Master thesis in Applied Computer and Information Technology (ACIT)

Biomedical Engineering

An Investigation into using Deep Convolutional Neural Networks for ECG Analysis

Mohammad Awais Hameed

Supervisors: Pål Halvorsen, Steven Hicks, Vajira Thambawita

Department of Computer Science Faculty of Technology, Art and Design

Faculty of Technology, Art and Design

Contents

Dedication

I would like to dedicate this work to my family and friends for the support throughout the writing process.

Acknowledgement

I would like to express my sincere gratitude to my supervisors for all the guidance and support with the thesis work.

Abstract

In this day and age, the fascination surrounding deep learning and AI is at its absolute peak. Both in terms of hype and controversy the current interest level is unprecedented, with exciting developments happening at a lightning pace. Yet, as is often the case when capitalist motives are the driving force behind progress, use-cases that could potentially save lives are left behind. Specifically, deep learning has particularly exciting potential in the field of ECG analysis. In our research, we investigated the most prominent model type for this purpose, namely the Convolutional Neural Network (CNN).

To that end, a deep learning pipeline was developed based on the renowned PTB-XL dataset. The CNN was tasked with classifying ECG signals according to the 5 diagnostic classes; Normal, Myocardial Infarction (MI), ST/T Change, Conduction Disturbance (CD) and Hypertrophy. Several experiments testing factors such as Pooling and Batch Normalization were conducted. Simultaneously, different levels of parameters such as dropout and hidden dimensions were also examined. Our findings indicated that Average Pooling was the most influential factor and that its combination with Batch Normalization produced the most effective results.

The thesis also discusses ethical considerations regarding the use of such models in clinical practice, and approaches aimed at alleviating privacy concerns, such as synthetic datasets. Lastly, we emphasize the importance of developing explainable methods to better facilitate the use of deep learning models in the medical domain. In this context, the inclusion of doctors and radiologists can be considered of utmost importance.

Keywords: ECG analysis; CNN; Explainable AI

List of Figures

List of Tables

Chapter 1

Introduction

"The field of electrocardiography (ECG) analysis has traditionally relied on manual interpretation by experts. However, with the advent of deep learning techniques, there is an opportunity to improve the accuracy and efficiency of ECG analysis. This thesis explores the use of deep learning methods for ECG analysis, including the development and evaluation of models for various ECG-related tasks such as arrhythmia detection and heart disease diagnosis. The goal of this research is to demonstrate the potential of deep learning for ECG analysis and to identify areas for future work in this field."

To appreciate the potential of deep learning as a tool, consider that the previous paragraph was written in its entirety by the deep learning-based chatbot; ChatGPT using the simple prompt "write an introduction for a master thesis where the topic is 'deep learning for ECG analysis" [36]. While the use of deep learning in such language models has seen significant development, the application of this technology in biomedical engineering is still in its infancy. Specifically, for tasks such as ECG analysis, the majority of the work is done manually, and as mentioned, there is a clear potential for improvement in efficiency and accuracy.

For those unfamiliar with the field, deep learning can appear to be a singular, all-encompassing term. However, it is essential to understand that there are many different model types, and choosing the right one is often critical for a successful outcome.

In the context of ECG analysis, the model type that stands out among the competitors is a so-called Convolutional Neural Network (CNN). Both in the case of well-funded research studies and personal hobby projects found online, the application of these models has led to particularly promising results [18, 22]. Specifically, it is the ability of the CNN to account for spatial features that make it a particularly good fit for ECG analysis. Needless to say, further investigation of the most effective model type is key, if we are to uncover the full capabilities of deep learning for ECG analysis.

Moreover, a typical CNN consists of many different structural factors, and the configuration of these are essential to reach optimal performance. Thus, expanding our knowledge of the influence of various factors of the CNN for the task of ECG analysis is vital. Such factors may include the core components/building blocks of the network, techniques used to regulate the learning of the network, and other parameters associated with Deep Learning (DL) models.

At the same time, one of the key challenges in the application of CNN´s as well as other DL algorithms is the interpretability of the model's decisions. Unlike natural language processing models, such as ChatGPT, which can rely on a "black box" approach, it is crucial for the medical community, including doctors and radiologists, to have a clear understanding of the features and decision-making process of ECG analysis models. Without this transparency, there is a risk that medical professionals may be hesitant to use these models in practice.

Additionally, there is a pressing concern related to the privacy of ECG data that needs to be addressed. Since health data is considered especially sensitive, there are a number of regulations that prohibit the free use of such data in the development of DL models. Specifically, in the EU, the General Data Protection Regulation (GDPR) restricts the flow of information related to health to ensure patient anonymity [53]. Even in cases where said data has been anonymized, there are limitations in terms of the exchange of data between different countries since the combination of certain variables may lead to individual identification [14].

Considering the factors mentioned, it is apparent that the field of healthcare is in great

need of DL models that are both explainable and protective of privacy-related concerns. In this regard, it is therefore evident that further deepening the understanding of DL-models and in particular CNN´s, may provide several benefits in the area of healthcare and medical services. Thus, an assessment of the different structural factors related to the CNN, such as layers, parameters and regularization techniques can be considered to be vital.

1.1 Motivation and research question

The primary motivation behind this thesis is to contribute to the existing body of knowledge on the application of CNN´s in ECG analysis. In doing so, the aim is to further the understanding of this prominent model type and demonstrate the potential for progress in its effective implementation.

With DL and Artificial Intelligence (AI) being a rapidly growing area of research, this thesis may also generate increased public interest and attention to a particularly important use case.

Research question

To what extent can Convolutional Neural Networks be used for ECG analysis, and what structural factors influence their effectiveness?

The selected objectives allowed for an open exploration of factors related to CNN´s in ECG analysis, while not limiting the scope of our project. Furthermore, discussing various explainable methods for DL can be regarded as highly important, as the lack of explainability poses a significant obstacle to the adoption of CNN´s in ECG analysis.

1.2 Objectives

- The main objective of this research is to implement a CNN for ECG analysis from scratch. This will provide the opportunity to compare the model with existing work and experiment with various aspects of the structure to investigate potential improvements and gain insights into the performance of CNN´s.

- Secondly, assessing the feasibility and effectiveness of explainable methods is also a point of interest in this thesis.

1.3 Structure of the thesis

The structure of this thesis is divided into four main sections: background, method, results, and discussion. In the background section, the background information and theoretical framework for the thesis are presented. This includes a review of relevant literature and current state-of-the-art solutions. The method section describes the research design and methodology used for the development and evaluation of the CNN. The findings are thereafter presented in the results section. In the discussion section, the findings are interpreted in light of privacy and explainability issues. Finally, the conclusion summarizes the main findings and contributions of the thesis.

Chapter 2

Background

The following chapter presents an overview of the background and theoretical foundations of DL for ECG analysis. This includes the introduction of key concepts, relevant regulations, as well as a literature review on current state-of-the-art approaches.

2.1 Concepts

The following terms will be used in this thesis with the corresponding explanations;

2.1.1 Electrocardiogram:

The Electrocardiogram (ECG) is a measure of the electrical activity of the heart. Specifically, the electrical activity is displayed as time-series data and contains different features/segments as seen in Figure 2.1. The timing and the amplitude of these features carry clinical information about the state of the heart. Useful for diagnostic purposes, ECGs can be deployed to detect different cardiac disorders such as arrythmia and heart attacks [35]. ECGs are considered cost effective, non-invasive and practical in a variety of medical settings [32]. Figure 2.2 shows the most common setup for procuring ECG data, namely through 10 electrodes placed on the body of the subject.

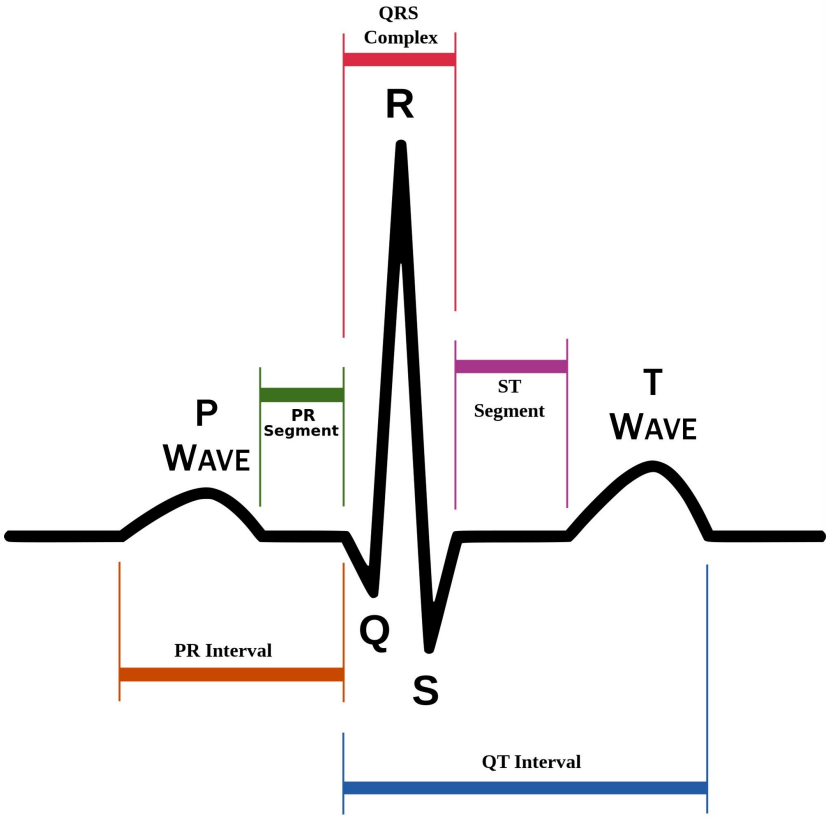

Figure 2.1: Figure showing the ECG complex. Segments are named P, Q, R, S, and T. [5]

2.1.2 Convolutional Neural Network:

A Convolutional Neural Network (CNN) is a type of artificial neural network often used for image and audio recognition. It consists of multiple layers of filters that process input data such as an image. The filters are designed to learn meaningful features from the input data, allowing the network to recognize patterns and make decisions [40]. These features are then passed through fully connected layers to make a prediction as visualized in Figure 2.3. CNN´s have been successful in a variety of tasks such as image classification, object detection, and image generation [13][57].

Convolution

As the name suggests, the most essential part of the CNN is the convolutional layer. Convolution refers to the process in which the input data is traversed and features are extracted. This operation occurs when a filter with a given kernel size passes through

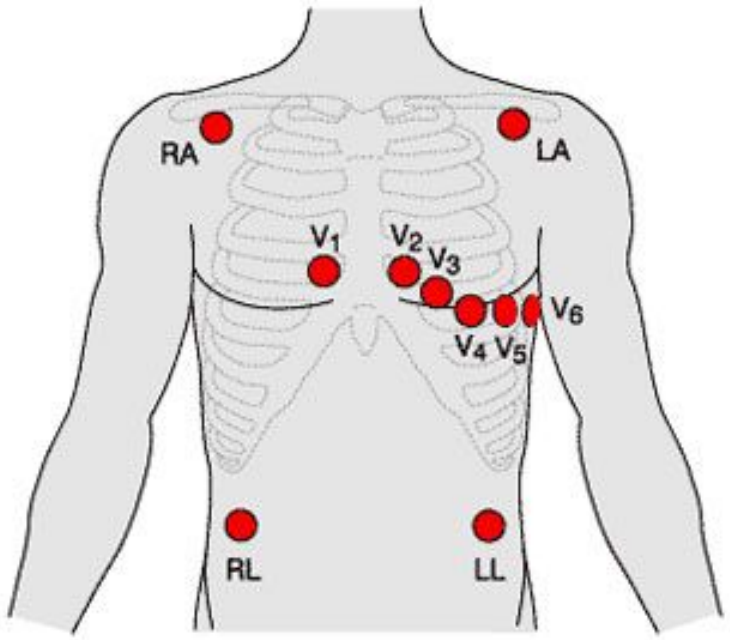

Figure 2.2: Figure showing the placement of the 10 electrodes for a standard 12 lead ECG configuration. [55]

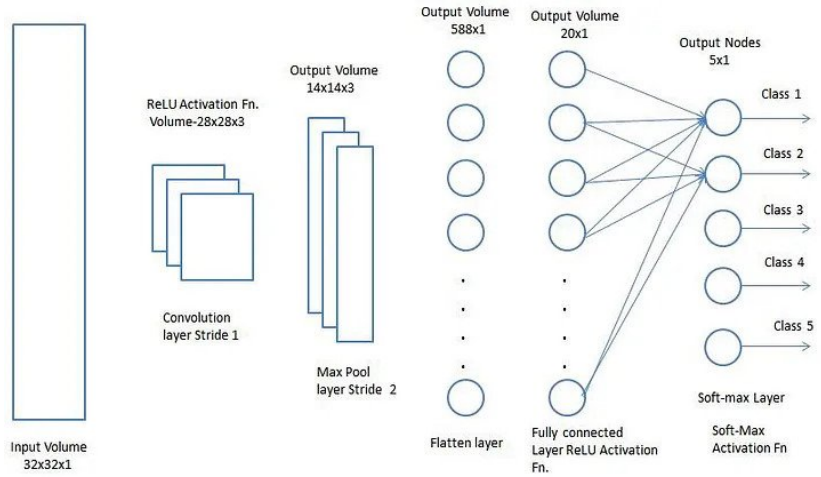

Figure 2.3: Figure showing the setup of a standard CNN designed to make predictions in accordance with 5 classes. [40]

the input with a certain stride length. When the filter is first applied in a region of the input, it calculates the dot product of the pixels within that region. The calculated value is then fed to an output array. The filter thereafter shifts, and repeats the process until the entire input has been covered. The resulting output array is known as the feature map [13]. The operation can best be understood through its most common

application which is a 2D convolution of an image, as demonstrated in Figure 2.4.

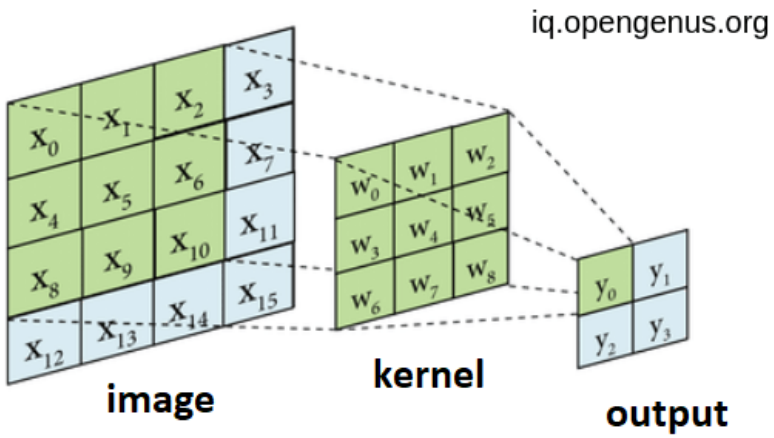

Figure 2.4: Figure showing 2D convolution. This type of convolution is often applied on images.[27]

Lastly, the ReLU (Rectified Linear Unit) transformation is typically applied to each feature map after every convolution to introduce nonlinearity to the DL model.

1D, 2D and 3D convolution

Apart from the popular 2D convolution, there are other types of convolutions such as 1D convolution and 3D convolution that serve their specific purposes [52].

As mentioned, CNN´s have traditionally been found to be effective at audio recognition. Since ECG signals bear resemblance to audio, it is natural that 1D convolution, which has been successful in audio analysis, has also shown promising results in ECG analysis. In contrast to 2D convolution, where the input signal is a two-dimensional matrix, such as an image, 1D convolution processes a one-dimensional input signal. The resulting feature map is also one-dimensional. Figure 2.5 illustrates this concept.

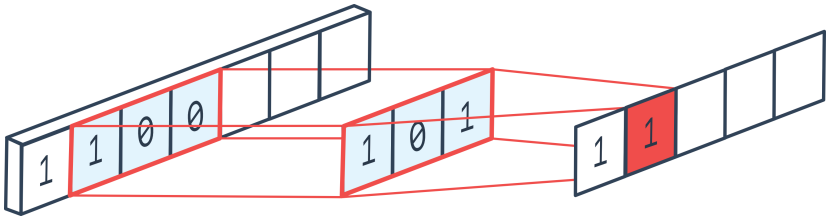

Figure 2.5: Figure showing 1D convolution. This type of convolution is often applied on time series data or audio signals. [30]

Lastly, 3D convolution is commonly employed in analyzing 3D image data. This convolution type is highly relevant in healthcare related purposes. Particularly for Magnetic Resonance Imaging (MRI) data, which is widely used in examining the brain and internal organs, or Computerized Tomography (CT) Scans, which combines X-ray images taken from various angles to create a 3D representation of the body. Similar to 1D and 2D convolution, 3D convolution can be utilized to classify this medical data or extract features from it. Additionally, given that video is a sequence of image frames, it possesses spatial features that can also be analyzed by applying 3D convolution [52].

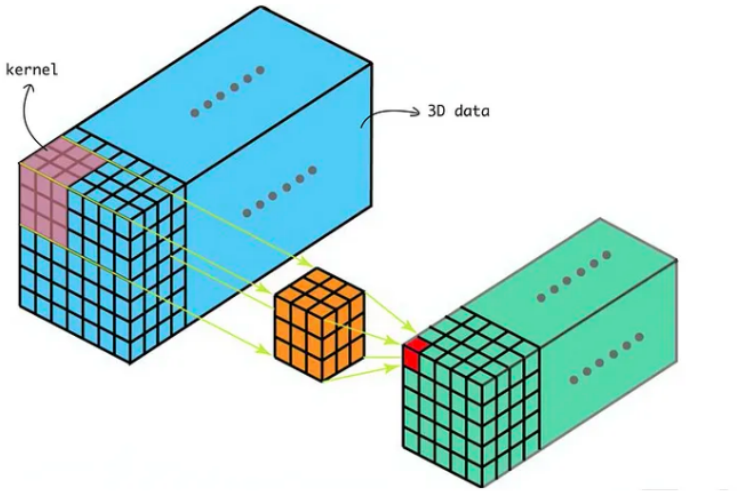

Figure 2.6: Figure showing 3D convolution. This type of convolution is often applied on data from imaging techniques such as MRI, or CT scans. [52]

Pooling

Pooling is a technique commonly used in CNN´s to reduce the dimensionality of the input data by summarizing a large set of values into a smaller set of representative values. The operation is quite similar to convolution, in that it uses a kernel of a given size to traverse the input data. The goal of pooling is to make the representations of the input more compact and manageable, thus saving computational resources while preserving the important information in the data [40].

The most common type of pooling is Max pooling, where the maximum value within each region is selected as the output value. Average pooling on the other hand extracts the average value within each region as the output value [13]. The difference in the

two methods of pooling is visualized in Figure 2.7.

It is worth mentioning that Max Pooling often is preferred due to its ability to act as a Noise Suppressant, meaning that it reduces unwanted noise in the data [40].

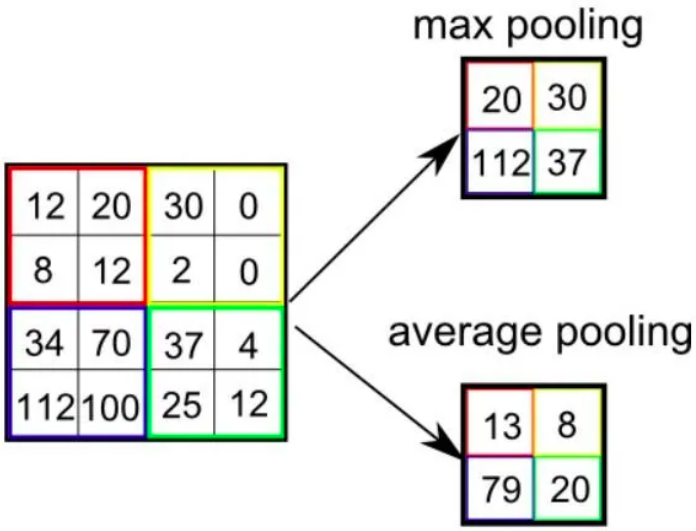

Figure 2.7: Figure showing the effect of the two main Pooling methods, on a given feature map. [40]

A convolutional layer is often followed by a pooling layer. Together, the convolutional and pooling layers make up a so-called 'convolutional block' [13].

Dropout

Dropout is a regularization technique often used in CNN´s to prevent overfitting of the network during training. Overfitting is the phenomenon in which the model learns features that are part of the statistical noise present in the data set. The result is that the model may perform very well on the training data but poorly on new data [56].

The idea behind dropout is to drop out or "turn off" some of the neurons in a layer of the neural network during each training iteration. During each iteration, the dropout technique sets the outputs of some neurons in the network to zero. This means that these neurons are effectively "turned off" and their outputs are not used for that iteration of training. Figure 2.8 demonstrates this in effect. The dropout rate is a hyperparameter in the range of 0 to 1 that determines the probability of a neuron

being turned off [56].

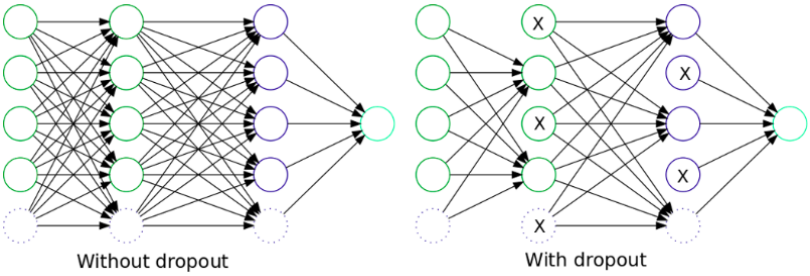

Figure 2.8: Figure showing the effect of dropout on fully connected layers. [48]

By randomly dropping out neurons, dropout prevents the neural network from relying too heavily on any one neuron, forcing the network to learn more robust and generalized features. Dropout is a simple and effective technique for regularization, and it has been shown to improve the performance of neural networks on a wide range of tasks [56].

