Is Attention Really All You need?

-

Study of a Transformer-Inspired Data-Driven Diagnostic Algorithm for Automatic Detection of Cardiac Arrhythmia in 12-Lead ECG-Signals

A thesis submitted to the University of Manchester
for the degree of
Master of Philosophy
in the
Faculty of Science and Engineering

Tom D. Denker

2022

Faculty of Science and Engineering, School of Engineering, Department of Physics and Astronomy

Contents

List of Figures

List of Tables

Abbreviations

AF	Atrial Flutter
AFIB	Atrial Fribrillation
ANN	Artificial Neural Network
CAD	Computer Aided Diagnostics
CNN	Convolutional Neural Network
DNN	Deep Neural Network
ECG	Electrocardiogram
FCN	Fully Connected Network
FN	False negatives
FNR	False Negative Rate
FP	False Positives
FPR	False Positive Rate
GD	Gradient Descent
HP	Hyper Parameter
IID	Independent and Identically Distributed
LR	Learning Rate
LSTM	Long Short Term Memory Network
MLE	Maximum Likelihood Estimation
NLP	Natural Language Processing
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic
SB	Sinus Bradycardia
SGD	Stochastic Gradient Descent
SI	Sinus Irregularity
SR	Sinus Rhythm
ST	Sinus Tachycardia
SVT	Supraventricular Tachycardia
TN	True Negatives
TNR	True Negative Rate
TP	True Positives
TPR	True Positive Rate

Abstract

Classifiers based on a deep neural architecture inspired by the *transformer*, as proposed by Vaswani et al. (2017)[Advances in neural information processing systems (pp. 5998-6008)], were developed and studied with regards to their performance on the task of detecting different types of cardiac arrhythmia in 12-lead ECG signals. All classifiers were trained on the Shaoxin People's Hospital 12-lead ECG database and further evaluated on the PTB-XL database. Two different labelling regimes were employed. Performance on the Shaoxin test set ($96.5 \pm 0.3\%$ and $92.6 \pm 0.45\%$ overall accuracy) was found to slightly exceed that of a simple CNN benchmark ($95.9 \pm 0.4\%$ and $90.25 \pm 0.55\%$ overall accuracy) as well as the performance of the DNN-LSTM model proposed and reported by Yildirim et al. (2020) [Computer methods and programs in biomedicine, 197, 105740] (96.13% and 92.24% overall accuracy) for both labelling regimes. Evaluated on *synthetic sequences* of concatenated examples from the Shaoxin database, the proposed algorithm showed superior ability to generalise to longer sequences and mixed labels compared to the CNN-benchmark, although performance generally decreased with the the length of the sequences. Performance on the PTB-XL database was low and approaching random guessing, presumably due to a combined effect of poor mapping of the different labelling regimes, differences in the underlying populations, the presence of mixed labels, differences in signal quality and potentially unknown artifacts.

Declaration of Authorship

I, Tom Denker, declare that this thesis, titled "Is Attention Really All You need? - Study of a Transformer-Inspired Data-Driven Diagnostic Algorithm for Automatic Detection of Cardiac Arrhythmia in 12-Lead ECG-Signals" and the work presented in it are my own.

I confirm that:

- No portion of the work referred to in the thesis has been submitted.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all the main sources of help. Clearly attributed.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Copyright Statement

Acknowledgements

I want to wholeheartedly thank my supervisor, Prof. Henggui Zhang, for having maintained faith in me and my abilities and giving me the freedom and time to develop ideas while providing guidance and support whenever needed. Furthermore, I would like to thank my advisor, Dr. Thomas Waigh, for his guidance on time-management and our exchanges about *deep learning* architectures and their applications. I would also like to thank Sohail Rathore and Peter Hollowell for the warm welcome into the Biophysics group after our return to university in the aftermath of the Covid-19 pandemic, and especially Sohail for our joint contemplations on man, god and the history of the world (all the while running code in the background). Lastly, thanks to my family and those special people in my life who motivate me every day to try and become a better version of myself. Those concerned will know this is for them.

Chapter 1

Introduction

Medicine is a science of uncertainty and an art of probability.

- Sir William Osler

The Electrocardiogram (ECG) provides sequential data of the electric potential between various positions on the surface of the body, which is assumed to be primarily induced by cardiac electric activity and hence provides insight into the electrophysiological state of the heart as a function of time. In clinical practise, ECGs are commonly used to diagnose various forms or cardiac arrhythmia (Gertsch, 2008) as well as other cardiac conditions thanks to their inexpensive and non-invasive nature.

Historically, this information has been extracted manually by experienced and often specially trained clinicians through visual inspection of the raw signal, resulting in significant expenditure of expert time as well as the risk of human error in addition to the already significant risk of misclassification owed to the inherent ill-posedness of the problem itself (Kara, 2021).

Unsurprisingly, various attempts have been made to automate this rather tedious classification process - which, for an experienced clinician, is often merely an exercise in pattern recognition without the need for higher order reasoning (Gertsch, 2008) - in order to reduce costs and improve patient outcomes. Approaches ranging from rule-based *expert systems* to *deep learning* techniques

have been proposed and tested in recent decades, at times reaching super-human performance (Hannun et al., 2019); and some have even found their way into clinical practise through commercial applications (i.e. Turakhia et al., 2019). Despite this, there is yet much progress to be made with regards to the performance, range of applicability, safety, interpretability and generalisability of the proposed methods, as well as their ability to detect the limits of their own competence (see section 2.5.4 for a detailed discussion).

1.1 Aims and Objectives

Based on the considerations detailed in the following chapter, **the overall aim of this work is to scrutinize a *deep learning* algorithm based on the *transformer* architecture (Vaswani et al., 2017) for the automatic detection and classification of certain types of cardiac arrhythmia**. It is our belief that such an algorithm is not only uniquely appropriate to the task at hand, but also more flexible and able to generalise, both of which are crucial aspects of clinical implementation. Furthermore, we hypothesise a high degree of *interpretability* by clinical experts inherent in the proposed architecture due to the technical nature of the *attention mechanism* and the lack of recursion.

Consequently, the following **objectives** have been formulated:

1. Determine the effect of training data imbalance and pre-processing on the performance of the proposed architecture.
2. Determine the performance effects of modifying the proposed architecture to accept additional patient data as input and perform *multitask* classification.
3. Determine whether the proposed architecture can outperform its benchmarks in terms of out-of-sample performance.
4. Determine whether the proposed architecture generalises well to longer sequences than those seen during training (as compared to benchmarks).
5. Determine whether the proposed architecture generalises well to data drawn from other sources (as compared to benchmarks).
6. Determine the degree to which the proposed architecture is interpretable by clinical experts.

Chapter 2

Theoretical and Historical Background

2.1 The Physics of Electrocardiography

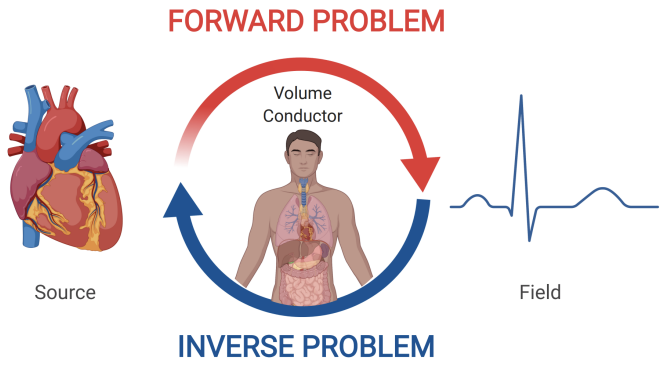


Figure 2.1: Illustration of the forward- and inverse problem of electrocardiography. From Kara (2021).

The human heart and torso can be modelled as a source of electric potential within a volume conductor, where the electric activity of the former induces a time-varying surface potential on the latter (Kara, 2021). Drawing conclusions about the state of the source based on measurements of the surface potential of the volume conductor is then what is known as solving the *inverse problem of electrocardiography* (compare fig 2.1), which in its broadest sense can be said to

be the underlying purpose behind all forms of ECG interpretation.

Despite the inherent ill-posedness of the problem, excaberated by the very low spatial resolution of a standard clinical 12-lead ECG, clinicians have historically used this technology with remarkable success and the ECG is arguably the most important and widely used physiological signal related to cardiac activity and health. As such, it provides insight into the electrophysiological activity of the heart and is used to diagnose a wide variety of pathologies including various types of cardiac arrhythmia (Gertsch, 2008).

The twelve leads of the standard clinical ECG are obtained using 10 electrodes arranged across the chest and limbs (see fig 2.2.). The right leg electrode exclusively serves as a ground wire, while the remaining nine electrodes are combined to produce the six chest and six limb leads[1]

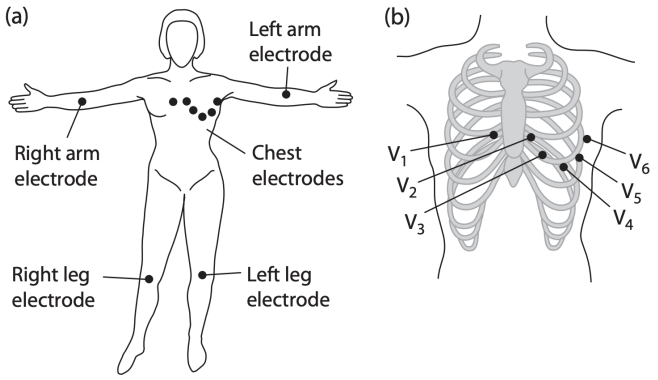


Figure 2.2: Schematic of positions of the nine 12-lead ECG electrodes on the human body. From Kusumoto and Bernath (2011).

The limb leads refer to various combinations of the limb electrodes (excluding the right leg), as illustrated in table 2.1 and figure 2.3. Their *vectors* (i.e. the direction in which changes in potential difference are detected) are lined up in the *frontal-* (or *coronal-*) *plane* of the heart (compare fig 2.4).

The chest leads V_1-V_6 (numbered from right to left) use their their respective electrode as a positive, and the sum of the limb leads as a negative

[1]It is important to stress in this context that a *lead* is a purely mathematical concept derived from the potential difference measured between two or more physical electrodes.

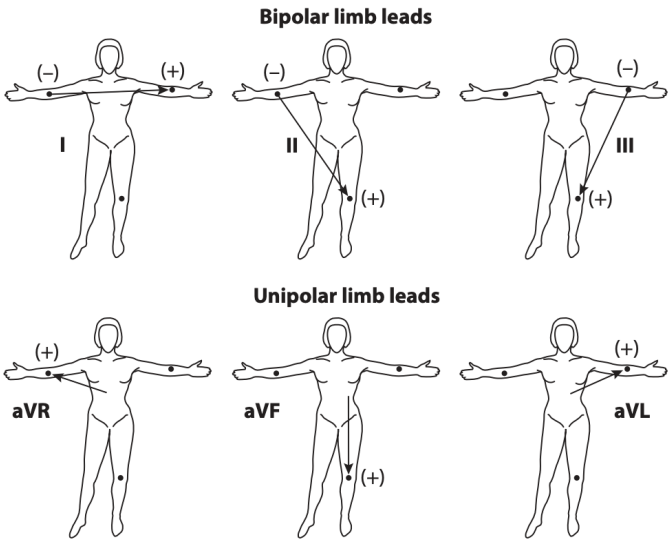


Figure 2.3: Schematic of the directions of view for the six limb leads of a 12-lead ECG.From Kusumoto and Bernath, 2011.

electrode. Since the latter is assumed to be at or near the centre of the heart (certainly within the *frontal plane*), their respective vectors are hence lined up in the *horizontal-* (or *transverse*) *plane* of the heart (compare figure 2.4b).

