical practice and integrated into wearable health monitoring devices, allowing for early and accurate diagnosis of cardiac conditions and improving patient outcomes.

5. CHAPTER: REFERENCES

A Goldberger, A. L., Amaral, L. A., Glass, L., Hausdorff, J. M., Ivanov, P. C., Mark, R. G., Mietus, J. E., Moody, G. B., Peng, C. K., & Stanley, H. E. (2000). PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. *Circulation*, *101*(23), E215–E220. https://doi.org/10.1161/01.cir.101.23.e215

Acharya, U. R., Oh, S. L., Hagiwara, Y., Tan, J. H., Adam, M., Gertych, A., & San Tan, R. (2017). A deep convolutional neural network model to classify heartbeats. *Computers in biology and medicine*, *89*, 389-396.Addison, P. (2005). Wavelet transforms and the ECG: a review. Physiological

Addison, P. S. (2017). *The illustrated wavelet transform handbook: introductory theory and applications in science, engineering, medicine and finance*. CRC press.

Aggarwal, C. C. (2018). Neural networks and deep learning.

American Heart Association. (2022). Prevention and Treatment of Arrhythmia. Retrieved from https://www.heart.org/en/health-topics/arrhythmia/prevention--treatment-of-arrhythmia

American Heart Association. (2023). What is atrial fibrillation (AF or AFib)?. Retrieved from https://www.heart.org/en/health-topics/atrial-fibrillation/what-is-atrial-fibrillation-afib-or-af

Brugada, J., Brugada, R., & Brugada, P. (1998). Right bundle-branch block and ST-segment elevation in leads V1 through V3: a marker for sudden death in patients without demonstrable structural heart disease. Circulation, 97 5, 457-60 .

Chen, Y., Zhang, C., Liu, C., Wang, Y., & Wan, X. (2022). Atrial fibrillation detection using a feedforward neural network. *Journal of Medical and Biological Engineering*, *42*(1), 63-73.

Cheng, Y., Hu, Y., Hou, M., Pan, T., He, W., & Ye, Y. (2020). Atrial fibrillation detection directly from compressed ECG with the prior of measurement matrix. *Information*, *11*(9), 436.

Chung, C. T., Lee, S., King, E., Liu, T., Armoundas, A. A., Bazoukis, G., & Tse, G. (2022). Clinical significance, challenges and limitations in using artificial intelligence for electrocardiography-based diagnosis. *International journal of arrhythmia*, *23*(1), 24.

Clifford, G. D., Liu, C., Moody, B., Li-wei, H. L., Silva, I., Li, Q., ... & Mark, R. G. (2017, September). AF classification from a short single lead ECG recording: The PhysioNet/computing in cardiology challenge 2017. In *2017 Computing in Cardiology (CinC)* (pp. 1-4). IEEE.

CompThree. (n.d.). ae.png [Image]. CompThree. https://www.compthree.com/images/blog/ae/ae.png

Daubechies, I. (1992). Ten Lectures on Wavelets. SIAM.

Eravcı, Ö., & Özkurt, N. (2023, September). Arrhythmia Detection with Custom Designed Wavelet-based Convolutional Autoencoder. In *2023 International Conference on Innovations in Intelligent Systems and Applications (INISTA)* (pp. 1-5). IEEE.

Faust, O., Shenfield, A., Kareem, M., San, T. R., Fujita, H., & Acharya, U. R. (2018). Automated detection of atrial fibrillation using long short-term memory network with RR interval signals. *Computers in biology and medicine*, *102*, 327-335.

G. Moody, A new method for detecting atrial fibrillation using RR intervals[J], Comput. Cardiol. (1983) 227–230.

Moody, G. B., & Mark, R. G. (2001). The impact of the MIT-BIH arrhythmia database. *IEEE engineering in medicine and biology magazine*, *20*(3), 45-50.

Goldberger, A. L., Goldberger, Z. D., & Shvilkin, A. (2017). *Clinical Electrocardiography: A Simplified Approach: Clinical Electrocardiography: A Simplified Approach E-Book*. Elsevier Health Sciences.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press. [pp. 204, 228]

Haykin, S. (1998). Neural Networks: A Comprehensive Foundation. Prentice Hall.

Mohrman, D., & Heller, L. J. (2006). *Cardiovascular physiology*. McGraw-Hill Professional.

Hu, Y., Zhao, Y., Liu, J., Pang, J., Zhang, C., & Li, P. (2020). An effective frequency-domain feature of atrial fibrillation based on time–frequency analysis. *BMC Medical Informatics and Decision Making*, *20*, 1-11.

Ruth Isabels, K., Mrudula Devi, K., Anand, R., Athe, R., Chowdhury, S. S., & Pund, S. S. (2023). An Intellectual Fusion Classification Prototypical for an Imbalanced Electrocardiogram Data. *SN Computer Science*, *4*(6), 721.

Issa, M. F., Yousry, A., Tuboly, G., Juhasz, Z., AbuEl-Atta, A. H., & Selim, M. M. (2023). Heartbeat classification based on single lead-II ECG using deep learning. *Heliyon*, *9*(7).