Batch Normalization

Similar to dropout, batch normalization is a technique often used in CNN´s to improve the training and stabilize the network. It is a normalization technique that normalizes the input data to each layer of the neural network during training by adjusting and scaling the activations [15].

The idea behind batch normalization is to improve the stability of the neural network by normalizing the inputs to each layer, thereby reducing the internal covariance shift. Internal covariance shift is a phenomenon in which the distribution of activations in a layer changes as the parameters of the previous layers are updated during training. This can make training slow and difficult, as the network has to keep adapting to the changing distribution of inputs. Batch normalization solves this problem by normalizing the inputs to each layer [15].

Batch normalization has been shown to have several benefits for neural networks, including faster training, better accuracy, and improved generalization. It also reduces the sensitivity of the model to the initial values of the parameters, which can help avoid overfitting and improve the robustness of the model [15].

Hidden dimensions

As seen in Figure 2.3, in the fully connected layers each neuron is connected to every neuron in the previous layer. Each connection has an associated weight and bias, which are learned during training. The weights and biases determine the transformation that is applied to the data at each layer.

The number of features that are used in a neural network layer is referred to as "hidden dimensions". The features in question are often abstract and not directly observable, hence the term "hidden".

It is the number of hidden dimensions in a layer that determines how complex the transformation can be. In other words, a layer with more hidden dimensions has the capability of learning more complex features, while at the same time requiring more training data and computational resources. Too few hidden dimensions can result in underfitting, where the model is not able to capture all of the patterns in the data, while too many hidden dimensions can result in overfitting, where the model learns the noise in the data instead of the underlying patterns.

Pipeline

A DL pipeline is a framework that enables developers to create, test, and optimize DL models for various applications. It is a sequence of interconnected components that are designed to perform specific tasks related to DL, such as data collection, preprocessing, feature extraction, model training, and evaluation. The pipeline may also include components for post-processing and deployment of the trained models [50].

The primary objective of this pipeline is to simplify the development process of DL models and reduce the time and effort required for each step. The pipeline achieves this through providing an end-to-end solution that allows developers to focus on specific tasks rather than managing the entire process. Moreover, a DL pipeline can also help automate repetitive tasks, reduce human error, and improve the overall efficiency of the development process [43].

Typically, a pipeline consists of five stages that are arranged in a logical sequence. The

first stage is data collection, where the pipeline collects the necessary data required for the specific DL task. The data may be obtained from sources such as published data sets. The second stage is data preprocessing, which involves cleaning, normalizing, and transforming the collected data to prepare it for training the DL models. This stage is critical because the quality and accuracy of the trained models depend heavily on the quality of the input data.

The third stage is feature extraction, which involves extracting relevant features from the preprocessed data. The fourth stage is model training, where the models are trained using the extracted features and optimized using various techniques. This stage is often the most time-consuming and resource-intensive, as it requires significant computing power to train complex DL models [59].

The fifth stage is model evaluation, where the trained models are evaluated using a test data set to determine their accuracy and performance. This stage is crucial for identifying any issues with the trained models and fine-tuning them for better accuracy.

The final stage is model deployment, where the trained models are deployed into production environments and integrated into the end application. This stage requires the deployment of the developed models onto cloud or edge devices, depending on the application's requirements. Figure 2.9 demonstrates the structure and flow of a typical DL pipeline.

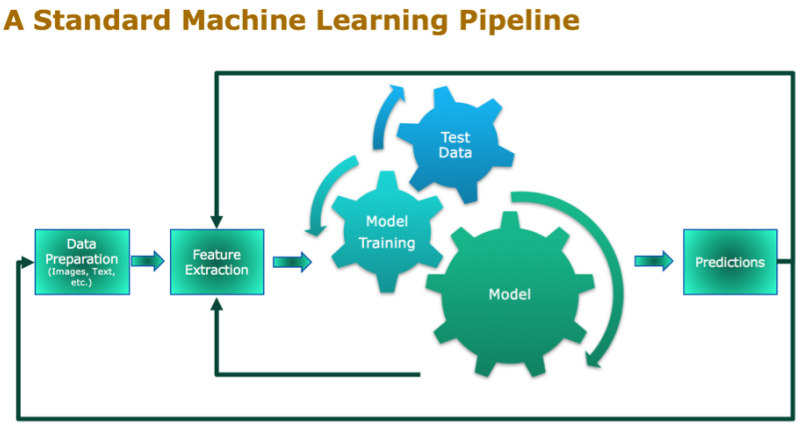

Figure 2.9: Figure showing a traditional DL pipeline. [59]

In summary, a DL pipeline is a comprehensive framework that provides developers with

the tools and resources to build, train, evaluate, and deploy DL models. It is a crucial component in the development of modern DL applications and enables developers to create highly accurate and efficient DL models for a wide range of use cases.

2.1.3 Alternative model-types

Recurrent Neural Network:

A Recurrent Neural Network (RNN) is a type of artificial neural network that has the ability to process sequential data. The network uses feedback connections that allow the previous outputs to be used as inputs in subsequent time steps. This helps the network to preserve information over long sequences and make predictions based on historical data [33]. RNNs are widely used in natural language processing and speech recognition. They can also be applied in areas like stock prediction, speech recognition, and machine translation [7] [29].

CNN-LSTM:

A CNN-LSTM is a type of DL network that combines CNN´s and Long Short-Term Memory (LSTM) networks. The CNN component is responsible for extracting features from input data, while the LSTM component processes the sequential information. The combination of the two allows the model to make predictions based on both the spatial and temporal aspects of the data. CNN-LSTM networks are commonly used in a wide range of applications, such as natural language processing, activity recognition, and video description [8].

Stacked Auto Encoders:

Stacked Autoencoders (SAE), is a type of neural network architecture that is commonly used for unsupervised learning tasks, such as dimensionality reduction and feature learning. It consists of multiple interconnected autoencoders, which are neural networks that aim to reconstruct their input, that are stacked on top of each other. The goal of an SAE is to learn a hierarchical representation of the input data, where each layer focuses on learning increasingly complex features. SAEs can be trained in an

unsupervised manner, which means that they do not need labeled data to learn useful representations. Once the SAE is trained, it can be used as a feature extractor for other supervised learning tasks [6].

Deep Belief Network:

A Deep Belief Network (DBN) is a type of DL algorithm that is based on unsupervised learning. It is composed of multiple layers of interconnected nodes, with the first layer being a Restricted Boltzmann Machine (RBM), which acts as a feature extractor. The subsequent layers are fully connected neural networks, and they are trained to construct a generative model of the input data. The goal of a DBN is to learn a compact representation of the input data, which can be used for tasks such as classification or data reconstruction. DBNs have been successfully used in a variety of applications, including image and speech -recognition [23].

2.1.4 Explainable Artificial Intelligence:

Explainable Artificial Intelligence (XAI) refers to an approach in which providing an explanation of how an AI model arrived at its prediction is key. In other words, XAI aims to assist the user in understanding the inner workings of a given DL model. Considering the use of DL models for medical image classification, XAI can explain what connections the model is making and highlight which parts of the image are most influential in prediction [25].

2.2 Regulations

2.2.1 Right to explanation

European Union's General Data Protection Regulation (GDPR)

According to Article 13 and 14 (on the right to information) and Article 15 (on the right to access), the controller is required to provide information on "the existence of automated decision-making, including profiling, referred to in Article 22(1)" and

"meaningful information about the logic involved, as well as the significance and the envisaged consequences of such processing for the data subject" [42].

Moreover, the European Data Protection Board recommends that controllers should provide the data subject with the "rationale behind or the criteria relied on in reaching the decision." The provided information should be detailed enough to allow data subjects to comprehend the reasons for the decision [42].

In light of these articles and recommendations, it is evident that a legal precedent exists that requires the right to explanation [16] [4] [**explanationwebsite**].

U.S. Food and Drugs Administration (FDA) Action Plan

In the "Artificial Intelligence and Machine Learning (AI/ML) Software as a Medical Device Action Plan" published by the FDA in 2021, they highlight the significance of interpretability within a collection of terms for AI implementation. [**fda**] [26].

2.2.2 Protection of sensitive health data

European Union's General Data Protection Regulation (GDPR)

In GDPR, the regulation of health data is covered under Article 9, which is titled "Processing of special categories of personal data". This article sets out the conditions under which processing of sensitive data, including health data, is allowed. Specifically, it states that processing of health data is only allowed under certain circumstances, such as with explicit consent from the data subject or if processing is necessary for specific purposes such as public health or medical research. The article also requires appropriate safeguards to be in place to protect the confidentiality and security of the health data being processed.

U.S. Health Insurance Portability and Accountability Act (HIPAA)

The U.S. HIPAA elaborates rules requiring, among other things, the formulation of policies and the setup of training systems for those who have access to sensitive data

[37]. Additionally, the transfer of health data in the United States is regulated through HIPAA [46].

2.3 Literature review

To gain a comprehensive understanding of the field of DL in healthcare, it is necessary to have an overview of the literature regarding current state-of-the-art solutions. This includes the preferred architecture of the DL models as well as the methods used to explain the outputs of the models.

2.3.1 Application and choice of models

Deep learning for ECG data

Hong et al. [18] offers a comprehensive overview of the use of ECG in various healthcare related purposes. The article, published in the journal "Computers in Biology and Medicine" in 2020, was co-written by researchers from the US and China [18].

To gather information, they extracted and analyzed 191 articles published between 2010 and 2020, that applied DL models to ECG data. A more detailed overview of their strategy is shown in Figure 2.10. Their findings demonstrated that DL architectures have been employed for various ECG analytics tasks, such as disease detection, localization, and biometric identification.

The results indicated that the most commonly applied choice of model was a CNN. Moreover, the authors noted that a hybrid architecture of a CNN and Recurrent Neural Network (LSTM-CNN) using expert features was found to yield the best results. Additionally, they highlighted the superior performance and fast computation of a CNN as a key advantage [18].

In the end they concluded that the use of DL in ECG data has grown significantly in recent years, with accuracy comparable to traditional approaches and even better results possible through ensemble methods.

However, the authors also recognized challenges in interpretability, scalability, and

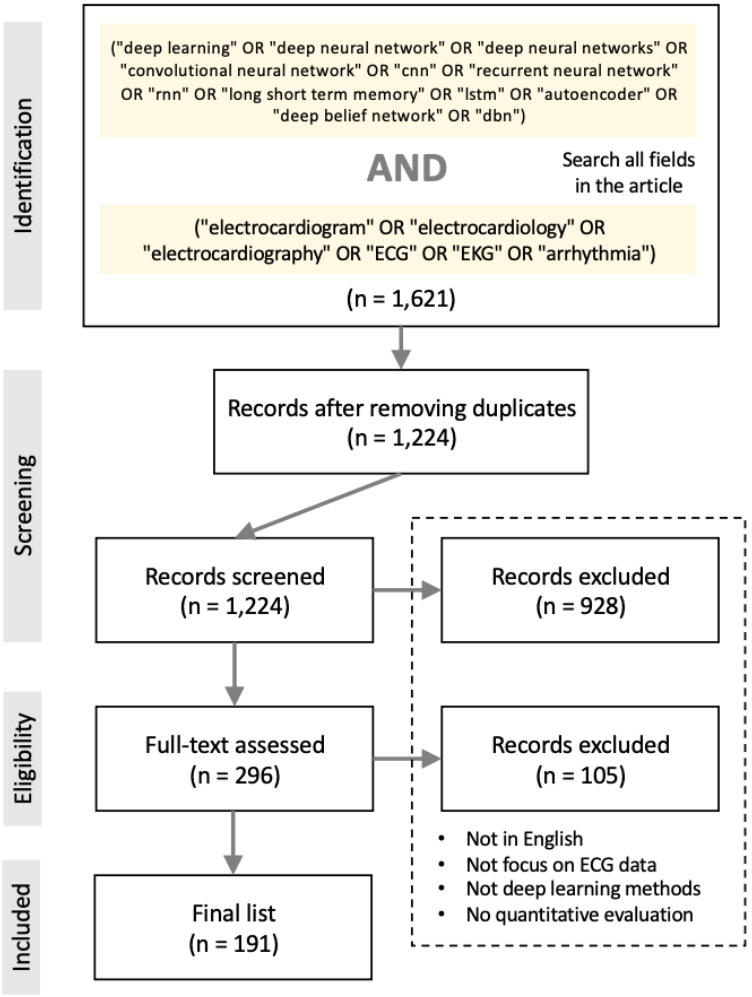

Figure 2.10: Figure showing the strategy of the selection. [18]

efficiency that need to be addressed. In regards to interpretability the researchers noted that "This challenge is much more severe in the medical domain because diagnoses without any explanation are not acceptable for medical experts." [18].

Overall, the article reflected a positive outlook on the use of DL analysis on ECG data, and the authors found DL techniques to be a promising solution for predictive healthcare tasks.

Deep learning in ECG diagnosis

Next, Liu et al. [24] provides a more detailed examination of the use of DL in ECG analysis for the specific "use-case" of diagnosis. Written by researchers from the University of Electronic Science and Technology of China and published in the journal

"Knowledge-Based Systems," the study builds upon the positive sentiment expressed in the previous review study [24].

The authors remarked that; "Deep learning shows outstanding performance on ECG classification studies in the recent few years ... Latest studies can achieve higher accuracy and efficiency than manual classification by experts" [24].

In the review the researchers categorized studies according to 4 classic DL architectures: SAE´s utilized in 8 studies, DBN´s which was used in 6 instances, CNN´s which were used in 19 studies, and RNN´s that were used in 14 cases. Moreover, they commented on the most prevalent model stating "CNN is widely applied in ECG diagnosis tasks in recent few years and outstanding performance has been achieved." [24]

In this context it is worth noting that the authors included CNN-LSTM models under the classification of RNNs as seen in Table 2.1. Further highlighting the feasibility and popularity of this particular hybrid model.

As in the previous review the authors expressed that in spite of the rapid development, limitation and open issues for DL methods were still present. Specifically, one of the aspects outlined was visualization. In this regard they state that "Poor interpretability is a key issue of architecture ... This [interpretability] can be realized by mathematic justification and visualization." Based on the reoccurring sentiment in both articles visualization and interpretability is identified to be among the biggest challenges for wider adoption of DL models for ECG analysis.

Deep learning in disease detection

Lastly, Murat et al. [31] delves deeper into the detection of specific diseases; in this case arrhythmia. Arrhythmias are a significant type of heart condition, that may occur alone or combined with other heart diseases. Symptoms of arrhythmias include a slow, fast, or unpredictable heartbeat, which may lead to high mortality rates in heart patients. Therefore, timely and accurate identification of arrhythmias is crucial for patient care [31].

In the article the authors reviewed and discussed peer-reviewed journal articles that uti-

Table 2.1: Models based on recurrent neural network, including CNN-LSTM hybrid networks. Recreated from article [24]

Application	**DL algorithm**	**Database**	**Result**
Arrhythmia classification	LSTM	MIT-BIH Arrhythmia Database	Accuracy 99.39%
Coronary artery classification	LSTM, CNN	Fantasia; St.-Petersburg INCART 12-lead arrhythmia	Accuracy 99.85%
Interpatient arrhythmia classification	GRU, CNN	MIT-BIH Arrhythmia Database	F1 score 61.25 for SVEB, 89.75 for VEB
Atrial fibrillation detection	LSTM	MIT-BIH Atrial Fibrillation Database	Accuracy 99.77% with blindfold validation
Atrial fibrillation detection	LSTM	MIT-BIH Atrial Fibrillation Database	Accuracy 99.77% with blindfold validation
Arrhythmia classification	LSTM, CNN	MIT-BIH Arrhythmia Database	Accuracy 98.10% Sensitivity 97.50% Specificity 98.70%
Heartbeat classification	LSTM	MIT-BIH Arrhythmia Database;St.-Petersburg INCART 12-lead arrhythmia; MIT-BIH SVDB Database	Accuracy 99.9% Sensitivity 99.8%% Specificity 99.9%
Atrial fibrillation detection and monitoring	LSTM, CNN	MIT-BIH Arrhythmia Database; MIT-BIH AF Database; MIT-BIH NSR Database	Accuracy 97.80% Sensitivity 98.98% Specificity 96.95%
Arrhythmia classification	LSTM, CNN Attention module	1st China Physiological Signal Challenge	PPV 82.6%, Recall 80.1%, accuracy 81.2%
Inter- and intra-patient heartbeat classification	LSTM-based auto-encoder; CNN	MIT-BIH Arrhythmia Database	Accuracy:99.53% for inter-patient, 99.92% for intra-patient
Arrhythmia classification	RNN	MIT-BIH Arrhythmia Database	Accuracy 99.3% for VEB; Accuracy 98.6% for SVEB
Arrhythmia classification	DELM-LRF-BLSTM	MIT-BIH Arrhythmia Database	Accuracy 99.32% Sensitivity 97.15%
Atrial fibrillation prediction	LSTM	Long-term AF Database; AF terminal challenge Database	Accuracy 92% 92% F-score
Atrial fibrillation	LSTM, CNN	Cardiology Challenge 2017 Dataset	sped up by 38% F1 score 89.55%
CAD, Myocardial infarction, congestive heart failure	LSTM, CNN	St.-Petersburg INCART 12-lead arrhythmia, PTB Database, BIDMC CHF Databases, Fantasia Databases	Accuracy 98.51% Sensitivity 97.89% Specificity 99.3% Positive predict 97.3%

lized DL for arrhythmia detection. Relevant information from the studies was inserted into a table as seen in Table 2.2. Thereafter an experimental study was conducted to

Table 2.2: Table showing a selection of state-of-the-art studies using DL on ECG data. Recreated from article. [31]

Database	**Number of Classes**	**Total Data**	**DL Technique**	**Results**
MIT-BIH Arrhythmia Database	5	83,648 beats	1-D CNN	VEB: Acc ¼ 99%, Sen ¼ 93.9%, Spec ¼ 98.9% SVEB: Acc ¼ 97.6%, Sen ¼ 60.3%, Spec ¼ 99.2%
Zio Patch	14 rhythm	64,121 records	34-layer CNN	PPV ¼ 0.809, Recall ¼ 0.827, F1 ¼ 0.809
MIT-BIH Arrhytmia Database + Synthetic data	5	109,449 beats	9-layer CNN	Set A: Acc ¼ 93.47%, Sen ¼ 96.01%, Spec ¼ 91.64% Set B: Acc ¼ 94.03%, Sen ¼ 96.71%, Spec ¼ 91.54%
PhysioNet Challenge 2017	4	8528 records	LSTM	10-folds CV: F1 ¼ 83.10% Entry: F1 ¼ 84%
MIT-BIH Atrial Fibrillation Database	2	100 beat window 99	LSTM	CV: Acc ¼ 98.51%, Sen ¼ 98.32%, Spec ¼ 98.67% Blind fold validation: Acc ¼ 99.77%, Sen ¼ 99.87%
MIT-BIH Arrhytmia Database	5	2520 segments (10 s)	2-D Deep CNN	Acc ¼ 99.0%
MIT-BIH Arrhytmia Database	5	16,499 beats with variable length	CNN-LSTM	Acc = 98.1%, Sen = 97.5% Spec = 98.7%
MIT-BIH Arrhytmia Database	13 15 17	833 fragments (10s) 976 fragments (10s) 1000 segment (10s)	1D-CNN	Acc = 95.20% Acc = 92.51% Acc = 91.33%

provide more insight into techniques that make DL effective for arrhythmia detection. Based on their examination of the selected articles the researchers found it evident that CNN models were the preferred alternative compared to other models [31].

In the discussion section the researchers add weight to the statements of the previous authors by noting that "hybrid models such as CNN-LSTM tend to produce successful results." [31]. Interestingly the authors also addressed some of the potential downsides of LSTM models such as the high resource utilization, and the need for a large data set. The authors recommended several techniques to combat this issue namely transfer learning, residual connections and data augmentation.

Similar to the previous review articles, the importance of interpretability was underlined. The authors stated in the conclusion; "Finally, what features are taken into account during the diagnostic process, due to the black-box nature of deep learning methods, is an important question mark." [31].

2.3.2 Data sets

Synthetic data set

Thambawita et al. [46] attempts to tackle the issue of privacy issues in data sets in an innovative way. The research was published in the journal Scientific Reports in 2021 and is centered around utilizing DL models for the generation of synthetic ECG data.

Specifically, 2 General Adversarial Networks (GAN) were implemented in order to generate 10-s 12 lead ECGs. Inspired by the ability of a specific model named WaveGAN to produce audio signals the researchers initially implemented a similar structure for their model. Thereafter, the researchers also implemented a novel generative model called Pulse2Pulse for the production of ECG signals.

The quality of the generated data was determined to be satisfactory given that 150 000 ECG samples were successfully uploaded to the commercial MUSE 12SL system (ECG system widely used in hospitals), with no samples being rejected as invalid by the system. The two DL models were subsequently reviewed and compared to each other. Their comparison indicated that both in terms of training time, and quality of the generated ECG data the Pulse2Pulse model performed better then the WaveGAN implementation.