Lead Name	Positive Electrode (+)	Negative Electrode (-)
I	Left arm	Right arm
II	Left leg	Right arm
III	Left leg	Right arm
aVR	Right arm	Left arm + left leg
aVF	Left leg	Right arm + left arm
aVL	Left arm	Right arm + left leg

Table 2.1: Positions of positive and negative electrodes of the limb leads in a 12-lead ECG.

As a clinically optimal trade-off between sensitivity, specificty and feasibility (Rawshani, 2021), the 12-lead ECG hence provides a comprehensive (albeit spatially coarse) 360° view of the heart's electrical activity, and more specifically a means of observing the waves of de- and repolarisation across the heart against time in a non-invasive and relatively inexpensive way (see section 2.2.1

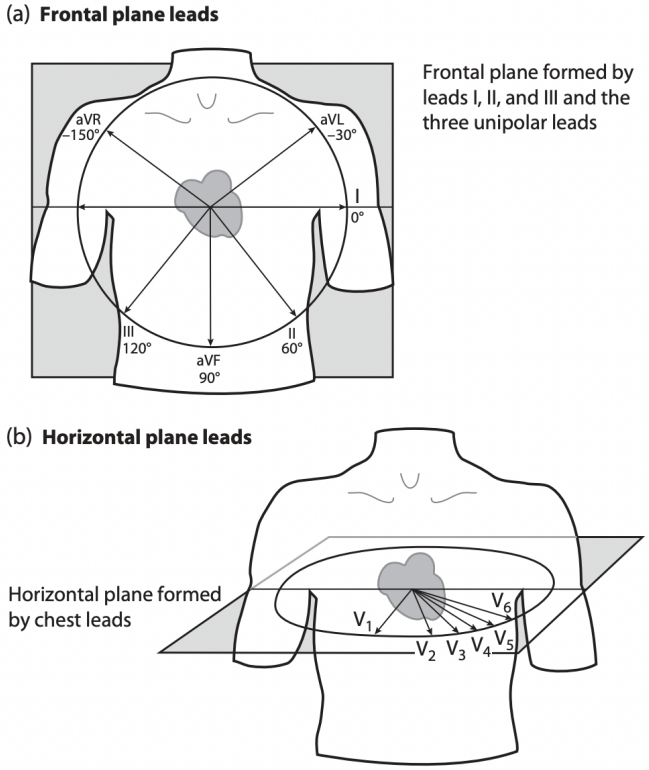


Figure 2.4: Schematic of the alignment of the six limb and six chest leads on the (a) frontal and (b) horizontal plane, respectively. From Kusumoto and Bernath (2011).

for more detail).

While all the data used in this work was obtained in digital form, in a clinical context ECGs are often presented on physical paper and in a characteristic format (figure 2.12 provides an example) where approximately one beat is displayed for each lead in addition to a *rhythm strip* (usually lead II) comprising multiple beats and allowing for the detection of less localised, rhythmic features (Kusumoto, 2020). An example of this form is displayed in figure 2.12, yet it is important to note that other forms of presentation are possible (i.e. figure 2.9).

2.2 Fundamentals of Cardiac Electrophysiology

2.2.1 Normal Cardiac Electrical Activity and the Normal ECG

The main function of the human heart is to maintain adequate blood circulation and thereby provide itself and all other organs with sufficient oxygen and other vital substances, as well as to ensure the disposal of waste and enable the gas exchange mechanism in the lungs (Tortora & Derrickson, 2018). As such, it is a vital organ whose inner workings hinge on complex electrophysiological and fluid mechanical dynamics. Specifically, in order to perform its life-preserving function, the heart must perform a perpetual series of orderly expansions and contractions referred to as *sinus rythm*[2].

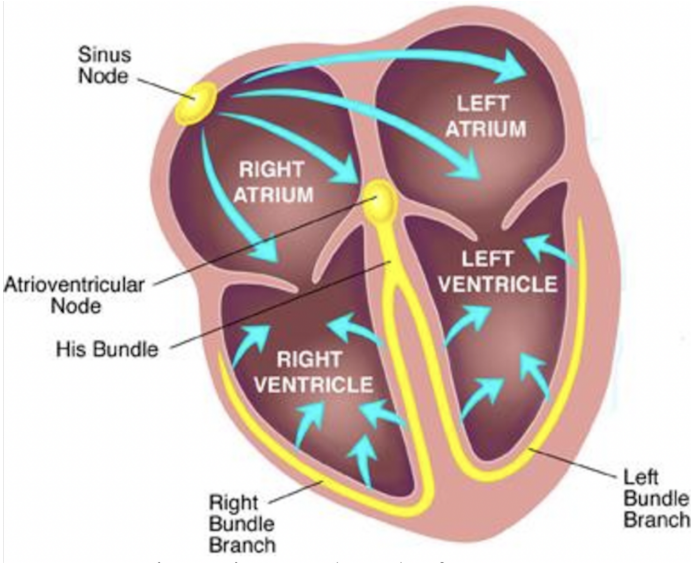


Figure 2.5: Simplified schematic of the normal cardiac conduction system. From Xing et al. (2014).

This motion is maintained through a periodically recurring series of electrochemical interactions between cardiac myocytes, resulting in a cascade of cellular depolarisations and repolarisations, causing contraction and subsequent expansion (resp). At rest, a negative potential of approximately -90 mV builds

[2]The term refers to the spatial origin of the electrical depolarisation of the cardiac muscle being at the sinus node (compare figure 2.5).

up inside the *cardiac myocytes* (heart muscle cells) relative to the extracellular space, mainly due to the combined activities of various protein pumps, exchangers [3], the "leaky" *inwardly rectifying current* I_{K1} which allows K^+ ions to flow out of the cell down their concentration gradient and thereby reach an overall electrochemical equilibrium, as well as the presence of large anionic protein molecules inside the cell that cannot cross the membrane (Kusumoto, 2020).

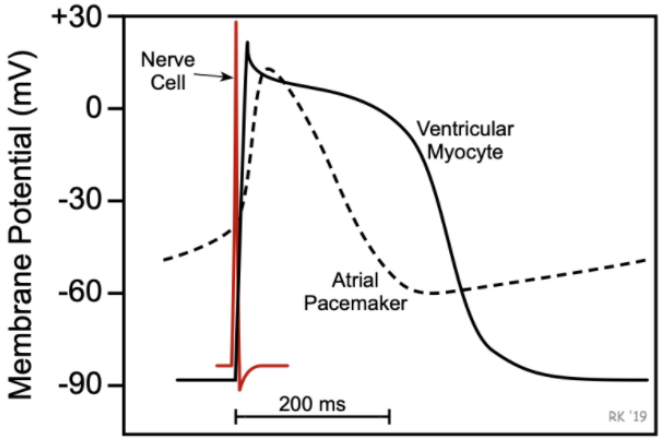


Figure 2.6: Schematic graphs of the fast and slow cardiac action potential as well as that of a nerve cell. By Richard E. Klabude, last accessed 28th December 2021 from https://www.cvphysiology.com/Arrhythmias/A010.

When the cell membrane of an atrial or ventricular cardiac myocyte is depolarised, either due to the prior depolarisation of neighbouring cells (as is the case during normal cardiac electrical activity) causing the outside potential to drop, or by an outside source of potential difference (such as a pacemaker), this leads to an opening of the hitherto closed NA^+ channels in the cell membrane and, as Na^+ ions flow down their electrochemical gradient, a spike in membrane potential up to approximately +10 mV occurs, corresponding to Phase 0 of the *cardiac action potential* (Kusumoto, 2020).

Closing of the NA^+ ion channel and brief opening of the *transient outward potassium channels* then facilitates Phase 1 (*early repolarisation*), which leads into the *plateau phase* (Phase 2), characterised by a stable membrane potential of approximately 0 mV. During this phase, the inward flow of positive charge in

[3]Namely the Na^+-K^+ ATPase and Ca^2+ ATPase protein pumps and Na^+-Ca^{2+} exchanger.

the form of Ca^2+ ions is balanced by the outward flow of K^+ ions, as enabled by the *L-type Calcium channel* and the *delayed rectifier potassium channels* respectively. Closing of the former then leads to a net outflow of positive charge and hence the onset of *rapid repolarisation phase* (Phase 3), during which the *delayed rectified channels* gradually closes and the "leaky" *inwardly rectifying current* takes over to facilitate *resting potential* (Phase 4).

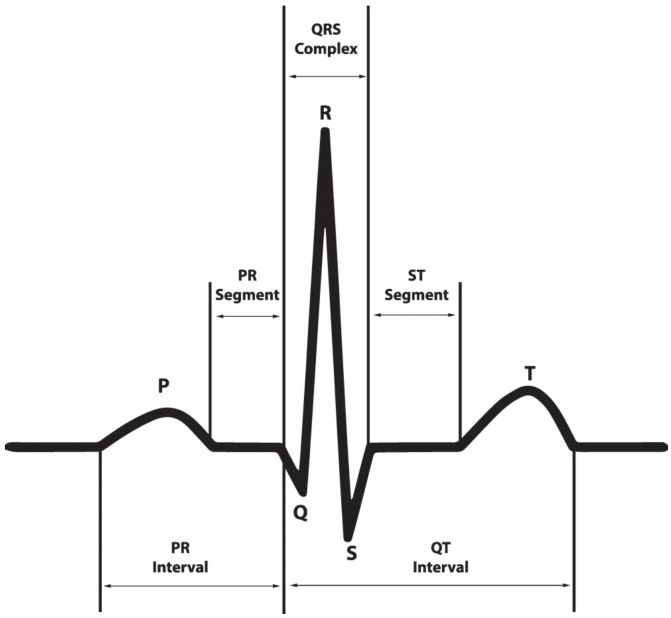


Figure 2.7: Schematic of a normal ECG and its components. From Zheng et al. (2020).

Cells in the *sinoatrial* and *atrioventricular* nodes exhibit a different type of de- and repolarisation behaviour, known as *slow response action potential*. Unlike working muscle cells, which exhibit *fast response action potential* as described above, these cells depolarise less rapidly, lack *plateau phase* and do not reach the maximum membrane potential of -90 mV. The electrophysiological behaviour is characterised by a different set of ion channels but similar in principle. Most importanly, these *slow response action potential* cells have the property of spontaneous depolarisation , making them act as natural *pacemaker cells*. Furthermore, they delay the pulse of depolarisation as initiated by the sinoatrial node as it reaches the atrioventricular node, thereby coordinating the sequences

of atrial and ventricular contraction, and also act as a buffer against overstimulation of the ventrices through rapid impulses caused by atrial arrythmia such as *atrial fibrillation* and *atrial flutter* (compare section 2.2.2). Figure 2.6 illustrates the temporal evolution of both fast- and slow-response action potential in contrast to that of a nerve cell.

The combined effects of de- and repolarisation of cells exhibiting fast and slow action potential in the atria, ventrices, the sinoatrial and atrioventricular nodes as well as the rest of the conduction system (compare figure 2.5) determines the shape of the normal ECG signal (see figures 2.7 and 2.8). In particular, the "smooth" shape of the p-wave is due to the contribution of *slow action potential* cells in the sinoatrial and atrioventricular nodes to the overall electrical signal (Kusumoto, 2020).

It is further important to note that the heart is not the only source of time-varying electric potential within the body and that besides the idiosyncratic noise of the ECG machine itself, the signals are obscured by constant noise, mainly from muscle contraction (i.e. Chowdhury et al., 2013). The issue of appropriate de-noising through pre-processing techniques is non-trivial and will repeatedly be touched on throughout this work.