Klabunde, R. (2011). *Cardiovascular physiology concepts*. Lippincott Williams & Wilkins.

Kumar, G. M., Reddy, K. R., Reddy, M. H., & Dharavath, K. (2023, October). ECG Based Arrhythmia Detection Using CNN. In *2023 International Conference on Self Sustainable Artificial Intelligence Systems (ICSSAS)* (pp. 126-131). IEEE.

Li, W., Cao, Y., Li, L., & Hou, S. (2023). Orthogonal Wavelet Transform‐Based Gaussian Mixture Model for Bearing Fault Diagnosis. *Discrete Dynamics in Nature and Society*, *2023*(1), 1307845.

Lilly, L. S. (2012). *Pathophysiology of heart disease: a collaborative project of medical students and faculty*. Lippincott Williams & Wilkins.

Liu, P., Sun, X., Han, Y., He, Z., Zhang, W., & Wu, C. (2022). Arrhythmia classification of LSTM autoencoder based on time series anomaly detection. *Biomedical Signal Processing and Control*, *71*, 103228.

Mayo Clinic. (2022). Ventricular fibrillation. Retrieved from https://www.mayoclinic.org/diseases-conditions/ventricular-fibrillation/symptoms-causes/syc-20364523

Mayo Clinic. (2023). Heart arrhythmia. Retrieved from https://www.mayoclinic.org/diseases-conditions/arrhythmia/diagnosis-treatment/drc-20350668

McBride, D., Mattenklotz, A. M., Willich, S. N., & Brüggenjürgen, B. (2009). The costs of care in atrial fibrillation and the effect of treatment modalities in Germany. *Value in Health*, *12*(2), 293-301.

McComb, J. M. (2006). Sudden cardiac death and ventricular arrhythmias. *Medicine*, *34*(7), 268-273.

Mohonta, S. C., Motin, M. A., & Kumar, D. K. (2022). Electrocardiogram based arrhythmia classification using wavelet transform with deep learning model. *Sensing and Bio-Sensing Research*, *37*, 100502.

Ochiai, K., Takahashi, S., & Fukazawa, Y. (2018, August). Arrhythmia detection from 2-lead ECG using convolutional denoising autoencoders. In *Proc. KDD* (pp. 1-7).

Ojha, M. K., Wadhwani, S., Wadhwani, A. K., & Shukla, A. (2022). Automatic detection of arrhythmias from an ECG signal using an auto-encoder and SVM classifier. *Physical and engineering sciences in medicine*, *45*(2), 665-674.

Olah, C. (2014). Understanding Convolutions. Retrieved from https://colah.github.io/posts/2014-07-Understanding-Convolutions/

Rajkumar, A., Ganesan, M., & Lavanya, R. (2019, March). Arrhythmia classification on ECG using Deep Learning. In *2019 5th international conference on advanced computing & communication systems (ICACCS)* (pp. 365-369). IEEE.

Rasmussen, S. M., Jensen, M. E., Meyhoff, C. S., Aasvang, E. K., & Słrensen, H. B. (2021, November). Semi-supervised analysis of the electrocardiogram using deep generative models. In *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)* (pp. 1124-1127). IEEE.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18* (pp. 234-241). Springer International Publishing.

Chen, S., Hua, W., Li, Z., Li, J., & Gao, X. (2017). Heartbeat classification using projected and dynamic features of ECG signal. *Biomedical Signal Processing and Control*, *31*, 165-173.

Sahoo, S., Kanungo, B., Behera, S., & Sabut, S. (2017). Multiresolution wavelet transform based feature extraction and ECG classification to detect cardiac abnormalities. *Measurement*, *108*, 55-66.

Sattar, Y., & Chhabra, L. (2020). Electrocardiogram.[Updated 2020 Jul 31]. *StatPearls [Internet]. Treasure Island (FL): StatPearls Publishing*.

Shaik, J., & Bhavanam, S. N. (2023). Arrhythmia Detection Using ECG-Based Classification with Prioritized Feature Subset Vector-Associated Generative Adversarial Network. *SN Computer Science*, *4*(5), 519.

Tihonenko, V., Khaustov, A., Ivanov, S., Rivin, A., & Yakushenko, E. (2008). St Petersburg INCART 12-lead arrhythmia database. PhysioBank PhysioToolkit and PhysioNet

Versaci, F. (2020). WaveTF: A Fast 2D Wavelet Transform for Machine Learning in Keras. ICPR Workshops.

Vincent, P. (2011). A connection between score matching and denoising autoencoders. *Neural computation*, *23*(7), 1661-1674.

Wei, T. R., Lu, S., & Yan, Y. (2022). Automated atrial fibrillation detection with ECG. *Bioengineering*, *9*(10), 523.

WHO. (2021). Cardiovascular diseases (CVDs). https://www.who.int/newsroom/fact-sheets/detail/cardiovascular-diseases-(cvds)

Widmaier, E., Raff, H., & Strang, K. T. (2022). *Vander's human physiology*. McGraw-Hill US Higher Ed USE.