The researchers discussed several notable advantages of the use of synthetic data for ECG analysis. Most recognized was the principle that the data was not tied to any one individual or group of people. Thus, the use of the data was not affected by ethical considerations related to privacy.

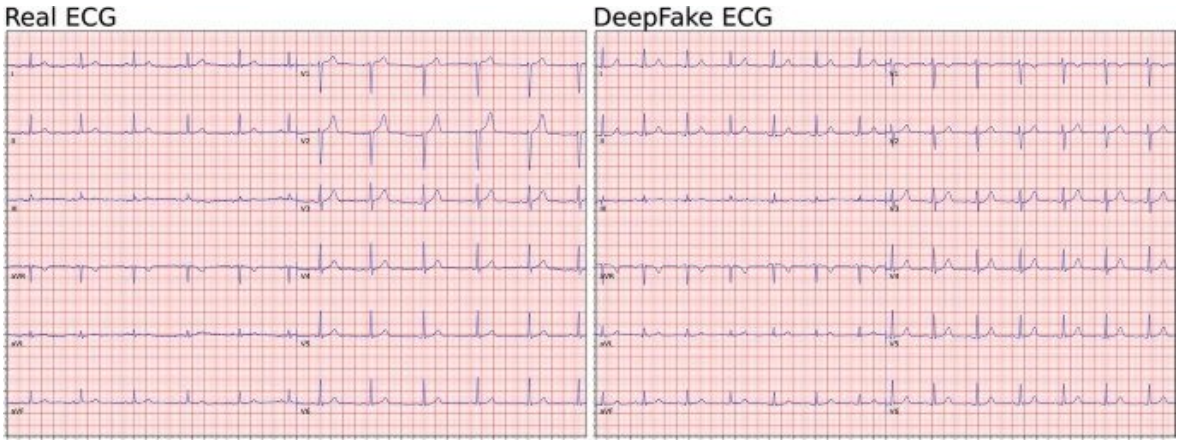

Figure 2.11: Figure showing the comparison of real ECG data and synthetic ECG data generated by a DL model. [46]

PTB-XL

In 2020 researchers from the national Metrology Institute of Germany (Physikalisch-Technische Bundesanstalt) presented a publicly available ECG data set suitable for the training of DL models. The data set named PTB-XL is to-date the largest accessible 10-s 12 lead ECG data set. It is compromised of a total of 21837 records from 18885 patients, and covers a range of diagnostic super- and sub classes. Furthermore, the data set includes an equal representation of both sexes, with males accounting for 52% and females for 48%, and covers all ages from 0 to 95 years [54].

The raw signal data was recorded by commercial ECG systems from the company Schiller AG between October 1989 and June 1996. The set of annotations were divided into 71 different statements, with multiple statements possible for an individual ECG. Two cardiologists were engaged with the task of selecting and verifying the annotation for each record.

In the PTB-XL data set the 5 diagnostic super-classes are represented by the following categories;

- NORM: Refers to a Normal ECG signal.
- MI: Myocardial infarction (MI) occurs when blood flow to the myocardium (heart muscle) decreases or stops completely. Commonly known as a heart attack MI

may cause permanent heart damage and death [11].

- ST/T Change: ST/T changes refers to the alteration of ST and T waves, and may indicate a cardiac disorder or a normal variation. Therefore, the correct interpretation of these changes relies on the context of the patient's clinical condition and whether similar findings have appeared in previous ECGs [38].

- CD: Conduction Disturbance (CD) refers to a disruption in the way the electrical signals move through the heart. When certain conduction disorders occur, they may lead to arrhythmias, or irregular heart beats [3].

- HYP: Hypertrophy is a condition characterized by the abnormal thickening of the heart muscle, and is also known as hypertrophic cardiomyopathy (HCM). A thickened (hypertrohied) heart muscle may not be able to effectively pump blood [12].

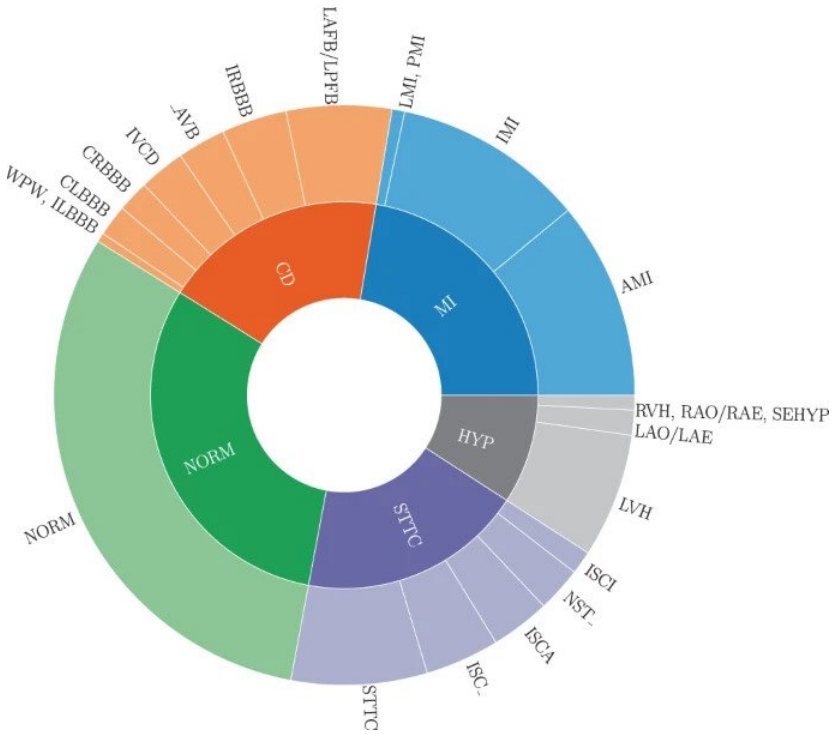

Figure 2.12: Figure showing the distribution of the PTB data set. The diagnostic superclasses; NORM = Normal, MI = Myocardial Infarction, CD = Conduction Disturbance, STTC = ST/T-Change and HYP = Hypertrophy all contain various diaganostic subclasses. [54]

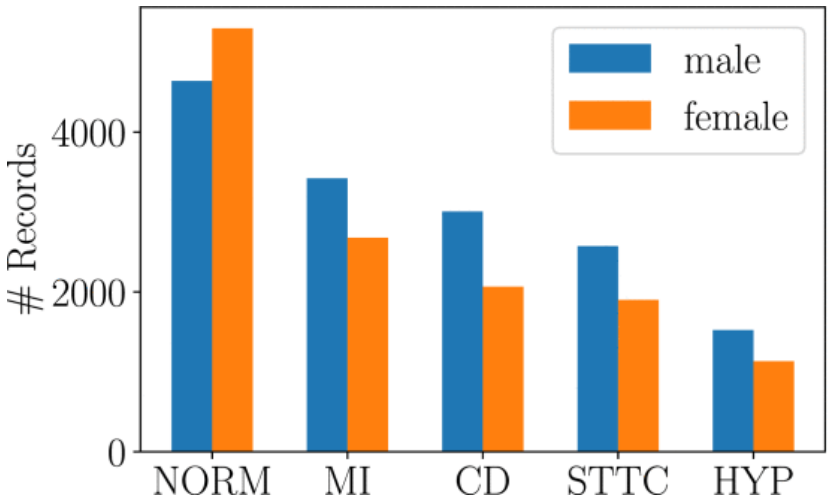

Figure 2.13: Figure showing the distribution of the different diagnostic super-classes in terms of male and female patients. [54]

The researchers commented on several of the advantages of the data set, stating; "the data set is distinguished by its diversity, both in terms of signal quality, but also in terms of a rich coverage of pathologies ... and a large proportion of healthy control samples that is rarely found in clinical data sets." [54].

Strodthoff et al. [44] presents the first bench-marking results for the PTB-XL data set. The article is written by several of the same authors behind the publication of the original data set.

In their approach they utilized the ICBEB2018 data set for comparison. Similar to the PTB-XL data set the ICBEB2018 data set is a data set of considerable size, containing 6877 12-lead ECGs lasting between 6 and 60 seconds. The data set was released for the 1st China Physiological Signal Challenge 2018 held during the 7th International Conference on Biomedical Engineering and Biotechnology (ICBEB 2018). Specifically, in the study models trained on the PTB-XL data set were compared to models trained on the ICBEB2018 data set [44].

A variety of tasks suitable for bench-marking were conducted such as inferring ECG statements related to the rhythm of the ECG signal (multilabel classification), inferring ECG statements related to the form of the signal (multilabel classification), inferring a subject's sex (binary classification) and inferring a subject's age (regression) [44].

Overall, the results showed that the performance of the models trained on the respective data sets was consistent. The authors brought forth some insights. Firstly, after analyzing various DL-based timeseries classification algorithms, they noted that CNN´s demonstrated the highest level of performance on all tasks. Moreover, their study indicated that PTB-XL had superior performance when used to pre-train models aimed at analyzing smaller data sets as seen in Figure 2.14. Thereby, making it a powerful resource for pre-training a model that may later be finetuned on a smaller data set [44].

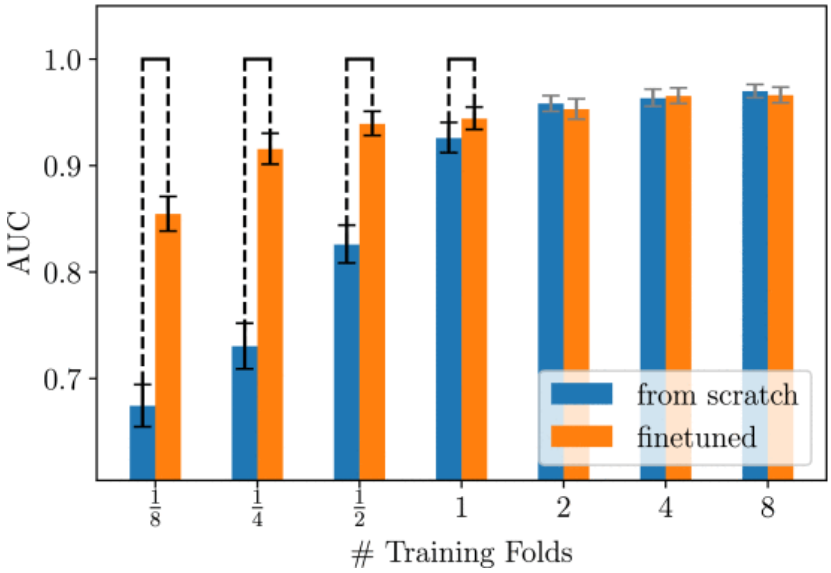

Figure 2.14: Figure showing performance of models pre-trained on the PTB-XL data set on the ICBEB2018 data set. The results showed statistical significance when decreasing the size of the ICBEB2018 training set.[44]

Lastly they observed, that the use of ensemble models only lead to slight performance increases, and that with the exception of of super-diagnostic classification this performance increase was not statistically significant.

2.3.3 Explainable methods

Loh et al. [25] provides an in-depth look at the explainable AI methods used in healthcare between 2011 and 2022. In terms of ECG analysis, the article found that methods such as SHapley Additive exPlanations (SHAP), Gradient-weighted Class Activation Mapping (GradCAM) and Local Interpretable Model-agnostic Explanations (LIME)

were commonly used. SHAP and GradCAM were the most popular, whereas only one article utilizing LIME was found [25]. The research article using LIME studied the use of a wearable bio-signal detector, and although ECG data was used as input it did not aim to provide insights into ECG-specific problems [49].

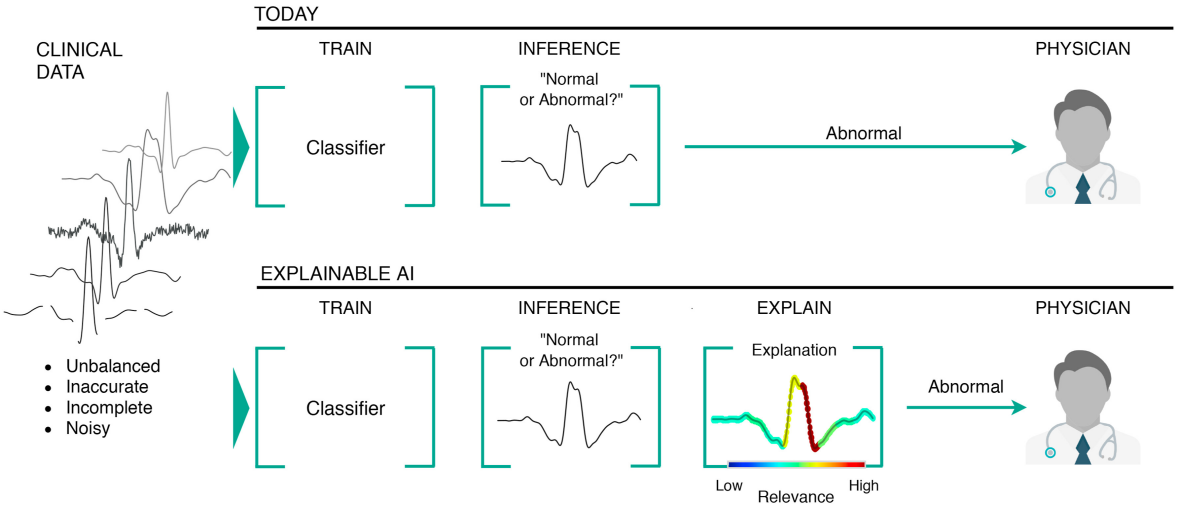

Figure 2.15: Figure comparing the pipeline of traditional DL models to DL models that apply explainable methods. [34]

Looking at the aforementioned review article, in the case of ECG analysis, one study was found to utilize SHAP, while 3 were found to use GradCAM.

LIME

Neves et al. [34] presented research on the technical feasibility and practical usefulness of 3 types of visual explanations. Specifically, they applied and compared SHAP, LIME, and a novel method called Permutation Sample Importance (PSI).

The authors commented on their observations; "Both PSI and LIME are adequate methods to explain a time series classifier by measuring the relevance of each sample for the classification. These findings have a broad impact with regards to the applicability of such methods in real-world practice." [34].

Furthermore, an informative user study was also conducted to evaluate the potential of the visual explanations on ECGs as shown in Figure 2.16. The researchers found that "the explanations provided by PSI and LIME were more sensitive to the temporal

ordering when the derivative is also considered." They also noted that SHAP's performance across all methods was lower than PSI and LIME. Additionally, the study found that the inclusion of the derivative improved the explanations for CNN predictions in terms of temporal dependency. [34].

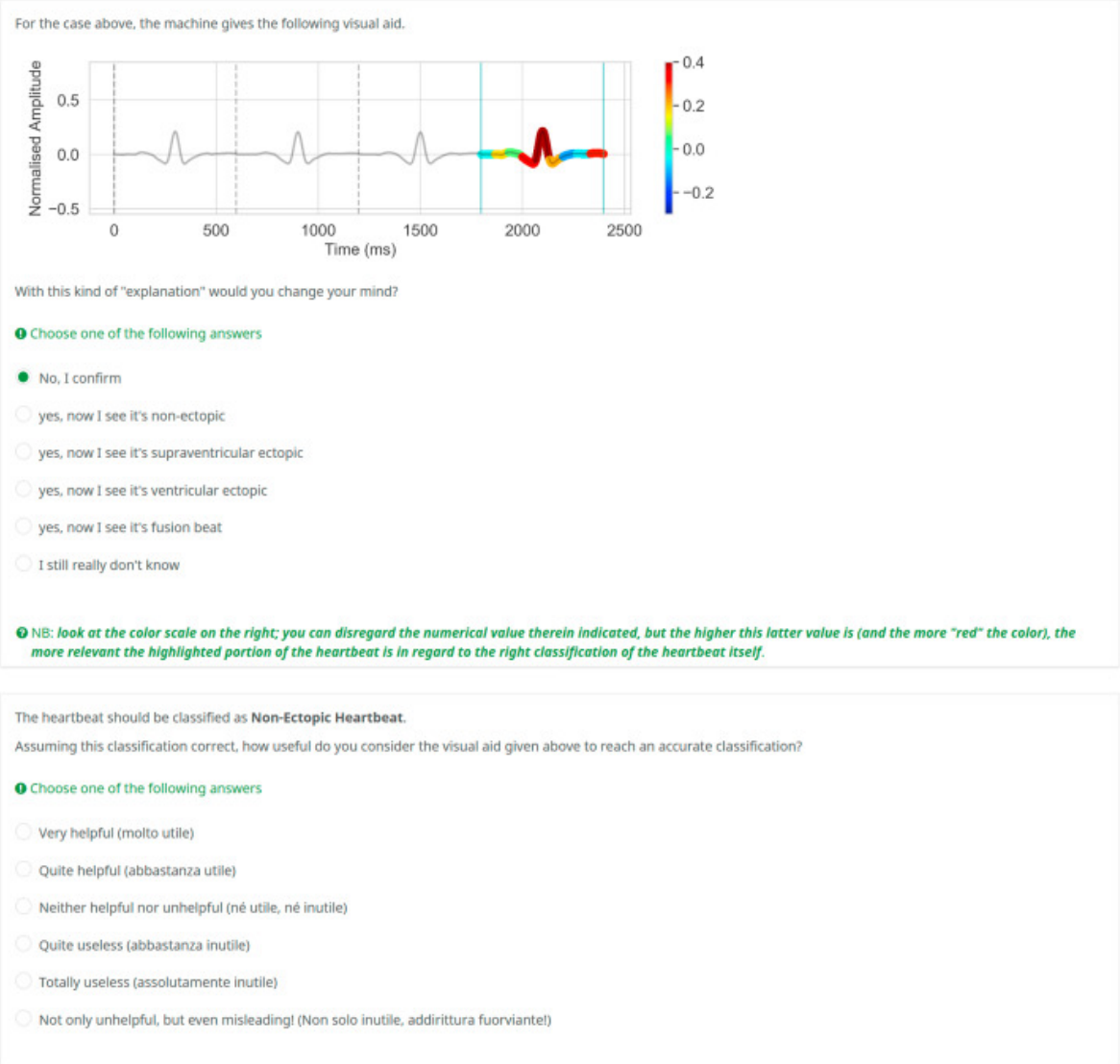

Figure 2.16: Figure showing an example-question from the conducted user study. [34]

In the end they proposed using the time series derivative to develop XAI methods for the measurement of feature importance in the temporal domain.

GradCAM

The GradCam method is originally a technique for visualizing the parts of an image that a CNN uses to make a classification decision. GradCam stands for Gradient-

weighted Class Activation Mapping. In this method, the CNN's output is used to calculate the gradients of the last convolutional layer of the network. These gradients are then used to weight the feature maps in that layer, which results in a heat map that shows which parts of the image are most important for the classification decision as seen in Figure 2.17.

Figure 2.17: Figure showing application of GradCAM. The segments of the image that the model used to make the prediction are highlighted. From a DL model used to classify images of cats and dogs. [39]

Studies implementing GradCAM

Considering implementations of the GradCAM method, one notable article, was written through collaboration of Norwegian and Danish researchers in 2021. The article titled; "Explaining deep neural networks for knowledge discovery in electrocardiogram analysis" was published in the journal "Scientific Reports", and included a novel approach built on the GradCAM method.

The authors explained how an implementation of attention maps could provide meaningful and detailed visualizations. Thus, the paper attempted to tackle a key point discussed in previous articles, namely the importance of visualization. Specifically, they highlighted prevention of fatal mistakes, identification of novel features, and improved ability to place legal responsibility in the event of mistakes as key advantages [17].

The researchers went into further detail on their approach clarifying how a modification of the traditional GradCAM method helped provide an accurate representation of

what regions of the ECG were most decisive for their CNN model. In this modified version called ECGradCAM visualizations were generated for each lead of the ECG, and thereafter the average values across all leads were used to produce the final attention maps [17].

Specifically, a heat map was produced in which the most important areas were marked as hot (red color), and the less important regions were marked as cold (blue color). In this context importance can be understood as the weight a specific area contributed to the overall prediction. Figure 2.18 shows the output of their implementation on ECG data.

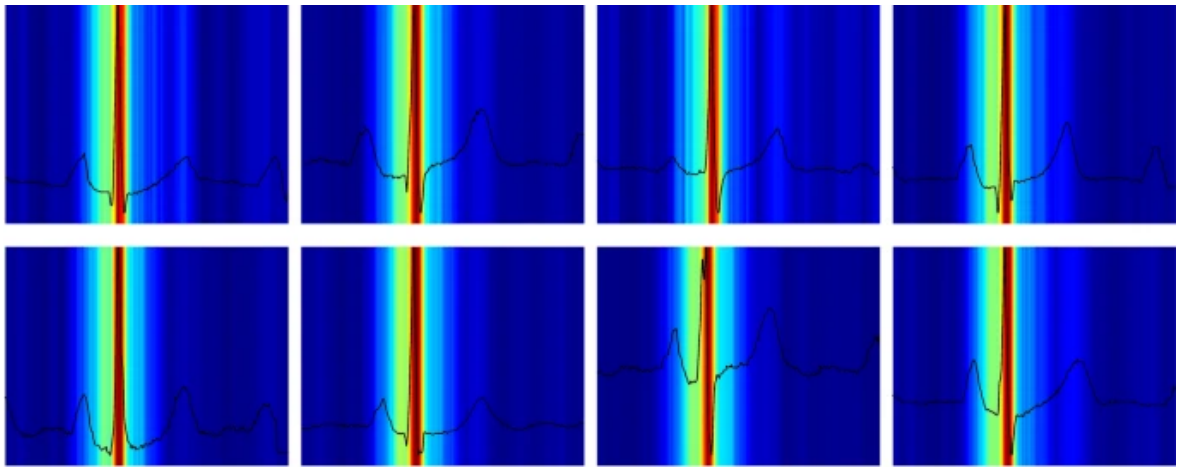

Figure 2.18: Figure showing the related attention map for sex prediction. The researchers noted that the QRS complex was of high importance for the model. [17]

In order to evaluate the work 2 cardiologists were then incorporated in the study, and tasked with manually annotating a set of 20 randomly selected ECGs. Compared to the outputs from the model the cardiologists scored significantly lower in terms of precision and consistency. Furthermore, using findings from the attention maps the researchers were able to discover novel medical knowledge related to sex prediction. Thus, the feasibility and potential of the explainable method was further substantiated [17].