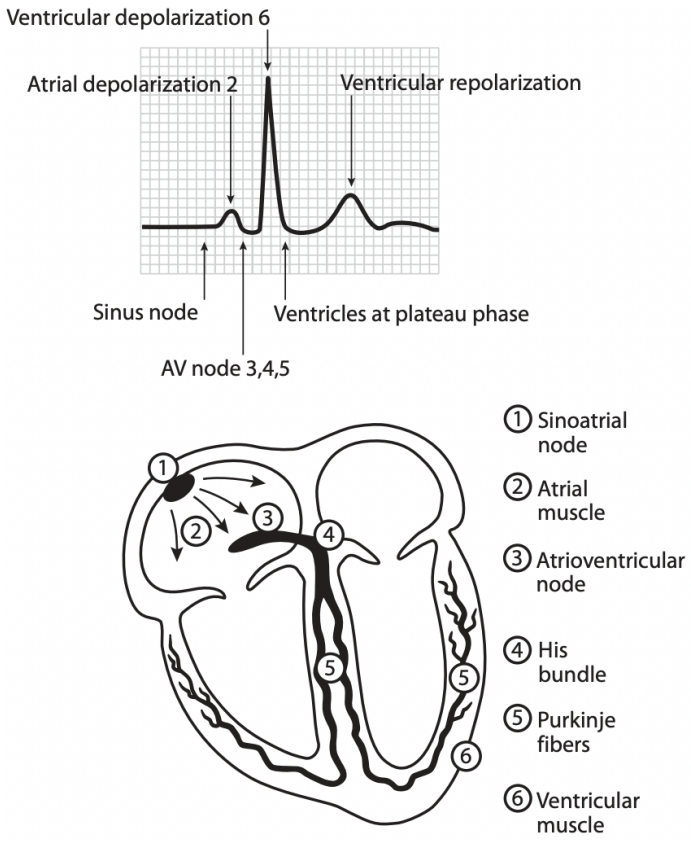


Figure 2.8: Illustration of the connection between the characteristic features of a normal ECG cycle and the cumulative effects of underlying cardiac electrodynamics. From Kusumoto and Bernath (2011).

2.2.2 Cardiac Arrythmia

As mentioned above, the presence of sinus rhythm can be considered a necessary (yet not sufficient) condition of normal cardiac electrical activity (Gertsch, 2004). The heart's failure to operate in this manner leads to a range of serious and in some cases life-threatening rhythmic abnormalities collectively known as *cardiac arrhythmia*.

Arrhythmia is present when the heart either beats too fast, too slow, or in an irregular manner (Kusumoto and Bernath, 2011). Common approaches to classifying the different known types of arrhythmia are to distinguish them by either their speed and rhythm characteristics, their anatomical origin or

by their underlying electrophysiological mechanism (Kusumoto and Bernath, 2011). Rhythm in particular can be classified as too slow (*bradycardia*) or too fast (*tachycardia*), anatomical origin as *ventricular* and *superventricular* (including *atrial* and *nodal*) and electrophysiological mechanism as either caused by abnormal impulse formation at a pacemaker site (*automaticity*) or by *re-entry* (Antzelevitch & Burashnikov, 2011).

Due to the constraints of the databases that were chosen for this study (compare chapter 4), we will limit our further discussion to the following common types of arrhythmia:

- Atrial Fibrillation - AFIB
- Atrial Flutter - AF
- Sinus Irregularity - SI
- Sinus Bradycardia - SB
- Supraventricular Tachycardia - SVT (including *sinus tachycardia* and excluding AF and AFIB)

Clinicians often identify different types of cardiac arrhythmia by recognising specific morphological and rhythmic patterns in a patient's ECG signal. The aim of this study is to propose and test a method of leveraging publicly available physician-annotated data and advances in machine learning technology in order to automate this process. In order to first provide an understanding of the task we want our model to perform, we will give a brief introduction yo the pathophysiology of the types of arrhythmia mentioned above, as well as the morphological and rhythmic intricasies of their associated ECGs.

Atrial Fibrillation (AFIB)

Atrial fibrillation (AFIB) is one of the most common forms of cardiac arrhythmia characterised by rapid electrical discharges caused by defects in the conductive properties of the atrial tissue (Waks & Josephson, 2014). While not immediately life-threatening, the most important complication of AFIB is cerebral stroke (Gertsch, 2008).

Rhythmically, AFIB is characterised by a fast heart rate as well as a very irregular ventricular response to the highly irregular impulses from the atria,

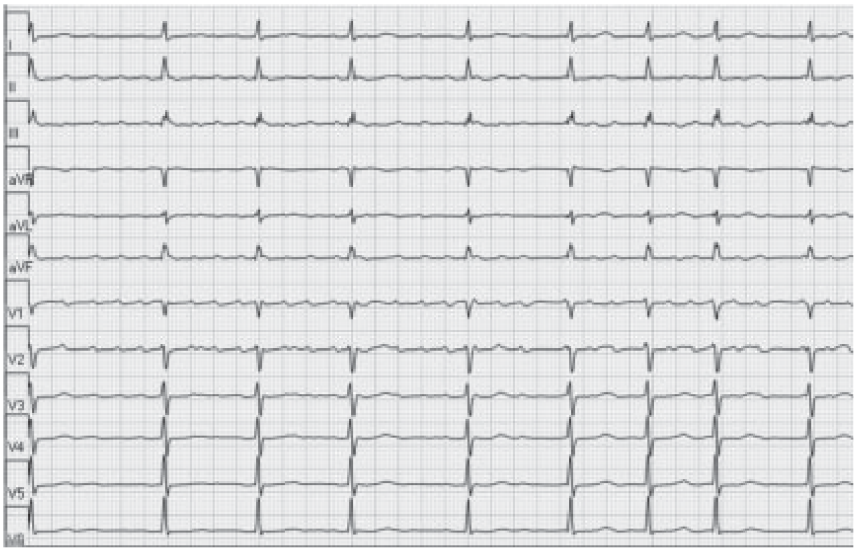


Figure 2.9: Example ECG of a patient with AFIB. From Kusumoto and Bernath (2011).

resulting in strong heart rate variabililty. This correlation is so strong that classifiers have been built solely based on RR vs. ΔRR plots (compare figure 2.7) in order to detect AFIB with some success (Lian et al., 2011).

Morphologically, AFIB is often characterised by missing p-waves due to the lack of coordinated atrial depolarisation (Ferguson et al., 2014) and sometimes by inverted t-waves (Kawaji et al., 2021). The presence of high-frequency f-waves (caused by the uncoordinated electrical activity of the atria) is a further diagnostic indicator (Gertsch, 2008). Figure 2.9 provides an example.

Atrial Flutter (AF)

While eight times rarer than AFIB, Atrial Flutter (AF) is still not an uncommon arrhythmia, particularly in elderly patients (Gertsch, 2008). Like AFIB, it is characterised by abnormal atrial electrical activity and a faster than usual (atrial) rate. Unlike in AFIB, however, AF is caused by re-entry (Kusumoto, 2020) and the rate tends to be more stable or at least "regularly regular" in an individual patient (Gertsch, 2008). In fact, it is this rhythmic feature that is most commonly used by clinicians in order to distunguish between AF and AFIB (Kusumoto & Bernath, 2011).

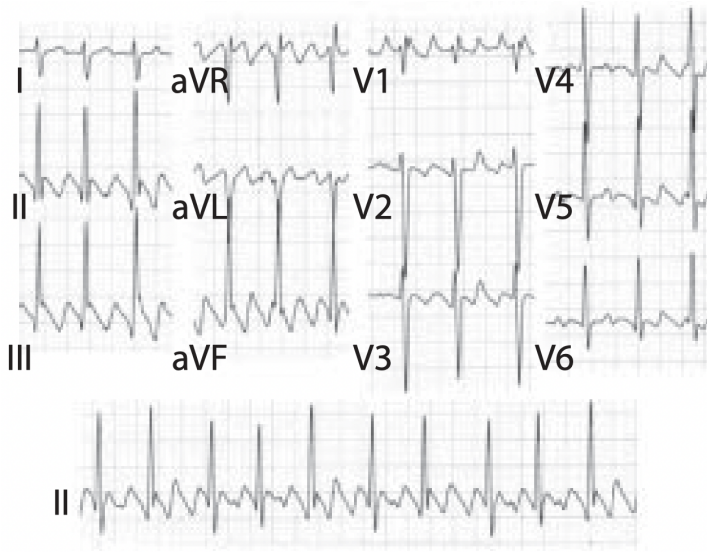


Figure 2.10: Example ECG of a patient with AF. From Kusumoto and Bernath (2011).

Morphologically, AF is characterised by *flutter waves* (F-waves) with a picket-fence-like appearance (Gertsch, 2008). Figure 2.10 provides an example.

Sinus Irregularity (SI)

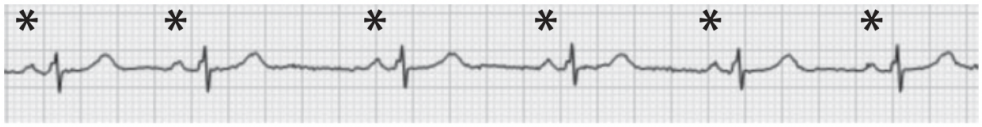


Figure 2.11: Example ECG of a patient with SI. The asterisks represent the locations of the p-waves. From Kusumoto and Bernath (2011).

Sinus irregularity (SI), also know as *sinus arrhythmia*, is a usually benign type of arrhythmia characterised by sinus rhythm with irregular spacing between the p-waves (representing irregular intervals between atrial depolarisations) and an otherwise normal ECG (Kusumoto & Bernath, 2011). Figure 2.11 provides an example.

Sinus Bradycardia (SB)

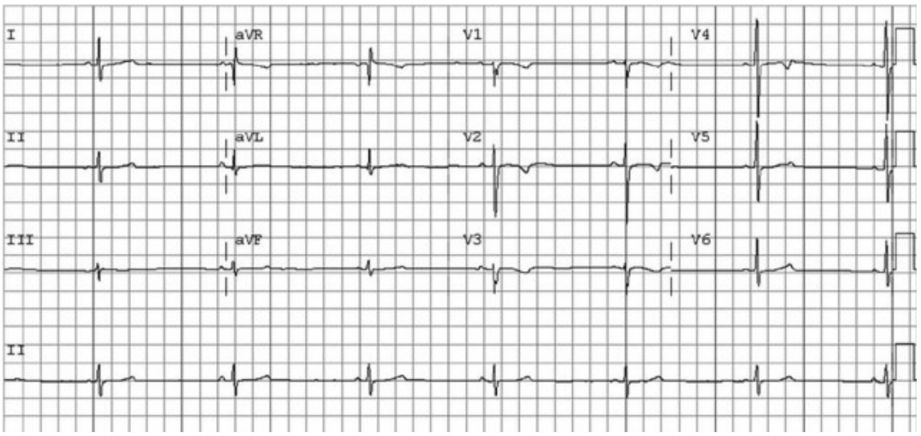


Figure 2.12: Example ECG of a patient with SB. From Drezner et al. (2013).

Sinus Bradycardia (SB) is a slow heart rate cause by a sinus node that is dysfunctional in fulfilling its pacemaker function, resulting in a rate of less than 60 beats per minute and an otherwise normal ECG (Gertsch, 2008). It is a condition commonly found in elderly people as well as well-conditioned athletes (Salyer, 2007, chapter 2). See figure 2.12 for an example ECG.

Supraventricular Tachycardia (SVT)

Supraventricular Tachycardia (SVT) is a fairly general term used for describing rapid heart rates (more than 100 bpm) that do not originate in the ventrices but instead have their site of abnormally rapid activity in either the atria or the nodes. They can hence be further distinguished by their anatomic origin (i.e. atrial or junctional) and their mechanism (*automatic* or *re-entrant*) (Kusumoto & Bernath, 2011). As opposed to ventricular tachycardias, supraventricular tachycardias usually have a narrow QRS-complex, and it is this morphological criterion that is often used to make the distinction (Kusumoto & Bernath, 2011). Figure 2.13 provides an example while figure 2.14 illustrates the simple taxonomy detailed above.

It should further be noted that SVT is sometimes used as an independent diagnostic class in addition to more specific classes such as atrial- or atrioventricula re-entrant tachycardia. This, however, is usually due to incomplete in-

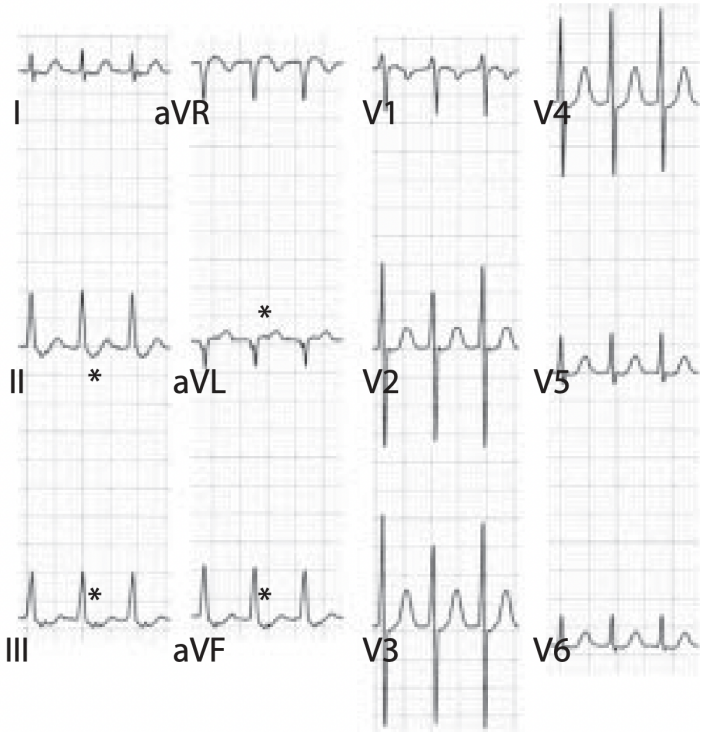


Figure 2.13: Example ECG of a patient with atrioventricular re-entrant tachycardia, a form of SVT. From Kusumoto and Bernath (2011).

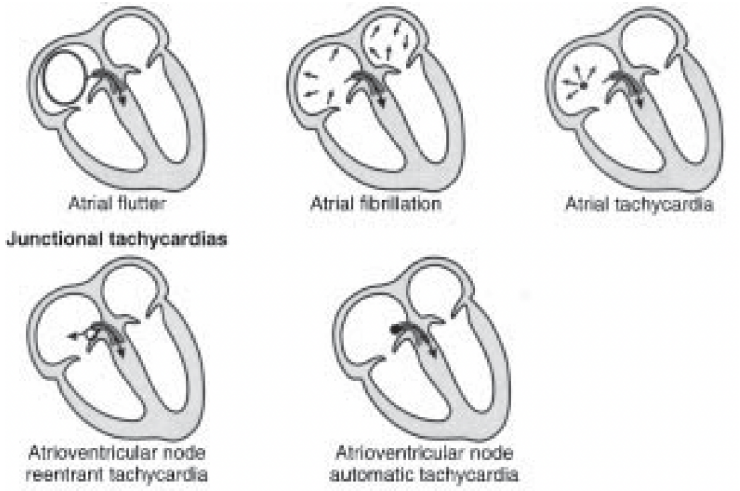


Figure 2.14: A simple taxonomy of supraventricular tachycardia. From Kusumoto and Bernath (2011).

formation and an inability of the diagnosing clinician to make a decision on the site and/or mechanism of the tachycardia (Zheng et al., 2020).

2.3 Historic and Current Approaches

When trying to provide a comprehensive overview of past and current computational approaches to solving the *inverse problem* in a broad sense as discussed in section 2.1, one first has to make a destinction between approaches revolving around first-principles-modelling and simulations based on a theoretical understanding of the electrodynamics and fluid mechanics of cardiac processes (i.e. along the lines of Clayton et al., 2011) and those that are based on either hard-coded rules of clinical decision making and/or statistical modelling.

While the former approach is clearly in a sense more elegant and shows much promise in fields like hypothetical drug testing (i.e. Whittaker, 2018) and improving our understanding of the pathophysiological mechanisms of cardiac conditions and how they relate to features of ECG and ECGI data (i.e. Kara, 2021), one could argue that such a high degree of physical detail and computational expense is not needed for most clinical classification tasks. In fact, most algorithmic approaches that are mainly concerned with diagnostic power as opposed to theoretical accuracy and insight into the inner workings of the heart therefore take the latter approach. That said, effective hybrid systems that combine machine learning and first-principles-modelling have been developed in recent years (i.e. Buerger et al., 2020).

The landscape of systems and approaches presented in literature for the automation of many ECG-related diagnostic tasks is so vast that in this section, only a selected sample can be presented in order to provide the reader with a non-comprehensive overview of the state of the field and the rationales which have lead to the development of the architecture proposed in this study.

As in other areas of what could be called *machine intelligence*, the first systems designed for the purpose of physiological signal classification and computer aided diagnosis (CAD) were so-called *expert systems*, based on hard-coded rules and/or methods from "traditional" statistics and signal processing (i.e. Ledos et al., 1988; Li et al., 1995).

More recently, with the advent of the strongly empirical approaches to computational predictive statistics hereinafter referred to as *machine learning*, a wide variety of data-driven approaches has been developed. The most prominent common characteristic of these approaches, which will be discussed in more detail in section 2.4, is that they do not rely on explicit (or *symbolic*) and subsequently hard-coded rules, but instead *learn* statistical patterns and decision boundaries directly from data. The necessary knowledge of human experts (the *ground truth*) is hence injected into the system through iterative parameter optimization ("training"), as opposed to hard-coded into the system in the form of first principles and/or clinical rules.

With regards to such approaches to time-series in general, Zhao et al. (2017) suggest a distinction between *model-based*, *distance-based* and *feature-based* approaches. In that order, these terms refer to approaches based on a clearly defined (generative) model of the data-generating process which is fit due the data available and then used to make predictions through some mode of comparison, approaches based on direct comparison of the input data with elements of the training data through some form of *distance measure* (also called *exemplar-based* models) and approaches based on the extraction of characteristic *features* from the data which are then used to draw (learned) class-distinctions, respectively.

The main short-coming of the *model-based* approach, which could also be deemed "traditional" time-series modelling, is clearly our lack of knowledge with regards to the data-generating processes of most real-world time-series (including ECGs), and/or their mathematical intractability (Zhao et al., 2017). Since ECG signals are dominated by determinstc, yet highly non-linear behaviour (Clifford et al., 2006), the modelling of such would ultimately take us down the path of first principle modelling described at the beginning of this section. Furthermore, many "general purpose" time-series models (such as ARIMA) were designed for signals dominated by *stochastic behaviour* and hence -amongst other issues- have strong stationarity requirements which ECG-signals and other physiological time-series in general do not meet (See section 2.5.2 for a discussion). Although there exists a number of empirical studies into this type of approach in the context of ECG-analysis (i.e. Vuksanovic and Alhamdi, 2013;Faal and Almasganj, 2021), it has been given no further consideration in

the development of the proposed algorithms due to the limitations described above.

Distance-based methods such as *k-Nearest-Neighbours* have also been investigated in the context of ECG-analysis (i.e. Faziludeen and Sankaran, 2016;Pandey et al., 2020) and shown to show satisfactory results on their respective tasks. However, by their very nature, the computational cost and memory usage of such methods tends to increase with the amount of available training data, raising concerns about scalability and efficiency.

To stay within the above taxonomy, the most prominent type of approach in contemporary literature is clearly the *feature based* approach, the aim of which is to design a *discriminative model* that draws decision boundaries between diagnostic classes based on relevant features. This is the domain of most (if not all) Machine Learning algorithms and will be the approach taken in this work.

2.3.1 Machine Learning Approaches in Literature

Various methods and architectures have been tried, including more "traditional" sequence-modelling methods such as and hidden markov models (see Gomes et al., 2009, for a comparison), *shallow* machine learning algorithms based on various characteristic features of the raw or transformed ECG data (i.e. Baydogan et al., 2013) and *deep learning* techniques (see below).

To make a further distinction, we note that most of the *shallow* architectures (such as *decision Trees* or *support vector machines*) by their very nature require extensive and elaborate pre-processing and *feature engineering*. This is due to inherent limitations in learning non-linear decision-boundaries from high-dimensional and strongly correlated input such as raw temporal and spatial data (see section 2.4.4 for a more detailed discussion). The process of generating a manageable number of reasonably uncorrelated input features from raw data, also known as *feature engineering*, is essential for the success of these methods and usually involves both experimentation and extensive domain knowledge (Goodfellow et al., 2016).

More recently, as a result of breakthroughs made in the field of *deep learning* (also historically called *connectionism*) after decades of fundamental research, a wealth of new approaches have been proposed based on Machine Learning models often called *artificial neural networks* (ANNs) or *deep neural networks* (DNNs) (see LeCun et al., 2015, for an introduction). The idea behind these models is to combine a relatively weak classifier, such as *logistic regression*, with a number of hierarchically organised *layers* of learned non-linear transformations, which essentially serve as a pre-processor extracting un-correlated features from raw-data (see section 2.4.4 for more detail). The main advantage of deep learning architectures is that they make very few assumptions about the underlying generative process (i.e. Zhao et al., 2017) and are able to learn their own predictive features from raw signal data, often resulting in higher accuracy and adaptability. Their drawbacks – the need for larger amounts of data and tendency to overfit compared to other methods that make stronger assumptions - have recently been mitigated through improvements in computing power, the capacity to generate and store large amounts of digital data and the development of powerful specialized learning algorithms and regularisation techniques (see section 2.4.5-2.4.7).

Approaches of this kind were presented by Hannun et al. (2019), He et al., 2019 and Zhang and Li (2021) amongst many others. While Hannun et al. (2019) use a fairly standard deep convolutional architecture and achieve superior performance on their proprietary data-set labelled by a panel of expert-cardiologists against a human benchmark, He et al. (2019) and Yildirim et al. (2020) employ a architectures based on a CNN-(bi-)LSTM network, which should in theory be able to learn both local morphological and global rhythmic features (compare sections 2.4.5 and 2.4.6). Some researchers also propose using these models in conjunction with pre-processing techniques such as *wavelet transforms* (Yildirim, 2018), as opposed to using the raw data as input to the model. Zhang and Li (2021) further employ a CNN-LSTM architecture combined with an attention mechanism which largely outperformes their respective benchmarks. The attention mechanism, developed by Bahdanau et al. (2014) and adapted by Luong et al. (2015) will also form an integral part of the algorithm proposed in this work (see section 2.4.7).

Furthermore, Natarajan et al. (2020) have proposed an architecture based on the *transormer* architecture for natural language processing as proposed by

Vaswani et al. (2017), which scored highest in the Physionet Challenge 2020 (Alday et al., 2020). While the author was unaware of their work at the time this study was conceived, ideas regarding design choices were nevertheless ultimately incorporated in this work (see section 3.1).

While all of the studies mentioned above report satisfactory results based on various performance metrics against their chosen benchmarks, it is crucial to keep in mind that these need to be interpreted in light of the available data, the type and quality of labelling, the type and difficulty of the classification task and the computational costs of the models used. There is therefore very limited utility in reporting figures such as overall accuracy and other metrics here without providing extensive context.

2.4 Fundamentals of Machine Learning

The aim of what is commonly known as (*supervised*) Machine Learning is to estimate some unknown quantity or category (also called a *label*), y, given a known vector of data-points (also called a *feature*), $\boldsymbol{x}$. In general, this relationship is not perfectly deterministic and hence takes the form

$$P(y|\boldsymbol{x}) = \frac{P(\boldsymbol{x}, y)}{\sum_i P(\boldsymbol{x}, y_i)}. \tag{2.1}$$

.