In line with the previous article, Jahmunah et al. [20] also employed the GradCAM method to provide insight to the decisions of their DL models. However, as seen in Figure 2.19 when compared to the ECGradCAM technique their version contained

certain differences.

In the paper a DL method was developed for detection of Myocardial Infarction (MI). MI accounts for the most deaths globally in terms of cardiovascular diseases, and thus accurate and timely diagnosis is considered to be crucial in order to ensure successful intervention.

To that end the researchers cited their motivation to be a result of the lack of literature regarding explainable models for MI detection. Consistent with the majority of the previously cited articles they gave thought to the shortcomings of non explainable methods, commenting; "The lack of explanation of the mechanisms of these models also poses a challenge as clinicians lose confidence in using deep models in clinical settings to aid in diagnostic decisions." [20].

During the course of the study, DenseNet and CNN models were utilized for the classification of both healthy subjects and patients with 10 classes of MI based on the location of myocardial involvement. After pre-processing, the R peaks of individual lead signals of 12-lead ECGs were detected to extract the beats (each beat was composed of sampled data from all 12 leads), and subsequently used as input to the DL models [20].

The Grad-CAM technique was thereafter applied to the outputs of both models to clarify the decisions made by the respective models. In this respect the authors comment; "The specific ECG leads and portions of the ECG waves most influential for the detection of each MI and healthy class, were marked. Overall, Lead V4 was the most activated lead with the most influence on the classification in both DenseNet and CNN models." [20].

Based on the results the authors concluded that "developed models combined with Grad-CAM are more likely to garner clinical acceptance and can be used to triage MI in hospitals and remote out-of-hospital settings." [20].

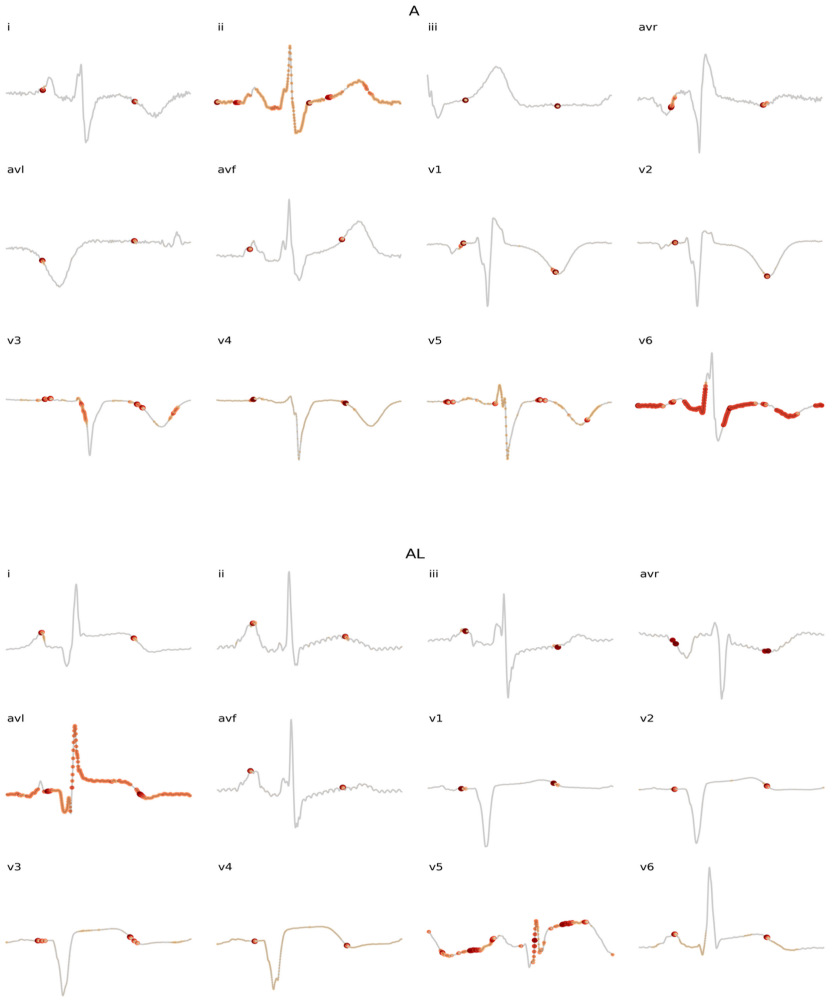

Figure 2.19: Figure showing how the GradCAM technique used in Jahmunah et al. [20] consists of certain variations compared to the ECGradCam method, and omits the use of blue and red colours in favour of dots. [20]

SHAP

The SHAP (SHapley Additive exPlanations) method is a technique for explaining the predictions of machine learning models. It provides a way to assign importance values to the input features of a model, indicating how much each feature contributes to the final prediction. The SHAP method is based on the concept of Shapley values from cooperative game theory. More precisely, considering a coooperative game with the same number of players as the number of features. SHAP will disclose the individual contribution of each player (or feature) on the output of the model, for each observation [47]. In other words, it works by calculating the contribution of each feature to the prediction for every possible subset of features. These contributions are then combined

to obtain the final importance values as seen in Figure 2.20.

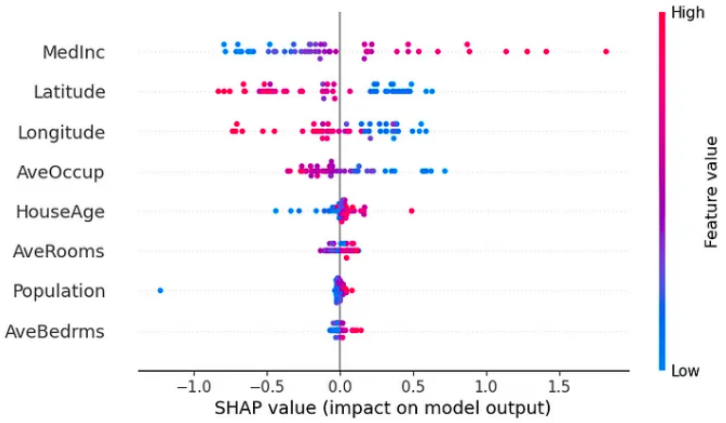

Figure 2.20: Figure showing application of SHAP. The contribution of different features to the prediction of the model are visualized. From a DL model used to predict prices on a data set of houses in California. [47]

Studies implementing SHAP

Another article also centered around the detection of MI named "Explainable Prediction of Acute Myocardial Infarction Using Machine Learning and Shapley Values" was published in 2020 in the journal IEEE. In this study the researchers opted to use SHAP as an explainable method for their model as opposed to GradCAM.

To conduct the study, 713,447 ECG measurements and related information regarding diagnoses, drug prescriptions, and selected laboratory test results were extracted from the Electrocardiogram Vigilance with Electronic data Warehouse (ECG-ViEW II) data set [19].

Thereafter 3 models were implemented; 2 DL models (CNN & RNN), and a decision-tree based model named XGBoost. All three models achieved a high prediction accuracy with the models scoring 89.9, 84.6 and 97.5 respectively. The researchers utilized Shapley values to identify the features that contributed the most to classification decision of the XGBoost model [19].

In line with the ECGradCAM method they utilized blue and red color schemes for visualization. Figure 2.21 demonstrates how features that contributed positively to the prediction (positive values) were marked with blue, while features that contributed

negatively to the prediction (negative values) were emphasized with red. However, in contrast to the GradCAM approach their application of SHAP did not involve the marking of an ECG complex.

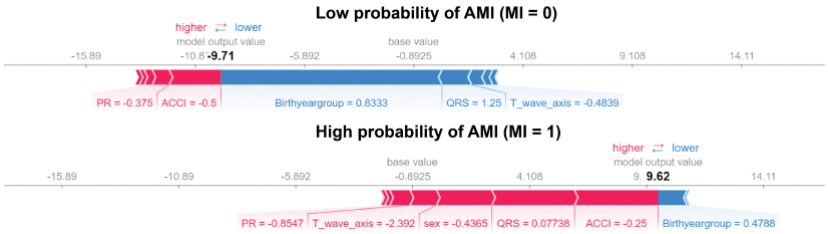

Figure 2.21: Figure showing the local explanation of 2 samples. One sample with MI and one sample without MI. [19]

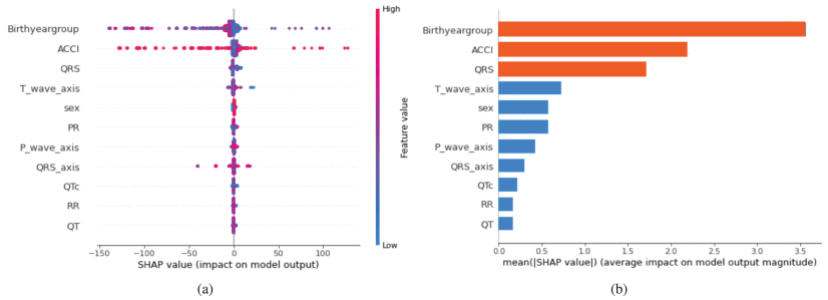

Figure 2.22: Figure showing (a) Local explanation summary and (b) Global feature importance. [19]

Based on Figure 2.22 Age (Birthyeargroup), ACCI, and QRS duration were observed to be the most important features for the prediction on the average in the whole testing data set. In light of the Shapley value analysis the researchers concluded that age, ACCI, and QRS duration were the most crucial variables in the prediction of the onset of acute MI. At the same time sex was found to be of less importance.

In light of their study they expressed that they found "Shapley analysis to be a promising technique to uncover the intricacies and mechanisms of the prediction model, leading to higher degree of interpretation and transparency." [19].

On the other hand, a different application of SHAP was employed in Anand et al. [2]. The research article titled "Explainable AI decision model for ECG data of cardiac disorders" was published in 2022, and in contrast to the previous study they focused on a general diagnosis of cardiac disorders.

Here the researchers implemented 8 different deep neural networks on the PTB-XL data set. After evaluating the different models the preferred structure was determined to be the ST-CNN-GAP-5 model. When comparing their results with the existing state-of-the-art results on the PTB-XL data set this model was found to be more effective [2]. The model was thereafter applied on a different data set consisting of patients with arrythmia in order to assess the generalizability. Here the results indicated the model was competitive in performance to the state-of-the-art models.

Finally the researchers applied SHAP to visualize the decisions of the model. In their approach they approximated a SHAP value for each input feature, and identified the top 500 SHAP values as the significant features that contribute to diagnosing a specific ECG record. They found that by using SHAP to interpret the DL model across various heart conditions, the same segments of ECG waves that would be analyzed by a trained cardiologist were highlighted [2].

Based on the top 500 SHAP values, the corresponding ECG wave/segments were highlighted in red color, while the features with lesser importance were highlighted in blue color as seen in Figure 2.23. Their approach bares resemblance to the various GradCam techniques applied on ECG DL models.

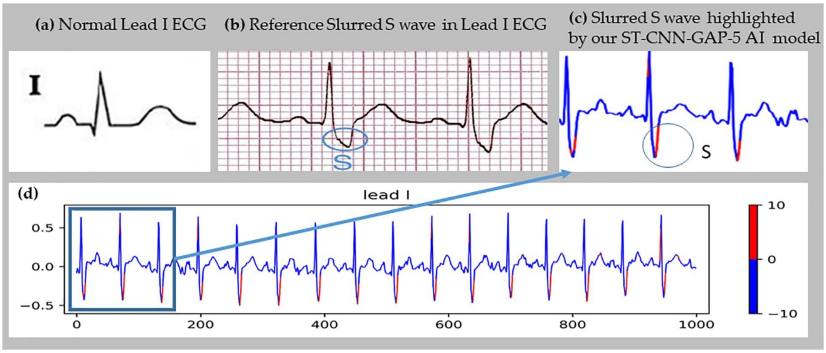

Figure 2.23: Figure showing the visualization of the SHAP approach. [2]

In conclusion the researchers were positive to the method commenting; "Results indicate that the model is able to highlight relevant alterations of the ECG waves as required by clinicians, making it explainable for diagnostic purposes." [2].

Alternative approaches

Aufiero et al. [4] also implemented GradCAM. Notably, in their research they first calculated the GradCam scores, and thereafter visualized the explanation in a unique way to the traditional GradCam methods used on ECG data.

The aim of their research was to successfully diagnose a rare heart disease called Congenital long QT syndrome (LQTS) using a DL model. This was determined to be especially important since most cardiologists are not experienced with patients carrying congenital LQTS and may not always recognize the accompanying ECG features. In addition, a proportion of disease carriers do not display obvious abnormalities on their ECG. Combined, this may cause under-diagnosing of a potentially life-threatening disease [4].

Using ECG data as input they implemented and trained a 1D CNN to classify genotype positive LQTS patients. The data was collected from a large 10-s 12-lead ECGs data set provided by Amsterdam UMC. In their approach a GradCAM score was retrieved for each wave type as seen in Figure 2.24. Here the analysis indicated that the most crucial wavetype for the decicion of the DL model was was the QRS complex. On the contrary, the GradCam score for the T and P wave showed a considerable amount of variation. Upon further analysis the researchers were able to assess that the first half of the QRS complex was more relevant than the second half for classification of LQTS patients [4].

To further visualize the relevance of the different sections the researchers calculated the median QRS complex for every LQTS patient. Then the median QRS complex from 100 healthy control patients was retrieved and plotted on top of each other as seen in Figure 2.25. The researchers were thus able to showcase that QRS complexes from the LQTS patients had a lower amplitude compared to the control group [4].

In the end the researchers concluded that the DL models performed better than conventional methods of detecting LQTS patients. When compared to the expert cardiologist the DL models performed better in terms of specificity, while performing the same in terms of sensitivity, The researchers further accredited the explainable method, com-

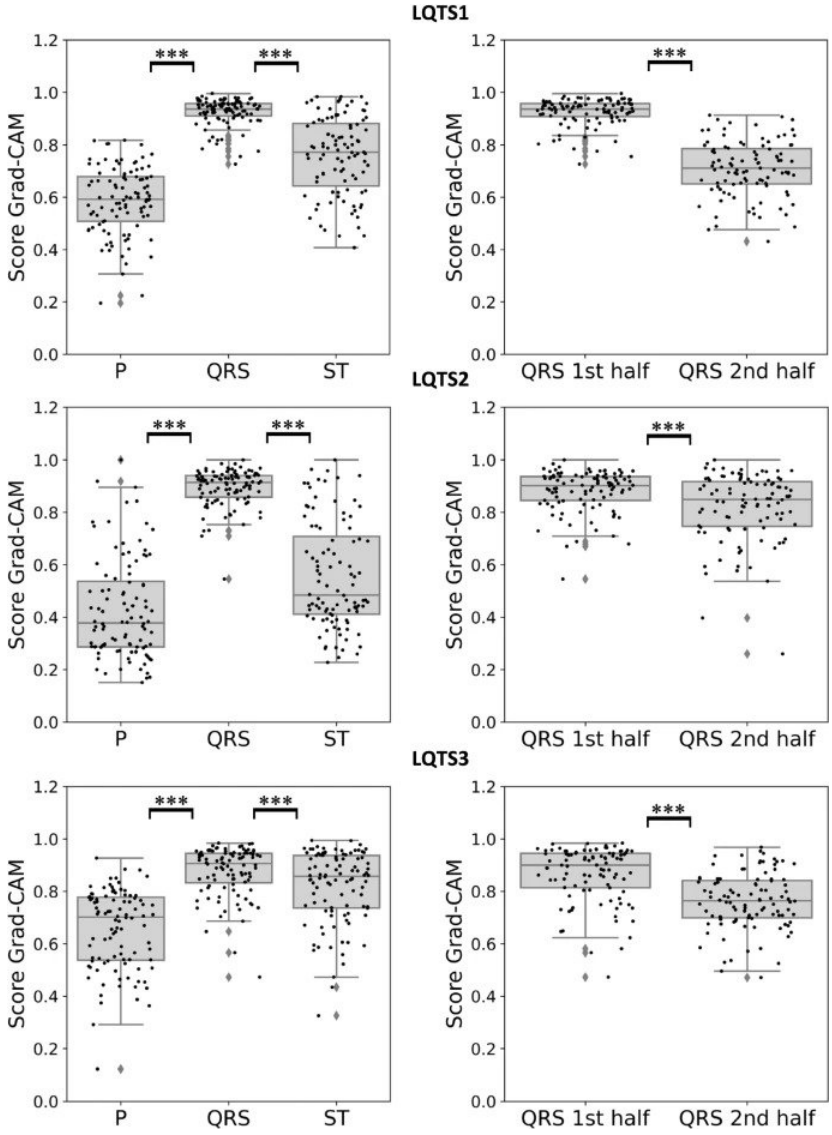

Figure 2.24: Figure showing the calculated GradCam scores for individual features in the ECG complex. [4]

menting; "The explainable AI technique identified the onset of the QRS complex as the most informative region to classify LQTS from non-LQTS patients, a feature previously not associated with this disease." [4].

Lastly they concluded that explainable DL models could be used to identify new features for LQTS from ECG data, and thereby broaden the understanding of the syndrome.

Simultaneously, van de Leur et al. [51] specifically addresses the heat-map based approaches that are most commonly applied as explainable methods in DL for ECG

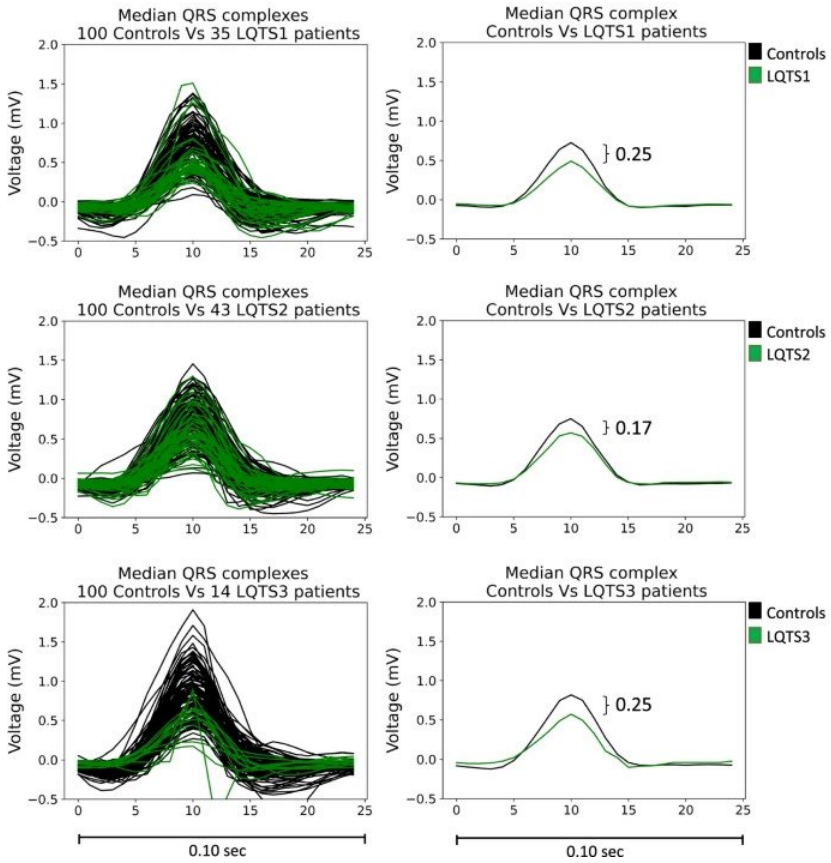

Figure 2.25: Figure showing the plot of the median QRS complex of LQTS patients compared to the median QRS complex of healthy patients. [4]

analysis. In the paper the researchers regard such approaches to be inaccurate.

The main reason behind their assessment is due to such methods only being able to the identify the temporal location of the important ECG feature. From this perspective heat-map based methods such as GradCAM and SHAP do not provide clear insight into exactly what feature is used for the models decision. They argue that even though the GradCAM method may highlight the QRS complex as being of high importance for the model's decision, it does not indicate whether it is the height of the R wave, or the shape of the QRS complex or something else entirely that is used for diagnosis [51].

Furthermore, they criticize heat-map-based methods for only providing explain-ability on the level of an individual ECG, and not for the model itself. The researchers comment that "this combination makes them susceptible to confirmation bias, as we assume that the feature we think is important is also the one that was used in the few

examples that were observed." [51].

To counteract this issue the researchers propose that there should be an increased focus on the development of pipelines that are explainable by design. To achieve this they suggest the use of Variational Auto Encoders (VAE). In this context VAE´s may utilize DNNs to compress an ECG into a limited number of explainable independent factors. Moreover, VAE´s also inherit the ability to reconstruct the original ECG from these factors [51].

In the proposed pipeline visualized in Figure 2.26 an individual ECG was firstly compressed into 32 factors (referred to as the FactorECG) using an encoder. Then the factors were reconstructed into the original ECG using the decoder. Both the encoder, and the decoder were in this framework CNN´s.

After training, the explainable pipeline was compared to current state-of-the-art 'black box' DNNs in the conduction of three tasks: conventional ECG interpretation, detection of reduced EF, and prediction of 1-year mortality. The results indicated that the novel pipeline was able to perform similar to state-of-the-art methods. The researchers further validated the pipeline through applying it on a different data set, in order to test the generalizability.

In the final part of the research the authors emphasized that in contrast to "black box" approaches, their pipeline provided meaningful insight into the morphological ECG changes important for prediction. Moreover, they recommended that future endeavors in the field of DL for ECG analysis should take this into account stating; "Future studies on DNNs for ECGs should employ pipelines that are explainable to facilitate clinical implementation by gaining confidence in artificial intelligence and making it possible to identify biased models." [51].

In summary, there is a notable amount of research related to DL and CNN´s for ECG analysis to consider. The various review studies all establish that CNN´s are particularly efficient and valuable for the purpose of ECG analysis. The articles further express the need for explain-ability and visualization, indicating a clear lack of research dedicated to this aspect of DL for ECG analysis. In terms of using, and col-

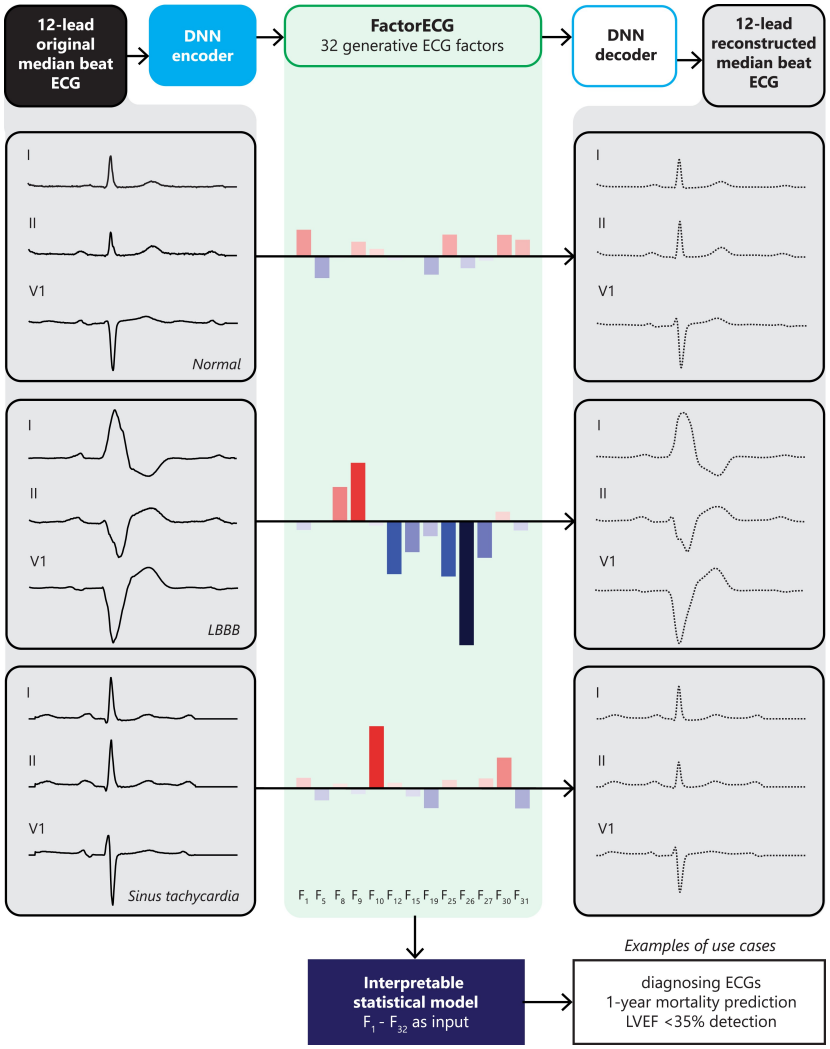

Figure 2.26: Figure showing the structure of the proposed pipeline. [51]

lecting ECG-data, the main-routes can be divided into real or synthetic data sets, with either approach having favorable and unfavorable aspects. Lastly, several XAI methods have been implemented on the purpose of ECG analysis, with the most notable being GradCam and SHAP. In the next chapter we will detail the development of our CNN, justification behind data set usage and factors investigated.

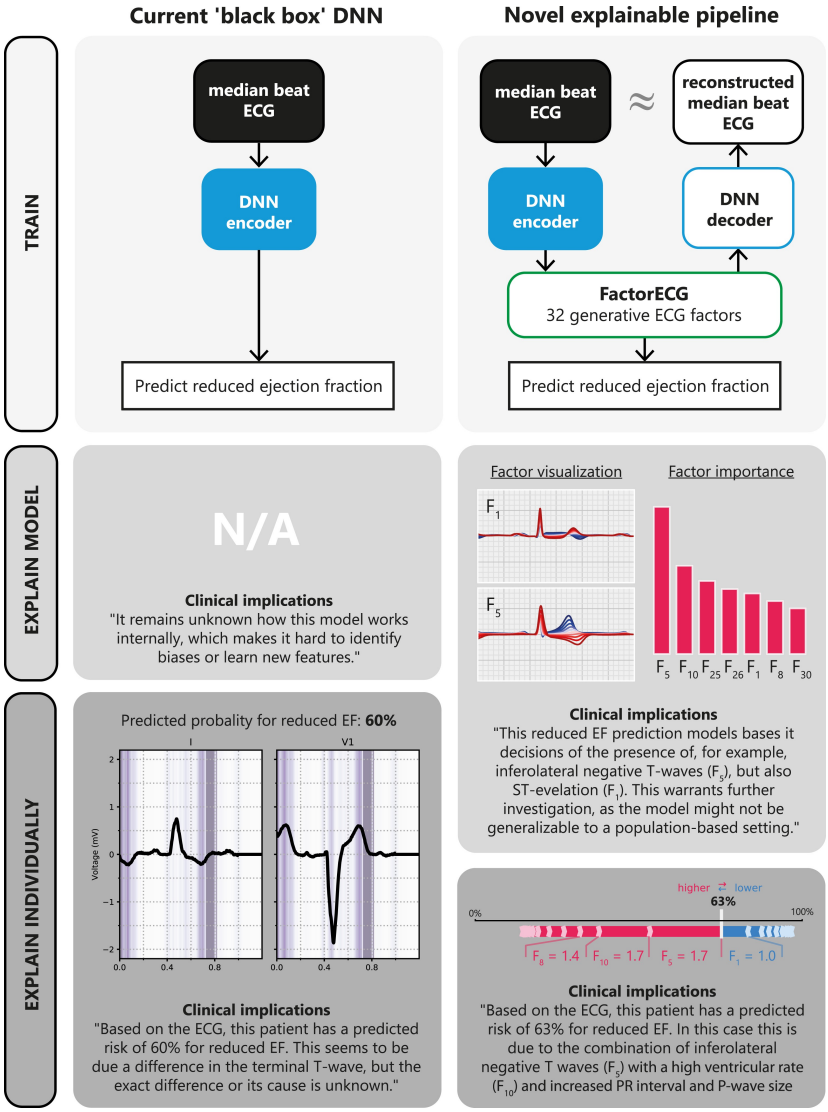

Figure 2.27: Figure showing the benefits of an explainable pipeline compared to traditional explainable methods. [51]

Chapter 3

Methodology

In this section the methodology of the thesis is explained. This includes the approach, and choices of data, models and evaluation metrics.

3.1 Data and data preparation

The following section details the reasoning behind the choice of data set used for development, and provides information regarding the data preparation techniques.

3.1.1 Choice of data set

The choice of data set for our study was the PTB-XL data set. As mentioned in the Background section, this data set is distinguished by the large size, and the diversity of the data [54]. Thus, it was determined to be suitable for the training and evaluation of the CNN model.

The synthetic ECG data set provided by Thambawita et al. [46] was also considered as an option. Similar to the PTB-XL data set their synthetic ECG data set can be said to be a sufficient representative of the general population with regard to both healthy and diagnosed patients [46]. Moreover, there is an added benefit of protecting privacy and health data since the data is generated rather than belonging to any individual. Given the lack of research utilizing deepfaked ECGS for training purposes, it would

Table 3.1: The 5 diagnostic super classes of the PTB-XL data set and the number of records found within each class. [54]

#Records	Superclass	Description
9528	NORM	Normal ECG
5486	MI	Myocardial Infarction
5250	STTC	ST/T Change
4907	CD	Conduction Disurbance
2655	HYP	Hypertrophy

also serve as a decent opportunity to explore and validate the feasibility of using a synthetic ECG data set.

Ultimately the PTB-XL data set was favoured due to the fact that it has been widely used as a benchmark in previous studies, allowing for comparison with existing research and promoting reproducibility.

3.1.2 Data Preparation

As mentioned in the Background section the PTB-XL data set contains ECG data classified within 5 categories known as diagnostic super-classes. Table 3.1 shows the class names as well as the number of records within each category. Moreover, these super-classes contain various sub-classes that may apply to each individual ECG. For our project the purpose of the model was set to classify the correct diagnostic super-class for each ECG sample.

During data preparation the diagnostic super-class information for each ECG signal was extracted from the csv-file containing diagnostic information. This column was then applied to the dataframe containing the raw data. Lastly, raw ECG signals were loaded and mapped to their corresponding diagnostic classes. This approach to data preparation was influenced by author recommendations [54].

The resulting data frame contained the raw ECG signals as well as the corresponding diagnostic classes. Finally, the prepared data were split into training, validation and

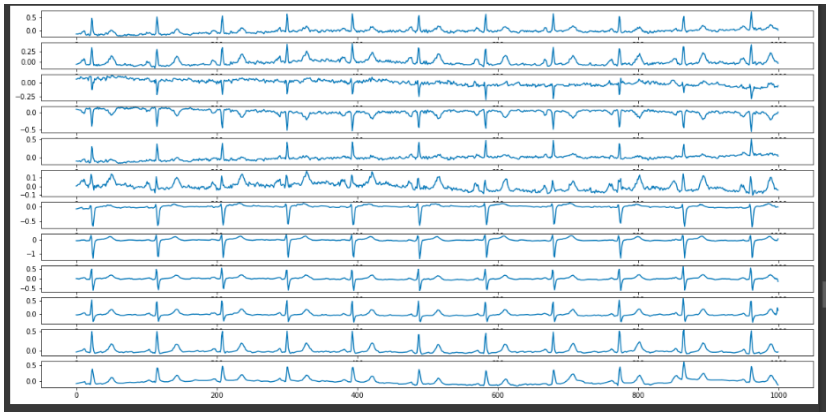

Figure 3.1: Figure showing samples of raw ECG data from the PTB-XL data set.

Table 3.2: The number of records found within each class in the training set.

#Records	Superclass	Description
7660	NORM	Normal ECG
4367	MI	Myocardial Infarction
4149	STTC	ST/T Change
3883	CD	Conduction Disurbance
2123	HYP	Hypertrophy

testing sets to facilitate model development. The training and validation sets were used to train the models, while the testing set was used to evaluate the performance of these models on unseen data. The chosen folds/splits were influenced by the recommendation of the authors behind the publication of the data set [54]. Specifically, 8 folds were used for training, with the last 2 folds used for validation and testing respectively. This was a natural choice given that the last two folds were of higher quality [54]. The distribution of diagnostic-superclasses within the training, validation and test sets is shown in Table 3.2, 3.2 and 3.2 respectively. Before training the data was normalized/standardized using the built in function Standard Scaler from the sklearn Library [41]. The code files for data preparation and the general pipeline can be found in Appendix A.

Table 3.3: The number of records found within each class in the validation set. [54]

#Records	Superclass	Description
923	NORM	Normal ECG
540	MI	Myocardial Infarction
560	STTC	ST/T Change
505	CD	Conduction Disurbance
296	HYP	Hypertrophy

Table 3.4: The number of records found within each class in the test set.

#Records	Superclass	Description
931	NORM	Normal ECG
562	MI	Myocardial Infarction
526	STTC	ST/T Change
510	CD	Conduction Disurbance
230	HYP	Hypertrophy

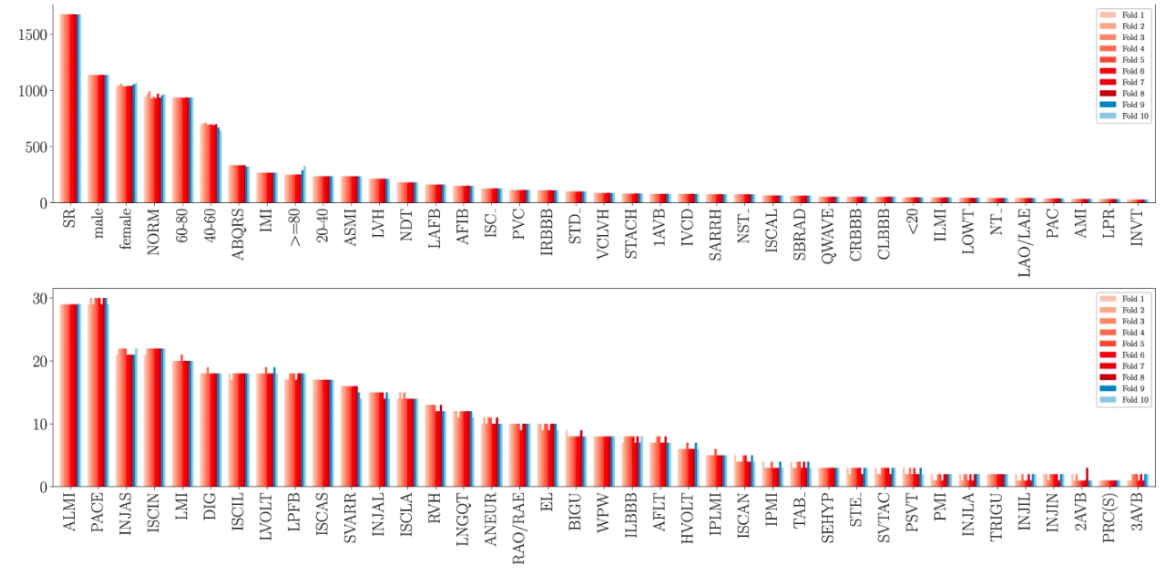

Figure 3.2: Figure showing the distribution of ECG statements, sex and age across 10 folds. [54]

3.2 CNN models

3.2.1 Justification for use of CNNs

The potential of utilizing a CNN for ECG analysis was affirmed by several of the review articles in the theory section [18] [24]. In the majority of these articles a CNN was namely the preferred alternative used by researchers, thereby showcasing the capability of this architecture. Key advantages were given in the superior performance and fast computation of this particular model. For our purpose the convolution type selected was 1D-convolution, given that this convolution type has been particularly effective on 1-dimensional time-series data such as ECG signals [1].

A CNN-LSTM model architecture was also strongly considered given that it was highlighted in several of the articles covering DL for ECG analysis [18] [24] [31]. In view of the ability of this model-type to account for both the temporal and spatial element of an ECG sample, the architecture was deemed especially appropriate. However, in the end, we opted for a CNN model, given the stronger support for its use in the literature and its efficient processing of ECG signals. Additionally, the high resource utilization of CNN-LSTM models served as a hindrance [31].

Other model types mentioned in the Background section such as RNNs, SAEs and DBNs were not chosen due to the comparatively low amount of literature centered around their use.

3.2.2 Base CNN

Figure 3.3 shows the first implementation of a simple CNN. A pipeline was developed to load the ECG data with the corresponding labels into the model. The loss and the accuracy of the model was then plotted in weights and biases, and provided a clear overview of the learning curve of the model.

Initially, the performance was sub-optimal based on all metrics as seen in figure 3.4. In the experiment 5 runs were initiated for 10 epochs each.

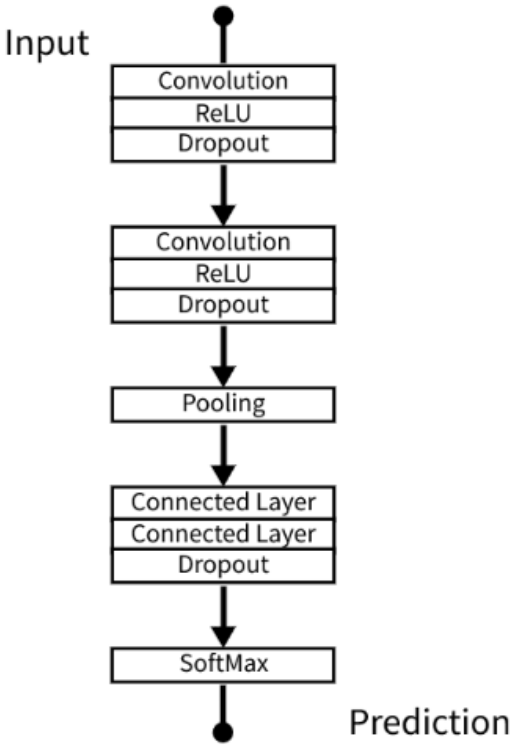

Figure 3.3: Structure of Base CNN model. To begin with SoftMax was used as an activation function.

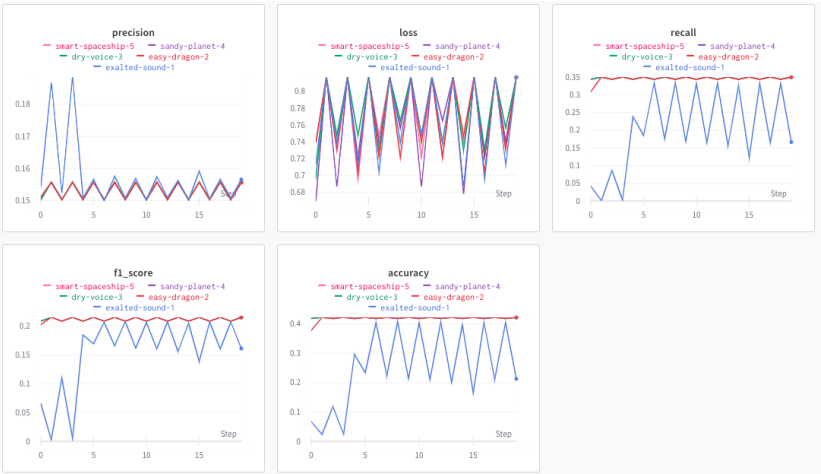

Figure 3.4: Performance metrics from initial testing. The graphs show that; Accuracy stabilized at 42 percent. Loss varied from 0.7 to 0.8. Recall stabilized at 0.35. F1-score at 0.2. Precision at 0.15.

Based on the initial training the activation function applied on the output layer was observed to be non-effective for the given task. The Softmax function was determined to be non-suitable for our multilabel classification problem and a Sigmoid function was

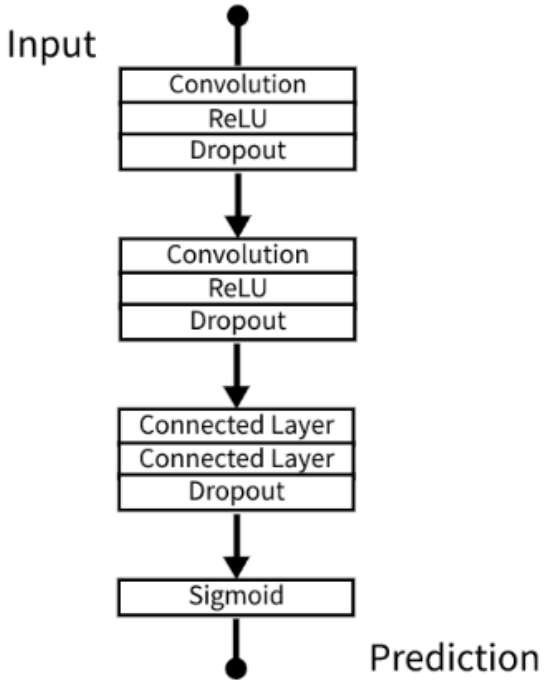

Figure 3.5: Base CNN model with Sigmoid activation function.

used instead as seen in Figure 3.5.

For the application of the CNN, the optimizer "Adam" was selected due to superior results. The two optimizers; "AdaDelta", "SGD", were also briefly tested, but were found to not function properly. The default tuning rate 0.001 is often used with "Adam", and was thus selected as suitable.

For the investigation the subsequent parameters, and components of the Base CNN model were altered and evaluated.

- Dropout rate (0 - 0.8)
- Number of hidden dimensions (32 - 512)
- Batch normalization
- Pooling method

The use of batch normalization was inspired by Hicks et al. [17] given that batch normalization was found to be effective in their CNN. Other parameters such as dropout

rate and number of hidden dimensions are considered central hyperparameters and are natural factors to investigate.

Our inclusion of the main pooling methods is also important to cover, as these are the two main approaches, and to our knowledge little research is published regarding their performance at different levels of dropout rate and varying number of hidden dimensions.

The figures below show the structure of the different CNN models that were tested;

3.2.3 Model with Max Pooling

Figure 3.6 shows the structure of the CNN model fitted with Max Pooling. The base model was altered by adding a Max Pooling layer after the two convolutional blocks.

3.2.4 Model with Average Pooling

Figure 3.7 shows the structure of the CNN model fitted with Average Pooling. The base model was altered by adding an Average Pooling layer after the first and second convolutional blocks respectively.