As stated in the previous section, what is unique about machine learning techniques is that, rather than trying to establish the given relationships through first principles (i.e. physics-based modelling) or hard-coded expert knowledge, they employ a combination of generic model architectures in conjunction with an appropriate *objective* or *cost function* and a corresponding *optimiser* in order to *learn*[4] the relationship incrementally from a set of training examples and corresponding labels (also known as the *ground truths*).

Mathematically, this is done by estimating a set of parameters, $\boldsymbol{\theta}$, that is assumed to parameterise the (probabilistic) relationship between $\boldsymbol{x}$ and y.

[4]We are aware that the use of the of the word "learning" in this context is an anthropomorphism whose appropriateness is an ongoing subject for debate, which is why we will hereinafter use the word in italics.

One way this can be done is by treating $\boldsymbol{\theta}$ as a random variable, formulating a prior belief about its distribution, $P(\boldsymbol{\theta})$, and then updating that belief based on the available data to obtain $P(\boldsymbol{\theta}|\text{Data})$. This is called the *bayesian* approach (Murphy, 2022).

An alternative approach is finding a single *point estimate* as a function of the data, $\hat{\boldsymbol{\theta}}$, called an *estimator*, that is optimal with regards to some pre-defined criterion, under the assumption that it represents the "best" approximation of the presumed "true" set of parameters, and hence

$$P(y|\boldsymbol{x}) \approx f_{\hat{\boldsymbol{\theta}}}(\boldsymbol{x}), \tag{2.2}$$

for some function $f_{\hat{\boldsymbol{\theta}}}$ defined by the model architecture and the *learned* parameters $\hat{\boldsymbol{\theta}}$.

This is known as the *frequentist* approach (Murphy, 2022). One very common *estimator* is the *maximum likelihood estimator* (MLE), i.e. the set of parameters that maximises the *likelihood* of the data, $P(Data|\boldsymbol{\theta})$.One well-known problem of this approach, particularly when using *maximum likelihood*, is known as *overfitting*. This will further be discussed in section 2.4.2.

While it has been argued (i.e. by Jaynes, 2003) that the *bayesian* view is more principled with regards to its mathematical foundation (the theory of probability), a philosophical discussion of the relative merits and underlying assumptions of both doctrines is beyond the scope of this work. It should be noted, however, that although we follow an approach more in line with the *frequentist* framework, this is solely for reasons of mathematical and computational feasibility, and in no way reflects our fundamental views on the nature of statistical inference.

2.4.1 Terminology

Please note that in the following, we will refer to a specific $\hat{\boldsymbol{\theta}}$, in conjunction with a defined computational structure in which to interpret it, as a *model*, the structure itself without the specific set of parameters as an *architecture*, and an *architecture* in conjunction with an *objective-* or *cost function* and an *optimiser* as a *learning algorithm*. A parameter that defines the architecture but are is

not *learned* from data is furthermore called a *hyperparameter* (HP). The act of *tuning* hyperparameters forms part of a meta-optimisation process also called *model selection* and will further be discussed in the next subsectio and well as section 5.6.

2.4.2 Model Complexity and Regularisation

Naturally, the (hypothetical) function describing the true relationship we are trying to model might be far more complex than the functions available to the model as defined by its architecture and the corresponding *hyperparameters* (the so-called *hypothesis space*). Such a situation would lead to what is known as *underfitting*. Converserly, the *hypothesis space* might also be too large with regards to the complexity of the relationship and/or there might not be sufficient incentive for the learning algorithm to choose simpler models over unnecessarily complex ones, causing the model to *learn* idiosyncratic the noise in the training data rather than (just) the relationship in question (Goodfellow et al., 2016). Such a situation is known as *overfitting*. While a full *bayesian* analysis, such as through what is known as *evidence approximation* (see for example Bishop & Nasrabadi, 2006, for a detailed discussion), has an inherent tendency towards favouring models of appropriate complexity and can hence be performed on the full training data, the guiding principle used to deal with these issues within the *frequentist* framework is the *bias-variance-tradeoff* (Dixon et al., 2020).

In practise, this is done through a form of re-sampling known as *cross-validation*, where the available data is separated into a *training* and a *validation* set. The so-called *generalisation gap* in performance on both sets of a model only trained on the *training* set is then closely monitored and viewed of a meta-optimisation problem in its own right. A large generalisation gap is indicative of overfitting (high variance), while a low gap paired with a high overall error is indicative of underfitting (high bias) (Goodfellow et al., 2016). While avoiding underfitting is necessary for the model make accurate predictions, avoiding overfitting ensures that what the model *learns* actually generalises to unseen data drawn from the same data-generating process or population and not just the idiosyncratic noise of the training set - it draws the distinction between mere optimisation and *learning*.

Most *regularisation* techniques are aimed at preventing *overfitting* through effectively limiting the models *effective capacity* (i.e. it's ability to fit the training data) and many can be interpreted as *bayesian priors* over the model parameters Goodfellow et al., 2016.

The process of optimising the *effective capacity* of a model (i.e. through choosing *hyper parameters* and employing *regularisation* techniques) is known as *model selection* and will be discussed in more detail in section 5.6.

2.4.3 Empirical Risk Minimization and (Stochastic) Gradient Descent

While some simple learning algorithms have closed-form solutions (i.e. the *normal equations* in the case of linear regression), most machine learning models rely on a technique called *gradient descent* (GD) in order to optimise their parameters incrementally by minimizing the *empirical risk* (i.e. the "error" between the data and the model predictions) as measured by some *loss function* (Murphy, 2022). To achieve this, the *loss function* is evaluated on the training data or a random subset, as in the context of *stochastic gradient descent* (SGD), and its gradient calculated with respect to the model parameters. Then, an incremental step, the *learning rate* (LR), is taken "down" the gradient by updating the parameters accordingly, leading to an incrementally more optimal solution.

For a given per-example loss function, $\mathcal{L}(\boldsymbol{x}_i, y_i, \boldsymbol{\theta})$ and corresponding *cost function*, $\boldsymbol{\nabla}_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$, representing the average loss over a set of m elements of the training data, the gradient is then given by

$$\boldsymbol{\nabla}_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) \approx \frac{1}{m} \sum_{i=1}^{m} \boldsymbol{\nabla}_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{x}_i, y_i, \boldsymbol{\theta}). \tag{2.3}$$

Many adaptations of the standard SGD-algorithm have been developed, including ADAM (Kingma & Ba, 2014) and its adaptations ADAGRAD and ADADELTA, which attempt to improve the *learning* process through improvements such as an adaptive learning rate.

For *deep neural networks* that consist of hierarchical layers, one needs to further *backpropagate* the scalar loss through the model in order to obtain en estimate of the gradient. The chain rule of differentiation is at the heart of

this process (since hierarchical models can naturally be viewed as composite functions) and for an in-depth discussion, we refer the reader to (LeCun et al., 2012).

2.4.4 Deep Learning

As mentioned in section 2.3.1, many machine learning algorithms struggle with unstructured and high dimensional input (such as raw spatial or temporal data) and are therefore not very well-suited for tasks such as computer-vision and signal-processing, at least not without significant *feature engineering*. This is known as the *curse of dimensionality* (Bishop & Nasrabadi, 2006).

A useful concept in this context is the so-called *manifold hypothesis*, i.e. the assumption that the statistical properties of many types of real-world data (such as images and sequences) do not fill the whole "space" of possible realisations but instead are arranged in ways that resemble much lower-dimensional *manifolds* of the space comprised by all possible degrees of freedom of any given input (Goodfellow et al., 2016).

Deep learning models exploit this hypothesis by employing a hierarchical set of *learned* non-linear transformations in order to project the input into a much lower dimensional and more easily separable *feature space*, within which predictions can then be made by relatively weak classifiers or regressors . Incidentally, this is also what *feature engineering* achieves manually for "shallow" architectures.

The most fundamental building block of deep learning models is sometimes called a *neuron*, and is essentially a unit that computes a weighted sum of all or some of the previous layer's outputs (and sometimes a bias term, b) based on learned weights w_i and then performs a fixed non-linear transformation (often a sigmoid or ReLu function) on the output:

$$f_{neuron} = \phi(\sum_i w_i x_i + b) \tag{2.4}$$

It can be shown that in the limit, even a single fully-connected (FC)[5]

[5]By *fully-connected*, we mean that all neurons of previous and subsequent layers are connected to each other.

hidden layer of such neurons can learn any possible input function. Therefore. such models are often considered *universal function approximators* (Hornik et al., 1989).

This expressiveness comes at a price, however, in the form of longer training times (often for many *epochs*, i.e. cycles through the whole training set) due to the potentially very high capacity that is hard to estimate due to the non-convexity of these models' cost functions (see Goodfellow et al., 2016, p.111 for a further discussion of this issue).

In sum, this results in the need for much larger data sets compared to *shallow* models and a correspondingly higher propensity to overfitting. Many regularisation techniques such as *weight-decay* (including but not limited to *Tikhonov regularisation*), *multitask learning* (i.e. letting the model perform multiple related tasks simultaneously) , *early stopping* (i.e. stopping training after a peak in performance on the test set has been reached) and *dropout* (i.e. "shutting down" some of the neurons randomly during training in order to avoid excessive co-dependence) have been devised in order to deal with this tendency (Goodfellow et al., 2016). While most of the named techniques will be revisited at some point throughout this work, *dropout* in particular has a very interesting *bayesian* interpretation that will be revisited in section 2.5.4 when discussing the aspect of *model risk* and its importance in the medical domain.

It must further be said that the tendency of ANNs to overfit and their resulting need for large amounts of data in order to prevent this[6], have recently been mitigated through improvements in computing power, the capability to generate and store large amounts of digital data as well as a more profound understanding of why and how they work at all(Valle-Perez et al., 2018)[7]. This has made deep learning techniques the method of choice for many researchers in the domain of ECG-analysis.

This tendency has further been supported by the development of powerful specialized learning algorithms. These are necessary since despite the promising

[6]The inverse relationship between the amount of available data and overfitting should be quite intuitive with regards to a notion of *degrees of freedom*. Goodfellow et al., 2016 provide a more rigorous discussion in light of *statistical learning theory*.

[7]Specifically, it is interesting to ask why they are able to derive at sensible solutions given their apparent overparameterisation.

results of Hornik et al. (1989), single-layer fully-connected networks are of little use due to their exploding computational and memory requirements. Furthermore, the so-called *no free lunch theorem* (Wolpert, 1996) states that there is no a-priori distinction between learning algorithms and that therefore, design choices that reliably lead to above-average performance must be made with a specific task in mind.

Two of the most frequently used specialised architectures for automatic ECG-interpretation as well as a third that is particularly relevant to this work will be discussed below.

2.4.5 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) and their defining layer architecture, the *convolutional layer* are particularly well adapted to the task of extracting features from large unstructured natural inputs such as images or sequences. Their strength lies in their ability to detect localised features (such as a line segment of certain shape) at any position in an image or sequence using the same set of weights (called a *filter* or *kernel*). They achieve this translation-invariability through *sliding* each *filter* across the image or sequence, thereby producing a *filter map* that is then passed on to the next layer. Mathematically, this operation is represented by a *discrete convolution*. For a one-dimensional time-series $x(t)$ and a kernel $w(t)$, the resulting *feature map* $f_{feature}$ is then

$$f_{feature}(t) = \sum_{a=a_{min}}^{a_{max}} x(a)w(t-a). \tag{2.5}$$

In order to detect localised features, filters usually have a certain *learned* shape within a narrowly defined region (called their *size*) and are zero elsewhere (Goodfellow et al., 2016).

In the standard CNN-architecture, blocks of convolutional layers combined with a non-linear activation function (usually the *ReLu* function) and downsampling (*pooling*) layers are then stacked, before the resulting feature maps are flattened out and fed into a FC-layer followed by a *sigmoid* function (for binary classification) which in turn produces a class label $\in [0, 1]$. While effective for short sequences and beat-wise classification (Acharya et al., 2017), it is clear

that this technique when used in isolation is fundamentally unable to capture the complex temporal relations (i.e. of large-scale rhythmic patterns) due to its fixed *field of vision* and, owed to its preservation of the (albeit often down-sampled) time-dimension, poses technical challenges when applied to sequences of variable length. For an in-depth discussion, we refer the reader to LeCun et al., 1989.

2.4.6 Long Short Term Memory Networks

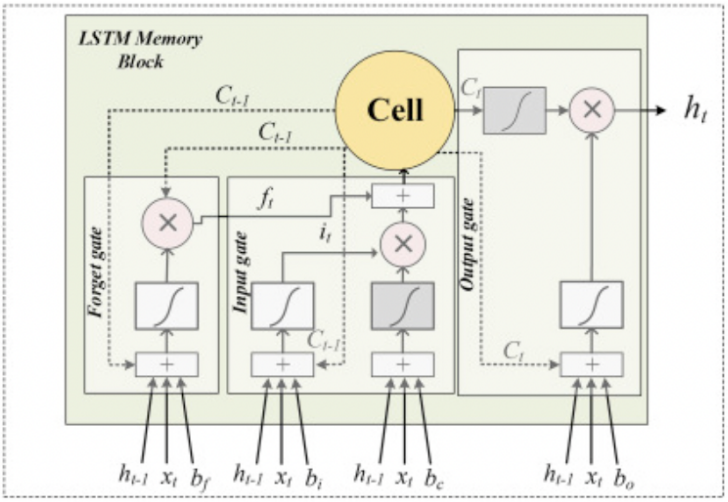


Figure 2.15: Schematic of a single LSTM-neuron, where h_{t-1} and x_t represents the previous hidden state and the current input respectively, while b_f,b_i,b_c and b_0 are bias terms associated with the respective gates. From Yildirim (2018).

While CNNs are very good at extracting relevant localised features from time-series or images, they are ultimately ignorant of the concept of *time* (or *sequentiality*), which has in part led to the development of *Long Short Term Memory Networks* (Hochreiter & Schmidhuber, 1997). The reason they will briefly be discussed here is that they form an essential part of many state-of-the-art classification algorithms for types of cardiac arrhythmia, including the one proposed by (Yildirim et al., 2020).

Like their conceptual forebarers, *recurrent neural networks* (RNNs) (Rumelhart et al., 1986), LSTMs introduce the notion of *before* and *after* by sequentially

processing the input features along the time-domain together with a *hidden state* stored from the previous step of the iteration (Hochreiter & Schmidhuber, 1997). That said, they represent an improvement over RNNs both because they are less prone to the so-called *vanishing-* or *exploding gradient problem* through the use of learned *gates*, and because these *gates* make the time-distributed structure more adaptive to varying statistical properties of the input data[8] (see figure 2.15 for a schematic illustration of a single LSTM-neuron).

Without going unnecessarily into the technical details of LSTMs, we can immediately notice a potential shortcoming in their use of sequential processing: Even if the input sequences are very long and in large parts irrelevant to the task, the model still has to iterate through the whole sequence while storing all relevant information in its hidden states. Besides being computationally wasteful and and impediment to parallelisation, as Vaswani et al. (2017) have pointed out, in practise this has also been found to diminish performance in very long sequences, i.e. those encountered in *Natural Language Processing* (NLP).

In the domain of ECG-analysis, LSTMs are often combined with stacked *convolutional* layers that serve as pre-processors, in ab attempt to use the former to extract "global" features such as those concerning rhythmic patterns form the more localised feature maps of the latter (i.e. He et al., 2019;Yildirim et al., 2020;Zhang and Li, 2021).

2.4.7 Scaled-Dot-Product Attention and the Transformer Architecture

As stated in the previous section, a well-studied issue of RNNs that still remains with LSTMs under certain circumstances is the *vanishing-/exploding gradient problem*, meaning the phenomenon of the loss gradient becoming very small/very large throughout the process of backpropagation [9], thereby rendering learning effectively impossible (Goodfellow et al., 2016). The problem is exacerbated when attempting to process long sequences with an abundance of long-term dependencies relevant to the given task, as, due to nature of recursion, information

[8] In fact, Dixon et al., 2020 have argued that "traditional" RNNs make assumptions similar to those of an ARIMA-process, making them fundamentally unsuitable for non-stationary data.

[9] for a detailed technical discussion of gradient descent and backpropagation, see LeCun et al., 2012

has to travel through an increasingly long computational graph as the temporal distance between two relevant data points increases (Vaswani et al., 2017).

One significant development to overcome these issues has been developed in the field of natural language processing (NLP), more specifically the area of *machine translation*. A common type of architecture in this domain has long been the *encoder-decoder architecture*, based on two serially interconnected LSTMs, called *encoder* and *decoder* respectively. As the name suggests, the *encoder* encodes the input - i.e. a sentence in language A - by turning it into a *feature vector*. This vector is then used as a hidden state to condition the second LSTM, which autoregressively[10] produces (*decodes*) an output in the form of a sentence in language B. To deal with the problem of increasingly long computational graphs between relevant parts of the two sequences, in 2014, Bahdanau et al. proposed a new mechanism that allows the decoder to "decide" on the most relevant parts of the input through a *learned* weighted sum over the encoder outputs, provided by an auxillary *alignment model* (or *attention layer*). This not only solves some of the technical issues to be discussed below, but also is much more closely in keeping with how humans are presumed to process verbal information (this aspect will further be discussed in section 2.5.3).

In 2017, Vaswani et al. proposed a new type of architecture for neural machine translation subsequently named *transformer*. The defining feature of this architecture is that unlike most comparable *encoder-decoder* type models, it does not use LSTMs or any other type of recurrent neural network, hence eliminating the technical limitations discussed above.

Instead, the *transformer* relies solely on the *attention mechanism* as modified by Luong et al. (2015) using computationally efficient dot products [11]. Vaswani et al. (2017) argue that the *attention mechanism* in itself is sufficient for the model to learn all relevant dependencies in the data and do so in a more effective and efficient way, without the need for recursion. The model has since been adapted many times and forms the basis of state-of the-art machine translation models such as BERT (Devlin et al., 2018) and roBERTa (Liu et al., 2019), as well as a burgeoning class of language-based *general intelligence* models such as GPT-2 (Radford et al., 2019) and its successor, GPT-3 (Brown

[10] This means that the output at each time-step is used as an input to the next.
[11] Sometimes called "Luong attention" as opposed to "Bahdanau-attention".

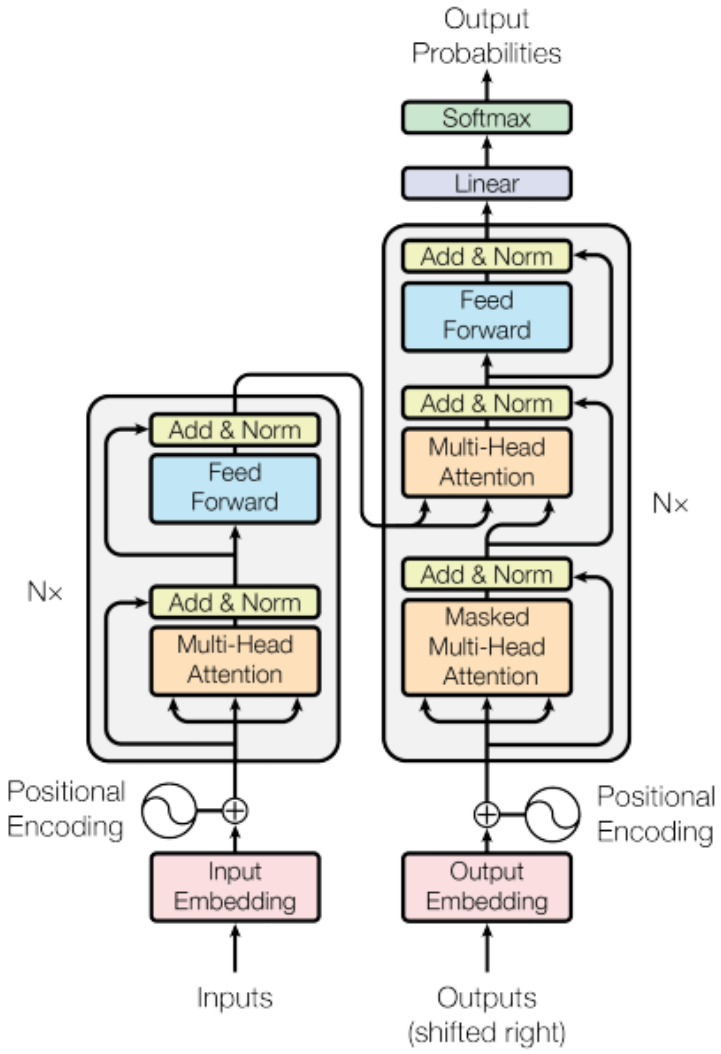


Figure 2.16: Illustration of the transformer architecture. From Vaswani et al. (2017).

et al., 2020). The architecture of the original *transformer* is illustrated in figure 2.16.

For reasons that will be explained in 3.1, we will limit our discussion to the *encoder* part of this architecture. The layer-type at the heart of it is the *multi-head attention* layer, which employs *scaled-dot-product attention* (or

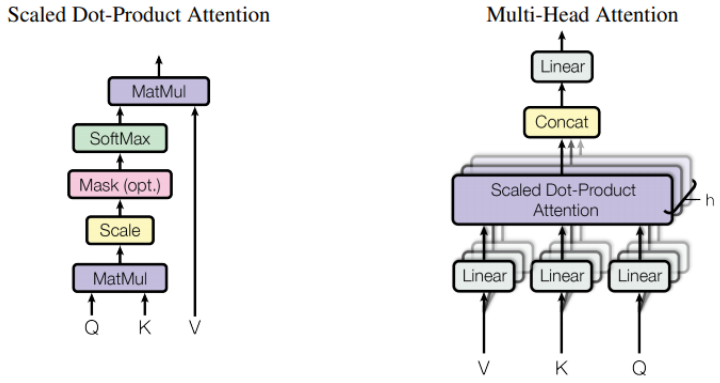


Figure 2.17: Illustration of the Scaled-Dot-Product Attention mechanism (left) and Multi-Head Attention layer used in the Transformer Architecture (right). From Vaswani et al. (2017).

scaled luong attention). For each example in the batch, it takes three matrices as input, which, at least in case of the *encoder*, are all identical and of dimensions $L_s \times d_{model}$, where L_s is the sequence-length and d_{model} is the dimension of the embedding vector of each word in a sentence. Within each individual *head* $\mathcal{H}_i$, each of these input vectors then gets projected into a d_{keys}-dimensional subspace[12] through by set of learned matrices $W_i^Q, W_i^K, W_i^V \in \mathbb{R}^{d_{model} \times d_{keys}}$ unique to that head, resulting in the matrices Q (for *query*), K (for *key*) and V (for *value*) of dimensions $L_s \times d_{keys}$ (Vaswani et al., 2017). The *scaled-dot-product-attention* mechanism then performs an operation that is best described as a *differentiable dictionary lookup* (Géron, 2019, p.559) with the standard dot-product as a similarity measure between *query* and *key* and given by equation 2.6.

$$Attention(Q, K, V) = softmax(\frac{Q^T K}{\sqrt{d_{keys}}})V \tag{2.6}$$

Trying to interpret this equation from the inside out (see figure 2.17 on the left for a computational graph), we notice that the expression $Q^T K$ produces a matrix of said similarity measures with dimensions $L_s \times L_s$. The *softmax* function is then applied row-wise, where

[12] d_{keys} is always smaller than or equal to d_{model} and often chosen so that $d_{model} = N_{heads} \times d_{keys}$, where N_{heads} is the number of attention heads.

$$softmax(\boldsymbol{x}) = \frac{e^{x_i}}{\sum_{j=0}^{L_s} e_j^x}. \tag{2.7}$$

The resulting matrix, still of dimensions $L_s \times L_s$ and now with rows normalised to 1, contains the so-called *attention-scores*. After applying the scaling factor, the final multiplication essentially "weights" each element of the sequence by the sum of its relevances (as measured by the similarity or dissimilarity of their respective *query* and *value* matrices) with regards to all the other elements of the sequence (including itself). It is important for understanding to note at this point that while in the encoder, the sequence is compared to itself (so-called *self-attention*), the Q, K and V matrices can nevertheless be expected to be very different from each other since they are the result of multiplication with the *learned* matrices W_i^Q,W_i^K and W_i^V and fulfill very different functions within the layer (compare equation 2.6).