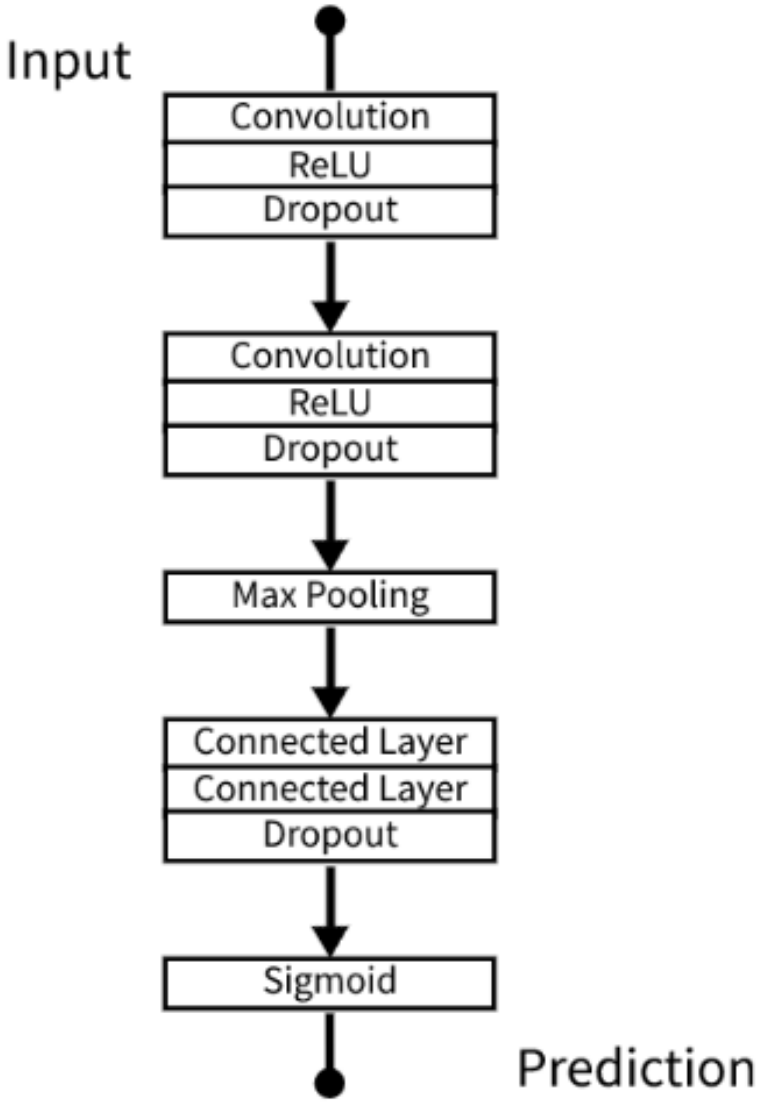

Figure 3.6: CNN Model with Max Pooling.

3.2.5 Model with Batch Normalization and no Pooling

Figure 3.8 shows the structure of the CNN model fitted with Batch Normalization. The base model was altered by adding Batch Normalization after Convolution in both convolutional blocks.

3.2.6 Model with Average Pooling and Batch Normalization

Figure 3.7 shows the structure of the CNN model fitted with Average Pooling and Batch Normalization. The base model was altered by adding Batch Normalization

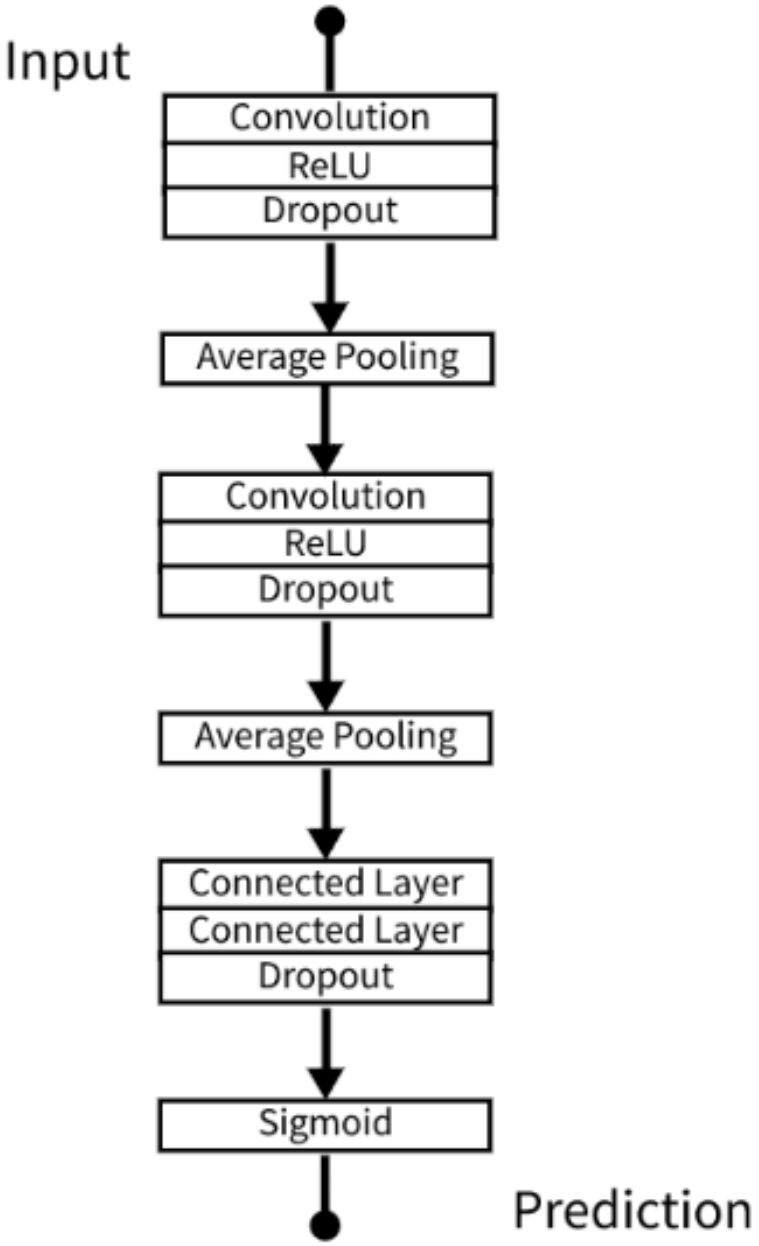

Figure 3.7: CNN Model with Average Pooling.

after Convolution in both convolutional blocks. In addition, Average Pooling layers were inserted after the first and second convolutional blocks respectively.

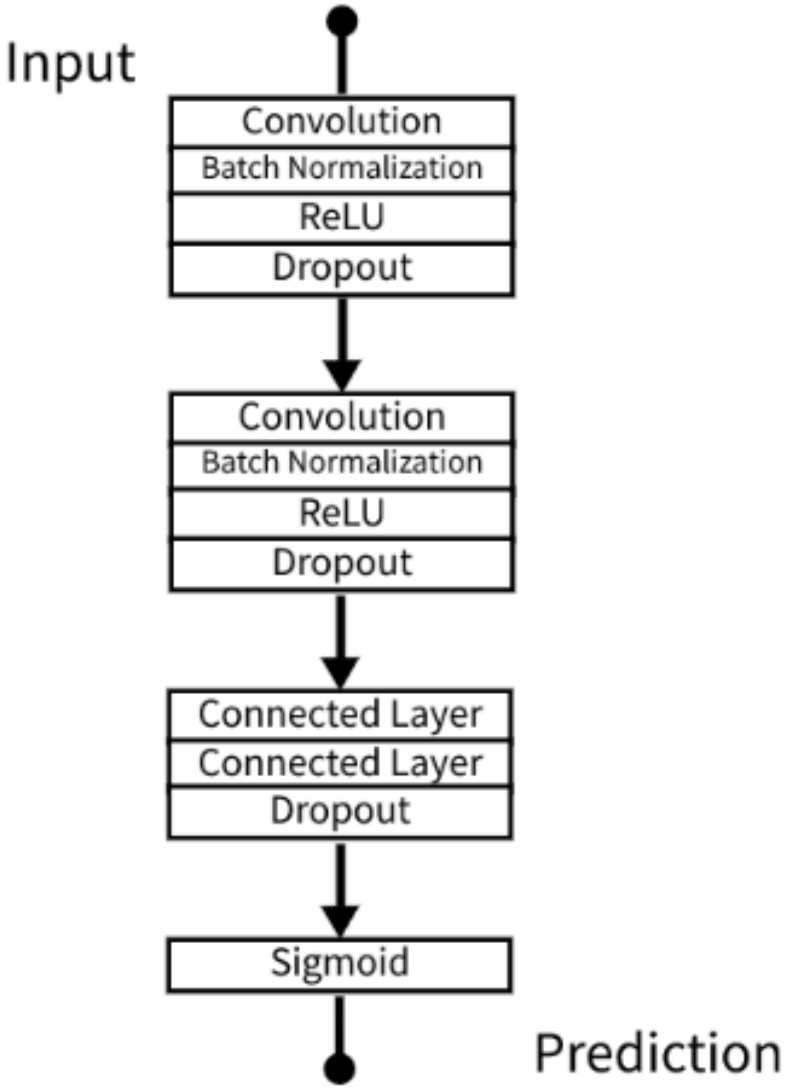

Figure 3.8: CNN Model with Batch Normalization and no Pooling.

3.3 Model evaluation

3.3.1 Metrics

The following metrics were chosen for evaluation, and analysis of the model during the training stage;

Accuracy: A measure of how well a model predicts the correct class labels among all the labels in the data set. It is calculated as the ratio of the number of correct predictions to the total number of predictions made by the model [28].

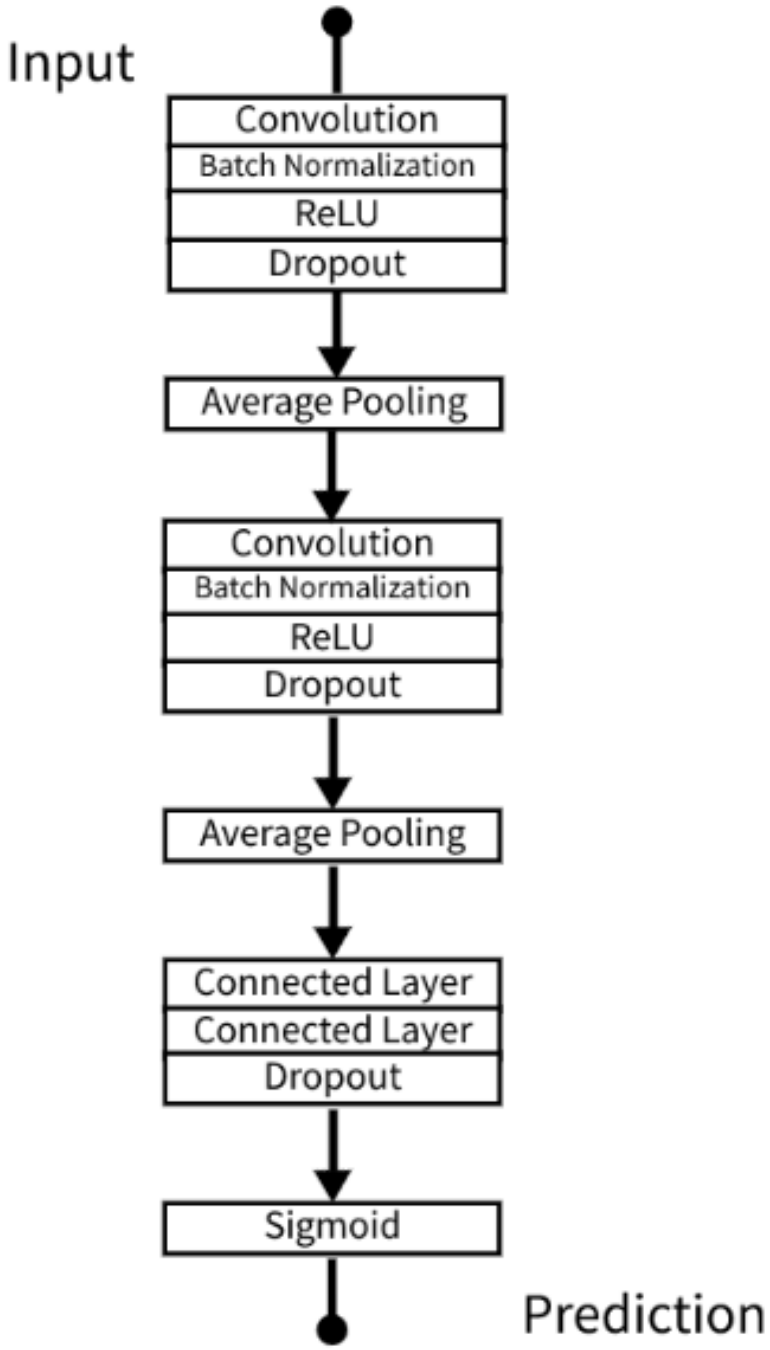

Figure 3.9: CNN Model with Average Pooling and Batch Normalization.

Loss: A measure of the difference between the predicted output of a model and the actual output. The loss function is used to guide the model to adjust its parameters during training to minimize the difference between the predicted and actual outputs.

Precision: A measure of how many of the predicted positive instances are actually true positives. It is calculated as the ratio of true positives to the sum of true positives and

false positives [28].

Recall: A measure of how many of the true positive instances were correctly identified by the model. It is calculated as the ratio of true positives to the sum of true positives and false negatives [28].

F1-score: A metric that combines both precision and recall to provide a single score that summarizes the overall performance of a model. It is calculated as the harmonic mean of precision and recall. A higher F1-score indicates better model performance [28].

These metrics are commonly applied across the field of machine learning, and act as reliable indicators to assess different aspects of a given model's performance.

3.3.2 Baseline Reference Models

A crucial benchmark for the assessment of our model was determined to be 72%. Out of the 21837 samples in the complete data-set, a majority 9528 are samples classified as Normal. Given that a program predicting the majority class (Normal) for every sample would result in an accuracy of 72%, a model providing an accuracy of less than this is essentially useless/redundant.

For this purpose, a baseline model was implemented returning the aforementioned results, in order to facilitate comparison with the CNN´s.

Test metric	DataLoader 0
test_acc	0.7204747796058655
test_f1	0.24759504804729088
test_pres	0.18664229222911846
test_recall	0.37858547954124483

Figure 3.10: Figure showing the results of the Baseline Model on the test set.

3.4 Tools

Various software and tools were utilized in the process. This section contains an overview of the most prominent tools and a brief summary of their application.

Google Colaboratory

Google Colaboratory, or "Colab" for short, is a cloud-based platform for running and sharing code in a Jupyter Notebook environment. It allows for the combination of executable Python code and text along with charts, images, HTML, LaTeX, etc. into a single document stored in Google Drive.

Colab was used throughout the process to write and execute Python code. Various features such as version control, real-time collaboration, and access to data sets were also utilized. Colab was also utilized due to faster and more efficient training through GPUs. Specifically, V100- or A100 Nvidia GPUs were applied depending on availability.

Weights&Biases

Weights and Biases, or W&B for short, is a machine learning experimentation and tracking platform. It allows users to log and visualize experiments, track model performance, and collaborate with others on machine learning projects. Its tools and visualizations help users to gain insights into the performance of their models and to make better decisions during the machine learning development process.

W&B was used to manage the machine learning process. Specifically, the accuracy and loss of different runs were logged to the platform. Moreover, the visualizations were important to gain insights to the further development of the models.

In summary, the considerable amount of research supporting the use of CNN´s for ECG analysis, makes this model-type a preferable choice for further investigation. Likewise, the literature also supports the use of the well-renowned PTB-XL data set for training and testing purposes. Factors that were determined to be of importance for investigation were Pooling, Batch Normalization, Dropout and Hidden dimensions, and these factors were investigated through the use of different model structures. In

the next chapter, the results from the various model structures are presented.

Chapter 4

Results

In this section the collected results from the 5 model-structures tested are presented. These structures include the Base Model (Figure 3.5), Model with Max pooling (Figure 3.6), Model with Average Pooling (Figure 3.7), Model with Batch Normalization (Figure 3.8), and Model with Average pooling and Batch Normalization (Figure 3.9). Selected graphs showing the training progress of the various models are also included.

The results show the performance of these models according to the metrics presented in the Methodology section, when used for prediction on test set. Each model was fitted for 25 epochs, with the best checkpoint from training stored. The best checkpoint was thereafter used for testing.

4.1 Performance comparison of pooling method

Figure 4.1 shows initial training progress from model with Max Pooling compared to model fitted with Average Pooling. The max pooling parameter was set to 4 in all tests. After 10 epochs the model utilizing global average pooling performed better in terms of accuracy.

Table 4.1: Table showing the results of different model components compared to the baseline reference models.

Model	Accuracy	Precision	Recall	F1-score
Baseline Reference Model - Minority	0.2795	0.1469	0.6214	0.2325
Baseline Reference Model - Majority	0.7204	0.1866	0.3785	0.2476
Base CNN	0.7909	0.6617	0.4055	0.5028
Max Pooling	0.7978	0.7079	0.4586	0.5566
Global Average Pooling	0.8341	0.7721	0.4221	0.5458
Batch Normalization	0.8094	0.7358	0.3926	0.5120

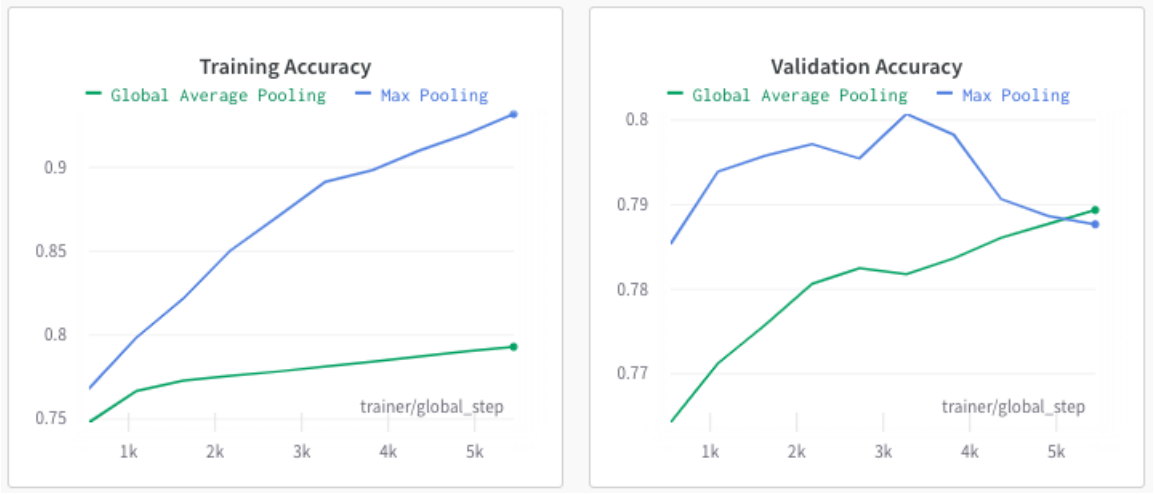

Figure 4.1: Max Pooling vs Average Pooling. After 10 epochs the model utilizing Average pooling performed better in terms of accuracy.

Table 4.1 shows the results of the Base CNN, Model with Max Pooling and Model with Average Pooling compared to benchmark results. In addition to the Reference Model predicting the majority class, a Reference Model predicting the minority class was also included.

4.2 Performance comparison with various dropout rates

Figures 4.2, 4.3 and 4.4 show 5 experiments with dropout rates incrementally increasing from 0 to 0.8. On both training and validation accuracy, the model with lower dropout rates performed better. From Model fitted with Batch Normalization (Figure 3.8)

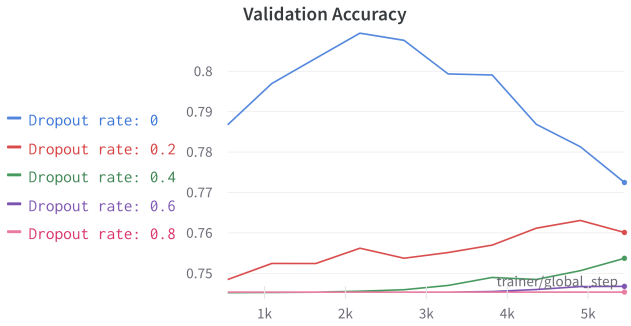

Figure 4.2: Validation accuracy at different dropout rates. The validation accuracy of the CNN is higher at lower dropout rates.

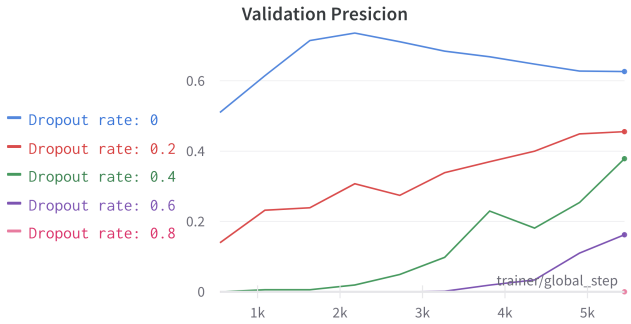

Figure 4.3: Validation precision at different dropout rates. Compared to the accuracy, precision is more similar at different dropout rates.

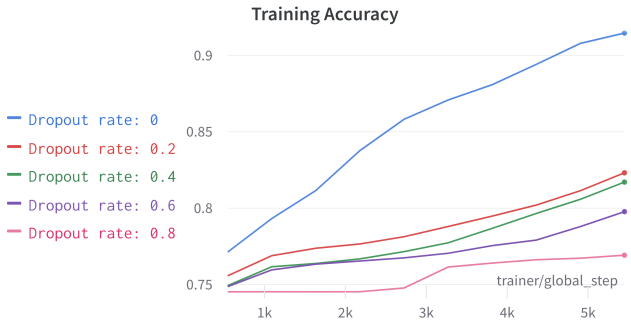

Figure 4.4: Training accuracy at different dropout rates. At lower dropout rates, the CNN is able to achieve higher accuracy and converge faster.

4.3 Performance comparison with various hidden dimensions

The number of hidden dimensions incrementally increases from 32 to 512. Hidden dimensions set to 64 performed best in training. In validation, hidden dimensions set

Table 4.2: Table showing the results of different dropout rates for a model fitted with average pooling.

Model	**Accuracy**	**Precision**	**Recall**	**F1-score**
Model with Average Pooling				
Dropout rate				
0	0.8341	0.7721	0.4221	0.5458
0.2	0.7631	0.4554	0.1225	0.1930
0.4	0.7537	0.3785	0.0539	0.0943
0.6	0.7468	0.1622	0.0225	0.0395
0.8	0.7454	0	0	0

Table 4.3: Table showing the results of different dropout rates for a model fitted with batch normalization and no pooling.

Model	**Accuracy**	**Precision**	**Recall**	**F1-score**
Model with Batch Normalization				
Dropout rate				
0	0.8094	0.7358	0.3926	0.5120
0.2	0.7311	0.4222	0.0712	0.1218
0.4	0.7217	0.3321	0.0361	0.0651
0.6	0.7302	0.1122	0.0129	0.0231
0.8	0.7261	0	0	0

Table 4.4: Table showing the results of different dropout rates for a model fitted with average pooling and batch normalization.

Model	**Accuracy**	**Precision**	**Recall**	**F1-score**
Model with Average Pooling + Batch Normalization				
Dropout rate				
0	0.8373	0.7729	0.4253	0.5486
0.2	0.7629	0.4524	0.1235	0.1940
0.4	0.7536	0.3780	0.0432	0.0775
0.6	0.7467	0.1615	0.0209	0.0370
0.8	0.7454	0	0	0

to 256 performed best. From Model fitted with Batch Normalization (Figure 3.8)

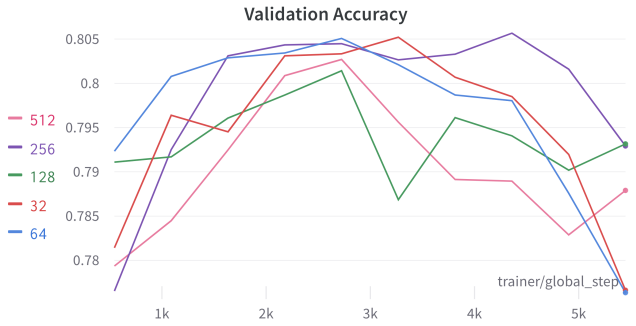

Figure 4.5: Validation accuracy with varying number of hidden dimensions. The CNN performed best with hidden dimensions set to 256.

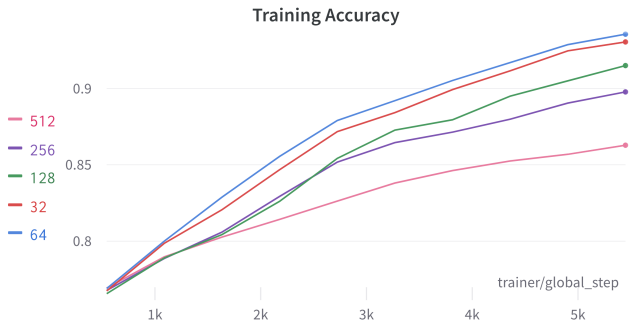

Figure 4.6: Training accuracy with varying number of hidden dimensions. The training curve converged most quickly with the hidden dimensions set to 64.

Table 4.5: Table showing the results of different hidden dimensions for a model fitted with average pooling.

Model	**Accuracy**	**Precision**	**Recall**	**F1-score**
Model with Average Pooling				
Hidden dimensions				
32	0.8341	0.7721	0.4221	0.5458
64	0.8051	0.7174	0.4188	0.5288
128	0.8014	0.6882	0.4023	0.5077
256	0.8392	0.7881	0.4236	0.5510
512	0.8027	0.6882	0.3461	0.4605

Table 4.6: Table showing the results of different hidden dimensions for a model fitted with batch normalization and no pooling.

Model	**Accuracy**	**Precision**	**Recall**	**F1-score**
Model with Batch Normalization				
Hidden dimensions				
32	0.8094	0.7358	0.3926	0.5120
64	0.7941	0.6932	0.3821	0.4926
128	0.7902	0.6755	0.3703	0.4783
256	0.8091	0.7358	0.3920	0.5114
512	0.7985	0.6625	0.3120	0.4242

Table 4.7: Table showing the results of different hidden dimensions for a model fitted with average pooling and batch normalization.

Model	**Accuracy**	**Precision**	**Recall**	**F1-score**
Model with Average Pooling + Batch Normalization				
Hidden dimensions				
32	0.8373	0.7729	0.4253	0.5486
64	0.8121	0.7304	0.4270	0.5389
128	0.8107	0.7012	0.4116	0.5187
256	0.8441	0.7941	0.4301	0.5579
512	0.8030	0.7002	0.4024	0.5110

Chapter 5

Discussion

The following chapter highlights and discusses the results gathered from the developed CNN. Among other aspects the relevant findings will be discussed in terms of impact and generalizability, Ethical considerations regarding the development and deployment of our model will also be emphasized. Lastly, the discussion section highlights the application of explainable algorithms in light of our own development process and the conducted literature review.

5.1 Performance of model

5.1.1 Initial results

The results from Table 4.1 demonstrate that the base CNN, as well as the CNN´s fitted with max pooling, average pooling, and batch normalization all yielded superior performance when compared to the Baseline Reference Models. Among the models in Table 4.1, the model utilizing average pooling returned the highest accuracy as well as highest F1-score. Compared to the most inaccurate reference model, namely the model predicting the minority class for every sample, the accuracy increased from 27% to 83%, which constitutes an increase of 56%. Similarly, upon comparison of the reference model which was set to predict the majority class for every sample, a significant leap in performance is observable. Here the accuracy increased from 72%

to 83%, representing an increase of 11%.

In addition to the increased accuracy achieved trough utilizing DL models, the significant increase in precision throughout Table 4.1 is also a promising indication that such models may be particularly effective in correctly identifying true positive cases. This result is particularly relevant in medical diagnosis, where false positive diagnoses may cause serious implications for patient care, as well as lead to unnecessary anxiety for relatives. Overall, it is evident based on the data from Table 4.1 that there is a clear potential of utilizing CNN's for the purpose of diagnosis classification, especially when compared to naive/dumb models.

5.1.2 Average Pooling

A closer inspection of the model fitted with Average Pooling is presented in Table 4.2, which provides an overview of the impact of dropout rates on this particular pooling method. Here the results indicate that the performance of the model deteriorates as the dropout rate increases. For instance, a dropout rate of 0.2 resulted in a decrease in accuracy to 76.31% and an F1-score of 0.1930. Further increases in the dropout rate led to a decrease in performance across all metrics, with the model with a dropout rate of 0.8 achieving an accuracy of 74.54% and an F1-score of 0. These results suggest that over-regularization through the use of dropout layers can hurt the performance of the CNN.

In terms of the effect of hidden dimensions on the performance, Table 4.5 shows how the model with 256 hidden dimensions had the highest accuracy of 83.92% and corresponding F1-score of 0.5510. The models with 64 and 128 hidden dimensions had lower accuracy and F1-scores compared to the base model. Increasing the number of hidden dimension beyond 256 to 512 resulted in an accuracy of 80.27% and an F1-score of 0.4605. This may imply that increasing the number of hidden dimensions beyond a certain point does not necessarily lead to improved performance and may even lead to overfitting. Overall, the findings in Table 4.2 and 4.5 highlight the importance of selecting appropriate values for hyper-parameters such as dropout rate and hidden dimensions to optimize the performance of the CNN.

5.1.3 Batch Normalization

Table 4.3 shows the results of different dropout rates for the model fitted with batch normalization. When evaluating the impact of different dropout rates, the results indicate similar to the findings from Table 4.2, that increasing the dropout rate causes a decrease in all performance metrics; accuracy, precision, recall, and F1-score. This trend is observed consistently across all dropout rates tested, from 0.2 to 0.8.

Similarly, when evaluating the impact of different hidden dimensions, the results from Table 4.6 indicate that the performance of the model decreases as the number of hidden dimensions increases. This trend is also observed consistently across the hidden dimension values; 64, 128 and 512. The value of 256 may be regarded as an exception, given that the performance increases when compared to the previous level of 128.

It is interesting to note that the model with batch normalization performed relatively poorly compared to the model with average pooling in Table 4.5. The best-performing model in Table 4.3 had an accuracy of 80% and a F1-score of 0.51 which is lower than several of the models in Table 4.5. This suggests that the use of batch normalization may not be as effective as using average pooling in this particular classification task.

Overall, the findings in Table 4.3 suggest that, for the model with batch normalization, it may be better to use a lower dropout rate and fewer hidden dimensions to achieve better performance. However, the overall performance of the model with batch normalization is still relatively low compared to the other model-architectures evaluated in this study.

5.1.4 Average Pooling used in combination with Batch Normalization

Lastly, Table 4.4 and 4.7 shows the results of the model with average pooling and batch normalization with different dropout rates and hidden dimensions. We can observe that the model achieves the highest accuracy, precision, recall, and F1-score when using a dropout rate of 0 and hidden dimension of 256. This combination results in an accuracy

of 84%, precision of 79%, recall of 43%, and F1-score of 0.5579.

As in the previous experiments, as the dropout rate increases, a decrease in all the evaluation metrics is observable, suggesting that high dropout rates may cause the model to lose crucial information during training. Consistent with the pattern found in Table 4.5 and 4.6, increasing the number of hidden dimension also produces a decrease in accuracy and precision, while recall and F1-score remain relatively stable.

Overall, we can conclude that the combination of average pooling and batch normalization improves the CNN's performance compared to using only one of these techniques. The best performance is achieved with a dropout rate of 0 and a hidden dimension of 256.

5.2 Interpretation

Within the frame of the investigation the maximum performance of the CNN was achieved through utilizing average pooling in combination with batch normalization. In the mentioned configuration the dropout layer was set to 0, and the number of hidden dimensions was 256 as seen in Table 4.7. Using this structure the model achieved approximately 84% accuracy and 79% precision. The recall and F1-score for the implementation returned 43% and 0.55 respectively.

A closer inspection of the gathered results indicate that average pooling was the most effective factor in increasing performance across all metrics. In particular, the accuracy of the baseline model utilizing Average Pooling (83%) outperformed the baseline model with Max Pooling (79%). This observation is inconsistent with a different experiment using the same data set and model type, in which Max Pooling was favored as the superior pooling method [22]. However, the effectiveness of Average Pooling is supported by Hicks et al. [17] as they were able to effectively use this method in their research. Moreover, batch normalization was also found to have a positive effect on the performance of the CNN. However, compared to the model utilizing Max Pooling the results indicate a relatively small increase of performance, and when varying the number of hidden dimensions no significant difference was found.

It is also worth considering that for the majority of the different runs, the F1-score remained relatively stable, close to a medium level of 0.5. Given that the F1-score represents the harmonic mean between precision and recall, it is evident through the results that in many cases the relatively high precision is offset by a low recall value. This trade-off between precision and recall can have significant implications in the context of ECG analysis. A low recall value suggests that the model is missing a large number of actual positive cases, resulting in false negatives. In the case of medical diagnosis, relying on a model with low recall value can lead to missed diagnoses and dangerous consequences for the patient.

One notable observation was that varying the dropout rates did not have the desired outcome on the performance of the CNN. For our experiment increasing the dropout rate led to lower accuracy in all cases across all runs. As previously mentioned dropout is generally useful to prevent over-fitting, but in this case it may have caused the model unable to learn key information and patterns during training. This suggests that our model is not particularly robust, which is not ideal.

The effect of introducing dropout is inconsistent with the outcome of the approach used in Hicks et al. [17] in which dropout rates were shown to have a positive effect on the predicted output of their model. It may be possible to point to the difference in the data set as one of the factors behind this discrepancy. In their research they utilized the Danish General Suburban Population Study(GESUS) [21]. Although, this data set has a near equal representation of male and female subjects similar to the PTB-XL data set, the PTB-XL data set is a significantly larger data set and includes a larger variance of different cardiovascular diseases.

In terms of the alteration of the number of hidden dimensions, 256 resulted in the best performance. This was evident both in the experiments utilizing Average Pooling found in Table 4.2, and in the experiments where the CNN was fitted with both Average Pooling and Batch Normalization found in Table **??**. Interestingly, by the exception of 256 dimensions the performance decreased in all other dimensions that were tested. The occurrence of this pattern could be due to over-fitting, as a higher hidden dimension results in a more complex model that is prone to over-fitting. In the case

of testing the CNN fitted with batch normalization and no pooling, the results were somewhat different. Here setting the number of hidden dimensions to 256 resulted in similar performance to the performance with 32 hidden dimensions. Still, these results suggest that 256 is the preferred number of dimensions since the performance decreases significantly in the other cases.

However, limitations of the study should also be considered while interpreting the results. As the research was conducted on a single data set, the generalizability of the findings to other data sets or populations may be limited. Additionally, as the model was trained on a static data set, it may not be able to detect emerging cardiovascular diseases or variations in health data over time.

Overall, the results show that there is a significant advantage to using deep learning for ECG analysis, while at the same affirming the feasibility of using CNN´s for this purpose as demonstrated in other studies. Moreover, the developed pipeline was found to be a productive and purposeful method of gathering results. Specifically, the streamlined and systematic structure facilitated the change of filters, layers, parameters and hyper-parameters in a time-efficient manner. Lastly, the experiments with dropout rates and hidden dimensions show that finding the optimal configuration of these hyper-parameters can be critical to achieve better model performance.

5.3 Impact and generalizability

The impact of the results can be established from certain perspectives. The development of a pipeline that can be further utilized for investigating and exploring the matter is one such perspective. Additionally, our research confirms the findings of previous well-conducted studies, which have established the feasibility and effectiveness of CNNs for the purpose under consideration.

More precisely, the research findings suggest that the developed CNN model is effective in the automated detection of cardiovascular diseases using ECG data. The model achieved high accuracy and precision in detecting the cardiovascular conditions; Myocardial Infarction, ST/T Change, Conduction Disturbance and Hypertrophy. The

results also indicate that pre-processing and normalization of the ECG data, as well as the inclusion of certain model components, such as batch normalization and pooling, are critical for achieving high performance in the developed CNN model. These findings could be useful for further development and improvement of DL models for healthcare applications.

Still, it may be difficult to gauge to what extent the findings of this study can be generalized, as it is a specific investigation of one Convolutional Neural Network. There are several limitations in this regard, which may not be applicable to other cases. Specifically, as this is a health data issue, there may be significant variations in the health data of the population, which may not align with our findings. Other studies may also implement unique pre-processing and normalization procedures. Nevertheless, our use of the PTB-XL data set, which is deemed one of the largest and most representative data sets available provides some grounds for generalization.

It is important to reiterate that the investigation was restricted to chosen factors, and thus is not a conclusive analysis. A number of other factors such as regularization, changing of parameters and tuning of learning rate may increase the performance of the model further.

5.4 Utility value

The utility value of our research lies in the development and testing of a CNN model for the automated detection of cardiovascular diseases using ECG data. Our research thus contributes to the growing body of literature on the use of machine learning techniques in healthcare, particularly in the early detection and diagnosis of cardiovascular diseases, which is crucial for preventing and treating such conditions.

While the developed CNN model achieved acceptable performance, it performed relatively modest compared to other studies found in literature [18]. Indicating that certain existing models are more proficient and could be better suited for clinical settings where high accuracy is critical for patient outcomes.

Given the importance of accuracy in healthcare-related work, it may not be advisable

to use the developed CNN model in isolation. However, the model could still be used in a compelling and innovative way. Given that there is a need for increased transparency around the use of DL models in ECG analysis our model could be used in conjunction with other CNN models in an interactive online web app. The proposed use-case of such an application could be to visualize different models performance through explainable algorithms. In this context, including a model that does not perform at the highest level could be useful to highlight how it arrives at its predictions and to compare its performance with other models.

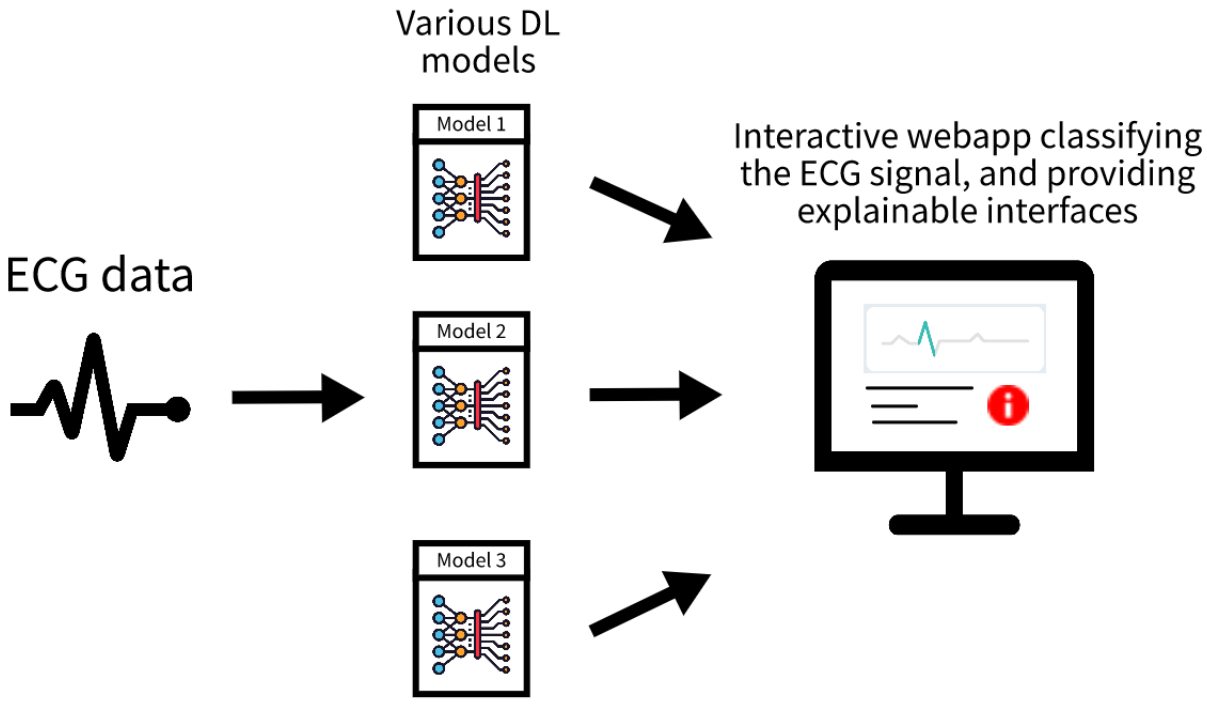

Figure 5.1: Diagram showcasing how a potential WebApp could be structured.

In other words, while our research presents a novel CNN model for automated detection of cardiovascular diseases using ECG data, its limitations in terms of accuracy suggest that other models may be more suitable for clinical applications. However, the developed model could still have utility in an interactive web app alongside other models, to highlight the differences in performances while providing insights into how the models arrive at their respective predictions. Additionally, the pipeline developed

in our research can be used as a starting point for further exploration of the use of DL techniques in ECG data, particularly in the analysis of the PTB-XL data set.

5.5 Comparison to state-of-the-art models

As mentioned the best edition of the developed CNN model achieved approximately 84.41% accuracy and 79.41% precision. The recall and F1-score for this implementation returned 43.01% and 0.55 respectively. Looking at several different research papers we can firstly establish an idea of how well our model performed compared to recent CNN models.

Among the 19 studies utilizing CNN´s presented in Liu et al. [24], the highest performing model achieved an accuracy of 99.78%. Similar to our developed CNN this model utilized 1d convolution. The data set used for training of the model was the PTB data set, which stems from the same institution behind the PTB-XL data set. Compared to our developed CNN the difference in accuracy was 15.37 percent. While this difference is substantial, it is important to note that the size and complexity of the data sets used in each study may have had a significant impact on the accuracy of the models.

On the other hand, the lowest performing CNN presented, achieved an accuracy of 80.9%, which is 3.51% lower than our developed CNN. This particular study was focused around the use of a 2D CNN, and utilized a self-constructed data set of 30,000 unique patients. The mean accuracy of the 19 studies utilizing CNNs in this review article was 95.44%. Therefore, our developed CNN scores are approximately 11% lower than the mean accuracy of CNNs found in the review article.

Murat et al. [31] also presents a selection of studies centered around the us of CNN´s for ECG analysis. Based on Table 2.2 it is observable that several of the models achieved an accuracy as high as 99%. The mean accuracy of the selected studies in Table 2.2 is calculated to be 97.21% which indicates a 12.8% difference when compared to the developed CNN. In light of the substantial difference in accuracy between the developed CNN and the mean accuracy from studies presented both review articles, it is clear that further investigation into the parameters of the network may be beneficial.