After concatenating the output of each head and a final linear transformation (ensuring projection back in d_{model}-dimensional space regardless of the choice of d_{keys} and N_{heads}), the output of this layer once again has dimensions $L_s \times d_{model}$ (see figure 2.17). It is then added to the original input[13], the result of which is normed and used as input to a time-distributed FC-network of hidden dimension d_{ffn} (treated as a hyperparameter) and output dimension d_{model}, resulting in a final output dimension of the *encoder* of $L_s \times d_{model}$, which is identical to the dimensions of the original input.

While in the original transformer for machine translation, the output of the transformer is further processed as query-input to the *decoder's* multi-head-attention layer in order to ultimately perform machine translation (compare figure 2.16), it is sufficient for our discussion to note that -given the correct objective function- it is plausible to assume that the features obtained and stored in the output of the *encoder* contain highly relevant information regarding the relative importance of different parts of the sequence which could just as well be extracted for the purpose of solving other tasks, such as sequence classification. **This hypothesis forms the central rationale behind the architecture that is and tested in this work (see section 3.1).**

[13]To ensure better gradient-flow in a ResNet-type manner (compare He et al., 2016).

Positional Encoding

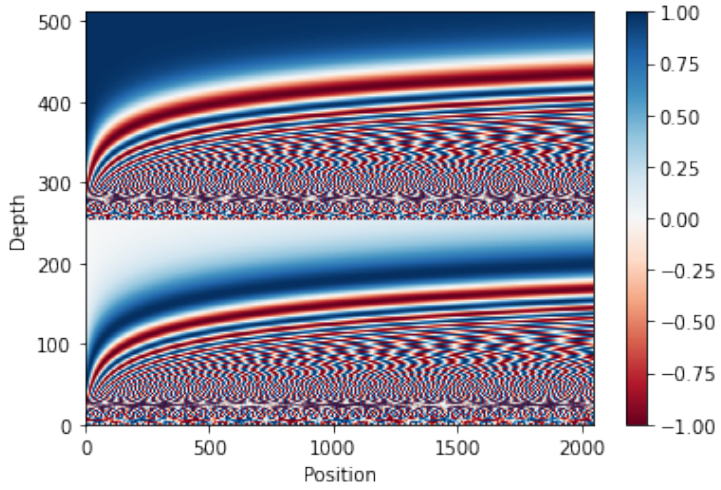


Figure 2.18: Visualisation of the positional encoding vectors in position-depth space based on the encoding method proposed by Vaswani et al. (2017).

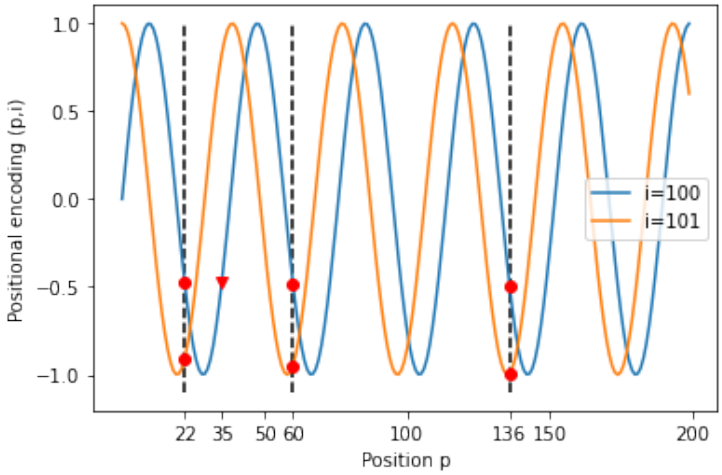


Figure 2.19: Illustration of how relative positional information is preserved by vectors of the positional encoding method proposed by Vaswani et al. (2017). Figure produced by the author, based on the work of Géron (2019).

Due to the lack of recursion and therefore the inherent inability of the model to learn a notion of absolute and relative temporal position (i.e. *here* and *there* as well as *before* and *after*), one needs to introduce a crucial step called *positional encoding*. On a high level, this comprises injecting positional information into the *embedding vector* of each word in the sequence by simply adding an *encoding vector* of the same dimension. This operation is represented by the plus sign in figure 2.16. As a result, the positional information becomes available in explicit form (i.e. as part of the vector representation of each word) and can hence be accessed by the attention mechanism without the need for recursion.

While these *encoding vectors*, not unlike the word embeddings used in NLP, could in principle be learned from data, Vaswani et al. (2017) instead propose the following equation for calculating the i-th element of the encoding at word position p:

$$P_{p,i} = \begin{cases} sin(\frac{p}{10000^{2i/d}}) & \text{for } i \in \{0, 2, 4, ...\} \\ cos(\frac{p}{10000^{2i/d}}) & \text{for } i \in \{1, 3, 5, ...\}, \end{cases} \tag{2.8}$$

.

where p is the position in the sequence, $i \in \{1, 2, 3, ..., d_{model}\}$ refers to the i-th element of the embedding vector and 10000 is a number arbitrarily chosen as the maximum possible sequence length before the resulting pattern repeats itself.

In order to understand the merits of this method, it helps to visualise the resulting encoding values in p-i-space (i.e. position-depth-space), which has been done in fig 2.18. We can immediately see that the resulting vectors are position-wise unique, hence each absolute position in the sentence (or other type of sequence) is represented by a unique encoding vector. In figure 2.19, we can further see that the proposed method of encoding not only ensures uniqueness of absolute position but also preserves relative position through the periodicity of trigonometric functions.[14]

[14]For a more detailed yet fairly informal discussion of the merits of this method and its roots in the mathematics of unit spheres, we refer the reader to Rothman, 2021.

2.5 Guiding Principles and Observations

This section will discuss a number of observations about the nature of the task of arrhythmia classification and principles of medical machine learning that cummulatively served as "boundary conditions" for the design choices presented in section 3.1.

2.5.1 Machine Learning and the Nature of Clinical Expert Knowledge

As we have noted before, it is reasonable to assume that while clinicians in training often rely on explicit *symbolic* rules when first learning the skill of ECG interpretation [15], the process soon gets automated and a major part of the knowledge involved becomes implicit or *procedural* (Kusumoto, 2020;Gertsch, 2008).

More generally, it can be argued that medicine, as an ancient discipline predating the advent of modern natural science and due to the hyper complex nature of the domain itself, has always had a very strong experiential and intuitive quality [16]. Furthermore, it could be said that particularly those aspects of practise that heavily rely on the clinician's prior experience and judgement are best represented by the concept of *procedural knowledge* (Patel et al., 1999). If these assumptions were true, then pattern recognition tasks such as those involved in ECG-analysis would certainly fall in this category, and provide further evidence that machine learning techniques, which are particularly well suited to tasks of a more *intuitive* nature (Goodfellow et al., 2016), are in fact appropriate.

Furthermore, as Jaynes, 2003 points out, even *expert systems* in the medical domain ultimately have to formulate their predictions in the language of (conditional) probability due to the nature of the domain[17]. Therefore, once we start fitting (i.e. *learning*) the associated distributions using available data through methods such as MLE, we have arrived at a point where machine learning techniques appear to be the most "natural" approach.

[15] A prominent example would be the absence of a p-wave and presence of high-frequency, low-amplitude f-waves in it's place during Atrial Fibrillation.

[16] Commonly referred to as the *art* as opposed to the *science* of medicine

[17] Which, while being sufficiently orderly to lend itself to rational analysis, has a strong stochastic element to it.

A Word on Pattern Recognition

As Kusumoto and Bernath (2011) point out, advances in fundamental research have provided a better understanding of the underlying pathophysiological mechanisms of cardiac arrhythmia and, as a result, medical education increasingly focused on studying the characteristics of the associated ECGs in light of a first-principles-based understanding of these mechanisms, as opposed to the more ad-hoc *pattern recognition* approach. However, it is very plausible to assume that with growing experience, a physician's approach to reading ECGs becomes increasingly *intuitive* and based around a growing mental repository of characteristic ECGs with which the currently seen one is compared (Gertsch, 2008).

2.5.2 Statistical Properties of ECG Signals

Like most physiological signals, ECG signals are *non-stationary*, *non-linear* and *noisy* (i.e. Maji et al., 2020; Al-Shoshan and Al-Shoshan, 2019), ruling out many "traditional" time-series methods aimed at signals dominated by stochastic, rather than non-linear deterministic effects.

They also elicit a high degree of auto-correlation [18] which is supported by clinicians' use of characteristic morphological features when making diagnosis (compare section 2.2.2.), and suggests the use of methods that are able to automatically extract and exploit these features (such as *convolutional* layers).

Furthermore, as discussed in section 2.4.4, the temporal auto-correlation implies that the use of *shallow* methods must be accompanied by a significant degree of *feature engineering*, and that *deep learning* methods that learn uncorrelated features from raw data are likely to be appropriate.

2.5.3 On the Analogy between Natural Language Processing (NLP) and ECG-Classification

Now that the general case for the use of *deep learning* techniques is established, we will provide a rationale behind the more specific design choices made in the

[18] This must be true due to the deterministic and periodic nature of the signal.

proposed algorithm.

As hinted at in section 2.4.7, the fundamental hypothesis behind the architecture proposed in this work is that there is a fundamental commonality in how we as humans process language and other types of sequential information not through systematic recursive iteration and the cumulative storing of previous information in the form of *hidden states* (as in RNNs and LSTMs), but by considering a sequence largely as a whole and looking for meaningful connections that allow us to draw conclusions. While the plausibility of this assumption is fairly obvious with regards to language - after all, we can only understand a sentence once we have received it in full and by establishing meaning through our implicit understanding of grammar and syntax-, revisiting the patterns described in section 2.2.2. suggests that the same might by true for the task of detecting cardiac arrhythmia in ECG signals.

More specifically, one could hypothesize an analogy between certain *segments* of an ECG-signal and *words* in natural language. If that analogy held true, it would then be desirable to allow the model to learn meaningful relationships between different parts of the sequence and to develop a sense of their functional relationship (i.e. *verbs* our *nouns* in the case of language corresponding to *QRS-complexes* or *p-waves* in the case of ECGs). Incidentally, this is exactly what the *encoder* module of the *transformer* architecture does through its dictionary-lookup-type approach (compare section 2.4.7).

This hypothesis, if true, would warrant the conclusion that the *transformer* architecture not only provides a technical improvement through its increased parallelisation and support for long sequences, but also represents a fundamentally more appropriate way of processing information in the domain of classifying physiological signals in general and ECGs in particular with likely implications for the design of specialised architectures for use in clinical practise.

2.5.4 Specific Challenges and Desiderata in Medical Machine Learning

In addition to the above considerations with respect to the nature of the data as well as the task at hand, the design choices made in the development of the

proposed algorithm were further informed by a set of desiderata which, while not specific to ECG-analysis, are nevertheless crucial in the domain of medical diagnosis.