It may also be purposeful to compare the results from our developed CNN to certain state-of-the-art CNN-LSTM models. As presented in the literature review several studies found CNN-LSTM models to perform very well when used for ECG analysis classification [18]. In this regard, the capability of the CNN-LSTM model to capture both the temporal and the spatial field is often emphasized [24] [31].

For this purpose Table 2.1 from the Background section is relevant. Based on the data from the table it is evident that the majority of the models utilizing CNN-LSTM structures performed at relatively high levels. Compared to the developed CNN model many studies returned significantly better results with several reaching accuracy's above 95%.

One study did achieve an accuracy of 81.2% which is lower than our model. The research study in question utilized the 1st China Physiological Signal Challenge data set. As mentioned in the Background section this data set holds many similarities to the PTB-XL data set in terms of size and distribution. Given that the rest of the models were fitted on smaller data sets, there may be a basis to argue that particularly large data sets such as the PTB-XL data set and the 1st China Physiological Signal Challenge data set have a negative effect on the overall performance of the model. Though a specific investigation into the performance of large data sets compared to smaller is needed to make any meaningful conclusion.

Since the mentioned CNN and CNN-LSTM models are employed in varying contexts and employ different data sets it is not feasible to draw a direct comparison with our model. Other factors such as variations in model architecture, hyperparameters, and training strategies additionally impact the outcome and the performance of the model. Nevertheless, these comparisons provide some indication that there is room for increasing the performance of the developed CNN.

5.6 Ethical dilemmas and explain-ability

Given the relatively high accuracy observable in our results, the implementation of the PTB-XL data set can be deemed successful for our intended purpose. However, due to the lack of studies employing synthetic data sets, it is difficult to gauge how

traditional data sets containing real ECG signals compare to synthetic data sets. There are nonetheless several reasons to argue that synthetic data sets can act as the future of data in healthcare, and thereby be suitable replacements for established ECG data sets such as the PTB-XL data set. In particular considering the privacy issues that revolve around the use of sensitive health data. At the same time it must be stated that issues of explainability may increase. Given that a model may have only trained on synthetic data it can be intricate to explain the process to doctors and radiologists. Thus it may become more troublesome to convince medical professionals that the technology can be trusted.

A potential solution to this issue could be to train partially on synthetic data sets, and partially on data sets containing real ECG signals. To that end, an approach in which DL models pre-train on a larger real data set, before finetuning on a smaller synthetic data set may be suitable as demonstrated in Strodthoff et al. [44]. In this way natural ECG data can still be used which may add a layer of security in knowing that the ECG data is representative of the general population. At the same time it can be helpful in accounting for new cardiovascular diseases that have not yet been discovered. On the other hand making use of a synthetic ECG data alleviates some of the concerns related to privacy.

Throughout our research, implementation and investigation, the importance of explainable AI was apparent. This need arises not only to ensure that the model is functioning correctly before deployment, but also for researchers to gain a better understanding of its workings during development. For our study when the model was not behaving correctly during the training process, debugging often became time-consuming and confusing. Although this problematic aspect of dealing with complex models is not new, it is exceptionally crucial in healthcare related tasks that explain-ability is present at every stage of the process.

In some cases, researchers may mistakenly believe that a model is functioning properly and ready for deployment in a clinical setting when, in reality, it is not. An infamous example of this occurred when a DL model was deployed to classify X-ray images of healthy and sick patients and achieved high accuracy. However, it was later discovered

that the model had simply trained to read the label of the corresponding medical department included in the X-ray image [58]. While we attempted to prevent such instances through techniques such as pooling and normalization, it is not possible to conclusively guarantee that unwanted learning behavior doesn't occur.

Our training process and experiences with understanding the properties of the CNN confirms that understanding these models before deployment is a necessity. The existence of multiple legal guidelines that recommend or endorse the use of explainable DL models further validates this claim [16, 37, 42]. Deploying such models without a clear understanding of their effects can not be considered ethically sound, as it can become a matter of life-and-death.

It is in this context that using DL models in combination with explainable algorithms is pivotal. Methods such as SHAP, Lime and GradCAM have been tried in several studies and is present in a large quantity of literature [34] [25]. Simultaneously, there is a growing number of studies applying and investigating these methods, and thus far the sentiment from the majority of the authors is overwhelmingly positive.

Still, some researchers have expressed criticism regarding certain flaws with the current trend of explainable algorithms, and put forth the adoption of explainable pipelines as a viable solution. As mentioned in van de Leur et al. [51] an explainable pipeline may be a preferred solution to enable visualization and understanding, as opposed to heat-map based explainable algorithms. Specifically, heat-map based methods such as SHAP, LIME and GradCAM only account for the temporal location of an important ECG feature. An exception can come in the form of PSI, which takes use of the time series derivative in order to emphasize important features [34]. An alternative application of GradCAM presented in Aufiero et al. [4] may also be effective to handle this aspect of the ECG signal.

The fact that heat-map based approaches only provide explanations after the prediction has been made and do not provide insight to the model itself is another drawback that affirms the need for explainable pipelines [51]. In our case, although not explainable, a robust pipeline effectivized the process and made it easier to debug and test the model

to ensure that is was working as intended.

Regardless, there is a clear need for studies enabling medical professionals with the opportunity to assess and provide feedback on different explainable techniques. The visual interface of various methods such as SHAP, Lime and GradCAM can in many cases appear very similar [20] [2]. Therefore, it is important to remember that the ability of an individual to understand a visual interface is subjective and may influenced by various factors such as individual learning styles, prior knowledge, and experience. In other words, what one person perceives as easy and straightforward, another may find complex and confusing. Ultimately, the intention is for medical professionals to use these models, and thus their input is the most valuable. By that extension the lack of studies centered around this in the current literature is concerning. Naturally, it is also essential to establish guidelines and conventions before general adoption can take place.

Nonetheless, more research needs to be conducted on this matter if DL for ECG analysis is to be widely accepted. Recent statistics show that there is a lack of trust that hinders the overall adoption of such systems [34]. In 2017, a survey carried out across 85 hospitals, namely showed that only around 5% of the hospitals expressed interest in implementing Artificial Intelligence (AI) solutions. Many of these hospitals were uncertain about when to start deploying these solutions, and cited the primary obstacles to be the lack of buy-in from executive and physician technology, as well as a lack of trust [9] [45].

5.7 Future points of interest

In summary, the research has several notable strengths. Firstly, it thoroughly examines the relevant literature, demonstrating a comprehensive understanding of the current state of research in ECG analysis. Secondly, through the development of a working pipeline and CNN model for ECG analysis, which can serve as valuable contributions to the field. Another strength is the choice of the PTB-XL data set, which is a large and rich data set that contains a wide range of ECG signals, enabling the model to learn

from a diverse set of data. In this regard, the data set contributors are anonymized, ensuring that the study adheres to ethical considerations and poses no risk to privacy.

At the same time, the research also exhibits certain limitations. The study was conducted on a single data set, which may limit the generalizability of the findings to other data sets or populations. Additionally, the model was constructed to predict the diagnostic super-classes, and thereby fails to account for various diagnostic sub-classes that may be of interest. In this context the limited time-frame of the study can be highlighted as a hindrance, and it is evident that an extended time-frame may allow for a more thorough investigation of relevant parameters. Moreover, while our research provides evidence for the potential utility of CNN´s in healthcare, further research is needed to validate the use of such models in clinical settings and to assess its impact on patient outcomes.

Opportunities for future research in this area are vast, and a focus on a more inclusive and thorough investigation should be prioritized. Implementing other models for easy comparison, such as RNN´s and CNN-LSTM models can also be considered a purposeful endeavor, that might yield insightful findings. Finally, incorporating explainable algorithms in future research is vital and can be achieved by one of two ways. Either through leveraging existing methods such as SHAP and GradCAM or by modifying the existing pipeline to create an interpretable framework.

Chapter 6

Conclusion

6.1 Findings

In conclusion our findings indicate that there is a clear potential of utilizing CNN´s for ECG analysis. Specifically, our developed CNN was effective at classifying ECG signals according to the diagnostic classes; MI, STTC, CD and Hypertrophy. In this context the structural factor that was found to be most influential was Average Pooling, and the combination of Average Pooling and Batch Normalization was determined to be the most effective implementation.

Moreover, through our work we conclude that the use of XAI methods is especially important to investigate and validate. This is due to the fact that both during the development process and after deployment, the models may contain errors that are not properly understood. Needless to say, the outcome of a CNN or any other deep learning model not functioning as intended in a medical setting can be fatal. In this context, adoption of explainable pipelines as presented in [51] may be the best way forward.

At the same time, if the explainable methods are to be widely adopted, then this also necessitates the need for a discussion around conventions. Given that the experience of every medical professional is different, their views about what constitutes an understandable visualization also varies. Different explainable methods may cause confusion

between professions working within the same medical facility, and is especially likely to occur in cases where there is cooperation between doctors from different hospitals or countries.

6.2 Limitations

The presented research is limited by certain factors. Firstly, the developed CNN was constructed to make predictions on diagnostic classes representing general cardiovascular conditions, and did not account for various underlying conditions. Secondly, the lack of testing different model structures to substantiate and further corroborate the findings is a deficiency. Particularly, given the vast amount of literature supporting CNN-LSTM models, implementing this specific model-type would be of use. Lastly, directly applying an XAI method on the developed CNN could bring forth interesting insights to the inner mechanics of the model, and shed light on the feasibility of utilizing the model in a clinical setting.

6.3 Future works

For future work, the developed pipeline can be utilized as a suitable starting point for further testing of various factors such as layers, filters, parameters and hyperparameters. The pipeline may also be altered to facilitate an explainable process.

Additionally, future works should be centered around the inclusion of medical professionals in the design process of explainable methods. Although, several studies focused on deep learning for ECG analysis have included medical professionals, these studies focus on affirming the feasibility of using the proposed model, and do not examine how the professionals experience the various XAI methods. Inspiration can be drawn from a recent study investigating the conditions in which AI can augment human diagnostic skills [10].

Bibliography

(1) Ahmed, A. A.; Ali, W.; Abdullah, T. A.; Malebary, S. J. *Mathematics* **2023**, *11*, 562.

(2) Anand, A.; Kadian, T.; Shetty, M. K.; Gupta, A. *Biomedical Signal Processing and Control* **2022**, *75*, 103584.

(3) Association, A. H. Heart Conduction Disorders, `https://www.heart.org/en/health-topics/arrhythmia/about-arrhythmia/conduction-disorders`, Accessed: 2023-03-24, 2022.

(4) Aufiero, S.; Bleijendaal, H.; Robyns, T.; Vandenberk, B.; Krijger, C.; Bezzina, C.; Zwinderman, A. H.; Wilde, A. A.; Pinto, Y. M. *BMC medicine* **2022**, *20*, 1–12.

(5) Avanzato, R.; Beritelli, F. *Electronics* **2020**, *9*, 951.

(6) Bakshi, R. Stacked Autoencoders. `https://towardsdatascience.com/stacked-autoencoders-f0a4391ae282`, Accessed: 2023-03-21, 2021.

(7) Bhoite, S.; Ansari, G.; Patil, C.; Thatte, S.; Magar, V.; Gandhi, K. In *ICT Infrastructure and Computing: Proceedings of ICT4SD 2022*; Springer: 2022, pp 635–643.

(8) Brownlee, J. CNN Long Short-Term Memory Networks, `https://machinelearningmastery.com/cnn-long-short-term-memory-networks/`, Accessed: 2023-03-21, 2019.

(9) Cabitza, F.; Campagner, A.; Balsano, C. *Annals of translational medicine* **2020**, *8*.

(10) Cabitza, F.; Campagner, A.; Ronzio, L.; Cameli, M.; Mandoli, G. E.; Pastore, M. C.; Sconfienza, L. M.; Folgado, D.; Barandas, M.; Gamboa, H. *Artificial Intelligence in Medicine* **2023**, *138*, 102506.

(11) Clinic, C. Heart Attack (Myocardial Infarction), `https://my.clevelandclinic.org/health/diseases/16818-heart-attack-myocardial-infarction`, Accessed: 2023-03-24, 2022.

(12) Clinic, M. Hypertrophic cardiomyopathy, `https://www.mayoclinic.org/diseases-conditions/hypertrophic-cardiomyopathy/symptoms-causes/syc-20350198`, Accessed: 2023-03-24, 2022.

(13) datagen Convolutional Neural Network: Benefits, Types, and Applications, `https://datagen.tech/guides/computer-vision/cnn-convolutional-neural-network/`, Accessed: 2023-02-15.

(14) De Montjoye, Y.-A.; Radaelli, L.; Singh, V. K.; Pentland, A. ". *Science* **2015**, *347*, 536–539.

(15) Doshi, K. Batch Norm Explained Visually — How it works, and why neural networks need it, `https://towardsdatascience.com/batch-norm-explained-visually-how-it-works-and-why-neural-networks-need-it-b18919692739`, Accessed: 2023-03-05, 2021.

(16) Goodman, B.; Flaxman, S. *AI magazine* **2017**, *38*, 50–57.

(17) Hicks, S. A.; Isaksen, J. L.; Thambawita, V.; Ghouse, J.; Ahlberg, G.; Linneberg, A.; Grarup, N.; Strümke, I.; Ellervik, C.; Olesen, M. S., et al. *Scientific reports* **2021**, *11*, 1–11.

(18) Hong, S.; Zhou, Y.; Shang, J.; Xiao, C.; Sun, J. *Computers in Biology and Medicine* **2020**, *122*, 103801.

(19) Ibrahim, L.; Mesinovic, M.; Yang, K.-W.; Eid, M. A. *Ieee Access* **2020**, *8*, 210410–210417.

(20) Jahmunah, V.; Ng, E.; Tan, R.-S.; Oh, S. L.; Acharya, U. R. *Computers in Biology and Medicine* **2022**, *146*, 105550.

(21) Juhl, C. R.; Miller, I.; Jemec, G.; Kanters, J.; Ellervik, C. *British Journal of Dermatology* **2018**, *178*, 222–228.

(22) Kaggle, `https://www.kaggle.com/search?q=ecg+classification+cnn`, Accessed: 2023-01-17.

(23) Kalita, D. An Overview of Deep Belief Network (DBN) in Deep Learning, `https://www.analyticsvidhya.com/blog/2022/03/an-overview-of-deep-belief-network-dbn-in-deep-learning/`, Accessed: 2023-03-21, 2022.

(24) Liu, X.; Wang, H.; Li, Z.; Qin, L. *Knowledge-Based Systems* **2021**, *227*, 107187.

(25) Loh, H. W.; Ooi, C. P.; Seoni, S.; Barua, P. D.; Molinari, F.; Acharya, U. R. *Computer Methods and Programs in Biomedicine* **2022**, 107161.

(26) Lu, L.; Zhu, T.; Ribeiro, A. H.; Clifton, L.; Zhao, E.; Ribeiro, A. L. P.; Zhang, Y.-T.; Clifton, D. A. *medRxiv* **2022**, 2022–11.

(27) Mandour, A. 2D Convolution in Python, `https://iq.opengenus.org/2d-convolution-in-python/`, Accessed: 2023-03-05.

(28) Memari, I. Precision, Recall, Accuracy, and F1 Score for Multi-Label Classification, `https://medium.com/synthesio-engineering/precision-accuracy-and-f1-score-for-multi-label-classification-34ac6bdfb404`, Accessed: 2023-03-13, 2021.

(29) Mishra, D. Applications of Recurrent Neural Networks (RNNs), `https://iq.opengenus.org/applications-of-rnn/`, Accessed: 2023-03-21.

(30) Mohneesh, S. Savitzky-Golay Filter for data Smoothing, `https://pub.towardsai.net/savitzky-golay-filter-for-data-smoothing-3b7c1c5e7f69`, Accessed: 2023-03-10, 2022.

(31) Murat, F.; Yildirim, O.; Talo, M.; Baloglu, U. B.; Demir, Y.; Acharya, U. R. *Computers in biology and medicine* **2020**, *120*, 103726.

(32) N. Fogoros, R. What Is an Electrocardiogram (EKG or ECG)?, `https://www.verywellhealth.com/electrocardiogram-ekg-ecg-1745304`, Accessed: 2023-02-23, 2022.

(33) Nabi, J. Recurrent Neural Networks (RNNs), `https://towardsdatascience.com/recurrent-neural-networks-rnns-3f06d7653a85`, Accessed: 2023-03-21, 2019.

(34) Neves, I.; Folgado, D.; Santos, S.; Barandas, M.; Campagner, A.; Ronzio, L.; Cabitza, F.; Gamboa, H. *Computers in Biology and Medicine* **2021**, *133*, 104393.

(35) NHS Electrocardiogram (ECG), `https://www.nhs.uk/conditions/electrocardiogram/`, Accessed: 2023-02-23, 2021.

(36) OpenAI, `https://openai.com/blog/chatgpt`, Accessed: 2023-01-17.

(37) Pesapane, F.; Volonté, C.; Codari, M.; Sardanelli, F. *Insights into imaging* **2018**, *9*, 745–753.

(38) Prutkin, J. ECG tutorial: ST and T wave changes, `https://www.uptodate.com/contents/ecg-tutorial-st-and-t-wave-changes`, Accessed: 2023-03-24, 2023.

(39) Reiff, D. Understand your Algorithm with Grad-CAM, `https://medium.com/golden-data/what-rights-related-to-automated-decision-making-do-individuals-have-under-eu-data-protection-law-76f70370fcd0`, Accessed: 2023-02-25, 2021.

(40) Saha, S. A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way, `https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53`, Accessed: 2023-02-15, 2018.

(41) scikit-learn sklearn.preprocessing.StandardScaler, `https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html`, Accessed: 2023-03-28.

(42) Service, E. P. R. The impact of the General Data Protection Regulation (GDPR) on artificial intelligence, `https://www.europarl.europa.eu/thinktank/en/document/EPRS_STU(2020)641530`, Accessed: 2023-02-25, 2020.

(43) Shashanka, M. What is a Pipeline in Machine Learning? How to create one?, `https://medium.com/analytics-vidhya/what-is-a-pipeline-in-machine-learning-how-to-create-one-bda91d0ceaca`, Accessed: 2023-03-13, 2019.

(44) Strodthoff, N.; Wagner, P.; Schaeffter, T.; Samek, W. *IEEE Journal of Biomedical and Health Informatics* **2020**, *25*, 1519–1528.

(45) Sullivan, T. Half of hospitals to adopt artificial intelligence within 5 years, `https://www.healthcareitnews.com/news/half-hospitals-adopt-artificial-intelligence-within-5-years`, Accessed: 2023-03-27, 2017.

(46) Thambawita, V.; Isaksen, J. L.; Hicks, S. A.; Ghouse, J.; Ahlberg, G.; Linneberg, A.; Grarup, N.; Ellervik, C.; Olesen, M. S.; Hansen, T., et al. *Scientific reports* **2021**, *11*, 1–8.

(47) Trevisan, V. Using SHAP Values to Explain How Your Machine Learning Model Works, `https://towardsdatascience.com/using-shap-values-to-explain-how-your-machine-learning-model-works-732b3f40e137`, Accessed: 2023-02-25, 2022.

(48) Tutorials, D. N. Dropout Layer in CNN, `https://dotnettutorials.net/lesson/dropout-layer-in-cnn/`, Accessed: 2023-03-11.

(49) Uddin, M. Z.; Soylu, A. *Scientific Reports* **2021**, *11*, 16455.

(50) Valohai What Is a Machine Learning Pipeline?, `https://valohai.com/machine-learning-pipeline/`, Accessed: 2023-03-13.

(51) Van de Leur, R. R.; Bos, M. N.; Taha, K.; Sammani, A.; Yeung, M. W.; van Duijvenboden, S.; Lambiase, P. D.; Hassink, R. J.; van der Harst, P.; Doevendans, P. A., et al. *European Heart Journal-Digital Health* **2022**, *3*, 390–404.

(52) Verma, S. Understanding 1D and 3D Convolution Neural Network | Keras, `https://towardsdatascience.com/understanding-1d-and-3d-convolution-neural-network-keras-9d8f76e29610`, Accessed: 2023-03-05, 2019.

(53) Voigt, P.; Von dem Bussche, A. *A Practical Guide, 1st Ed., Cham: Springer International Publishing* **2017**, *10*, 10–5555.

(54) Wagner, P.; Strodthoff, N.; Bousseljot, R.-D.; Kreiseler, D.; Lunze, F. I.; Samek, W.; Schaeffter, T. *Scientific data* **2020**, *7*, 1–15.

(55) WikiLectures Unipolar and bipolar biosignals, `https://www.wikilectures.eu/w/Unipolar_and_bipolar_biosignals`, Accessed: 2023-03-11, 2015.

(56) Yadav, H. Dropout in Neural Networks, `https://towardsdatascience.com/dropout-in-neural-networks-47a162d621d9`, Accessed: 2023-03-11, 2022.

(57) Yamashita, R.; Nishio, M.; Do, R. K. G.; Togashi, K. *Insights into imaging* **2018**, *9*, 611–629.

(58) Zech, J. R.; Badgeley, M. A.; Liu, M.; Costa, A. B.; Titano, J. J.; Oermann, E. K. *PLoS medicine* **2018**, *15*, e1002683.

(59) Zhou, L. How to Build a Better Machine Learning Pipeline, `https://www.datanami.com/2018/09/05/how-to-build-a-better-machine-learning-pipeline/`, Accessed: 2023-03-13, 2018.

Appendix A

Code

GitHub repository containing the code for the pipeline, models and data preparation can be found at: https://github.com/AwaisHameed/cnnPTBXL