While *safety* is perhaps the most important general concern in all areas of *machine intelligence* (Amodei et al., 2016), it has very specific implications in the field of of medical diagnosis. Amongst these, two recent review papers authored by Parvaneh et al. (2019) and Antoniades et al. (2021) discuss current areas for improvement and further research based on their assessment of the state of the field. They agree that issues regarding *generalisability*, *interpretability*, the quantification of *model risk* and reporting of *computational efficieny* are amongst the most pressing. These will be discussed in some detail in the following, with a particular emphasis on the impact on design decisions involved in developing the algorithm proposed in this work.

Interpretability

A very common objection to the use of machine learning, particularly *deep learning*, in the medical domain is a lack of transparency with regards to how these models come to their conclusions. Due to this characteristic, they are often described as *black boxes* in contrast to models based on hard-coded rules and/or first principles which have been explicitely formulated and validated by their human creators (see Durán and Jongsma, 2021 for a detailed discussion).

While on a principal level, the issue is unavoidable due to the very nature of how these models obtain their "skill", and although a case could be made for assessing an algorithm based on its proven diagnostic prowess rather than an immaterial discussion on how this prowess was achieved[19], it might nevertheless be desirable to attempt to gain insight into the inner workings of a successful model, both to bolster our own confidence and to provide a further element of *meta-regularisation* and protection against overfitting.

Since lack of understanding of the inner workings of machine learning models is a general phenomenon, a range of *model-agnostic* methods - such as *fea-*

[19]This argument is especially compelling in light of the fact that humans can be said to be as prone -if not more- to error as machines and that from a lay person's point of view, a clinical expert is a kind of *black box* himself.

ture importance (Che et al., 2016) - have been devised to gain insight. However, these methods come with a range of challenges outlined by Ribeiro et al. (2016).

An alternative is to exploit modes of representation "native" to a particular type of architecture that happen to possess a certain degree of interpretability for humans. As an example, it often makes sense to visualise the *filter banks* learned by convolutional layers[20] and to track their activation across the domain of convolution. While this approach is taken by Selvaraju et al. (2017) in the development of *Grad-CAM*, and employed successfully by He et al. (2019) to validate their model, Xi and Panoutsos (2018) combine a convolutional architecture with *fuzzy logic* an a classification layer based on *radial basis functions* (as opposed to softmax or sigmoid) in order to make interpretable predictions.

Part of the rationale for the choice of the *transformer*-like architecture proposed in this work is the fact that it allows the attention-mechanism to act directly on the *embeddings* without any recursion that would effectively "collapse" the time-domain before the point where *attention* is employed. The hypothesis is that this could allow us to visualise the temporal relationships extracted by the *encoder* and subsequently used by the classifier to make predicions. We could then compare these preditions with our knowledge of the morphological and rhythmic patterns of ECG-signals associated with various types of arrhythmia in order to add an additional expert-driven dimension to the validation process. Section 6.8 explores this aspect as it relates to **objective 6**.

Furthermore, using well-justified design choices and careful feature selection based on our understanding of the domain and the nature of the data can be seen as way of adding interpretability and robustness in itself, and is an approach that has been attempted in this work.

Generalisability and Benchmarking

As Parvaneh et al. (2019) point out, many studies report their results only within the scope of one particular data set which has been resampled to be used for training, model selection and testing. The problem with this approach is that while the model might well have learned features that generalise well to data from other sources, there is no way of testing whether this the case without

[20] Incidentally, this idea also forms the basis of the burgeoning art form of *deep dreaming*.

subsequently testing it on outside databases.

Furthermore, the nature of the task and hence the degree of difficulty and achievable performance varies largely depending on the training data, the quality, strength, balance and diversity of the labels as well as other factors. Therefore, it is imperative to compare the performance of any learning algorithm to a benchmark which, if not evaluated first-hand under identical conditions, should at the very least have been tested on the same dataset using identical labelling.

In this work, the proposed algorithm is consequently evaluated against the performance of a common-sense benchmark designed to allow conclusions on marginal performance gains of employing a transformer-type *encoder* module as well as against a third-party benchmark provided by Yildirim et al. (2020), trained on the same data. This coincides with **objective 3**. Furthermore, models based on the proposed architecture were scrutinised regarding their ability to *generalise* to examples from a different database as well as to longer sequences than those seen during training (**objective 4** and **5**.

Confidence Estimates, Model Risk and Adversarial Examples

One of the inherent problems of non-bayesian machine learning classifiers such as those based on MLE is that they represent *point estimates* $\hat{\boldsymbol{\theta}}$ of the supposed "true" set of parameters $\boldsymbol{\theta}$. This implies that while their output can take the form of a probability-measure, they must be inherently ignorant of the uncertainty associated with their predictions as they relate to the choice of model parameters themselves..

Therefore, as Gal (2016) points out, the probability measure returned by classifiers based on logistic-/softmax regression, i.e. $P(y|\boldsymbol{x}, \hat{\boldsymbol{\theta}})$, is not an adequate comprehensive measure of the uncertainty associated with a model's predictions, especially when dealing with input examples that are far away from the training data by any meaningful measure of distance or not part of the theoretical population at all - also known as *adversarial examples*[21]. In anthropomorphic terms, a model based on such point estimates "does know what it

[21] A classic example would be a picture of a Chimpanzee fed to a *binary classifier* trained to distinguish between cats and dogs.

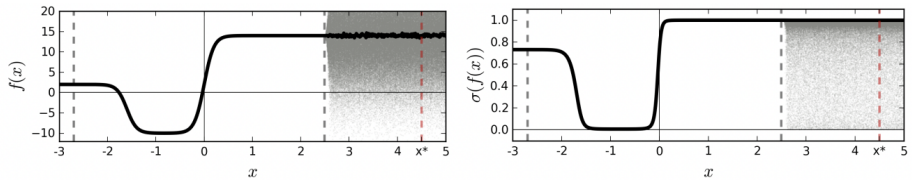


Figure 2.20: Illustration of the growing uncertainty (grey shading) associated with the output of an arbitrary *learned* function (left) and the corresponding decision function output (right), as predictions are made for examples "far away" from the training set. From Gal and Ghahramani (2016).

does not know" (or when it could likely be wrong), and hence, at inference, is unable to determine whether or not an example is drawn from outside its domain and will therefore likely result in nonsensical predictions [22]. A schematic illustration of this problem is displayed in figure 2.20.

Mathematically, what the predicted conditional sample distribution $P(y|\boldsymbol{x}, \hat{\boldsymbol{\theta}})$ fails to capture is the uncertainty associated with the parameter estimate itself or -in *bayesian* terms-, the full posterior distribution over $\boldsymbol{\theta}$ given the training data, i.e.

$$P(\boldsymbol{\theta}|\mathbf{X}_{train}) = \frac{P(\mathbf{X}_{train}|\boldsymbol{\theta})P(\boldsymbol{\theta})}{P(\mathbf{X}_{train})}. \tag{2.9}$$

The uncertainty in $\boldsymbol{\theta}$, expressed in the entropy of the distribution of $P(\boldsymbol{\theta}|\mathbf{X}_{train})$ (called the *posterior*), is also known as the *epistemic uncertainty* (or *model risk*) as opposed to the *aleatoric uncertainty* inherent in the non-deterministic nature of the prediction problem itself (i.e. $P(y|\boldsymbol{x})$). For a more detailed discussion of the different types of uncertainty affecting the predictions of *Machine Learning* models we refer the reader to Hüllermeier and Waegeman (2021) and/or Gal (2016).

In an effort to estimate the *model risk* associated with the predictions of *deep learning* classifiers, Gal and Ghahramani, 2016 have shown that *dropout*, originally devised as a regularisation technique to prevent excessive co-dependence between individual neurons (Srivastava et al., 2014), can be adapted in a straightforward way for this purpose. Based on the concept of *bayesian neural networks*

[22] Note that this problem is further excaberated by the use of the softmax activation function in *multiclass classification*, as the unitarity constraint essentially "forces" the model to make nonsensical predictions.

(i.e. networks whose weights take the form of random variables with distributions determined by *prior beliefs* as well as *evidence* in the form of data), they show that using *dropout* at inference over multiple iterations can be interpreted as *approximate bayesian inference* with a *gaussian prior*. The entropy of the resulting distribution over predictions can then be interpreted as a measure of *model confidence*. This method has been appropriately name *monte carlo dropout*.

However, as Lakshminarayanan et al. (2016) point out, in order for such an analysis to be truly bayesian, the *dropout rate* (i.e. the proportion of neurons to be shut down at each iteration) would have to to be *learnable* and change during training as the entropy of the posterior decreases. Instead, they propose a more frequentist interpretation of *monte carlo dropout* as a form of ensembling that allows one to assess model confidence by introducing randomness into the model and interpret the variability of predictions under such changes as a measure of *model risk*. They further argue that it is a common feature of ensembling methods to enable one to draw conclusions regarding *model risk* by looking at the statistical properties of the resulting set of predictions. There is also an ongoing debate on whether *batch normalisation* can serve a similar purpose (Mukhoti et al., 2020).

From a *frequentist* point of view, which assumes there exists one unique set of "true" model parameters, $\hat{\theta}$, to be estimated from data, uncertainty is commonly dealt with through re-sampling (or *bootstrapping*) the training data in order to assess uncertainty through variability in the estimated model parameters. Since in practise, the resulting distribution can be quite similar to the corresponding *bayesian posterior* (particularly in the limit of large training data sets and weak *prior* beliefs, $P(\boldsymbol{\theta})$, about the model parameters), Murphy (2022) refers to the distribution of model parameters resulting from bootstrapping as a "poor man's posterior". A corresponding ensemble method based on subsets of the training data drawn with replacement is known as *bagging* (i.e. Géron, 2019). For a further and more detailed discussion on various techniques of handling uncertainty in the context of *deep learning*, we refer the reader to Khosravi et al. (2011).

Within the scope of this work, we have adapted the view of Lakshminarayanan et al. (2016) insofar as that - barring a full bayesian analysis - some

form of *ensembling* is needed in order to assess and compare the performance of *deep learning* classifiers, especially when pushing them towards the edge of their ability to generalise. The method that has consequently been devised for reporting confidence intervals on the performance characteristics of different types of model architectures (see section 5.7) is based on this insight and, if only for pragmatic reasons, in keeping with the frequentist framework described above. Furthermore, while the classifiers involved in this study are neither based on ensembling (largely due to memory constraints) nor a full *bayesian* analysis or any computationally traceable approximation of such, the architectures have nevertheless been designed in such a way as to enable *monte carlo dropout* by including *dropout* layers between all layers containing learned weights.

Reporting Computational Efficiency

Parvaneh et al., 2019 further point out that detailed tracking and reporting of the computational efficiency and running time requirements of various algorithms is of great importance in light of their potential use in clinical and/or wearable devices with limited computational resources.

While limiting model-size served as a subordinate criterion of the model selection process (see section 6.1) and potential use in wearable devices has been anticipated through testing on longer sequences (compare section 6.6 and **objective 4**), the fact that the experiments for this work were conducted using the cloud-computing resources provided by Google Colab and are subject to changing allocations of hardware resources (including the model and make of GPU) made a consistent comparison of running time impossible.

Chapter 3

Model Architectures

This section will provide a detailed description of both the proposed architecture and the chosen benchmark architectures, as well as explain some of the rationales behind the design choices made.

3.1 The Proposed Architecture

As stated in section 1.1, the aim of this study is to investigate whether a *transformer*-inspired architecture is suitable for the task of detecting signs of cardiac arrhythmia in ECG signals. This problem is very different in nature to what the *transformer* architecture has originally been designed for, namely the auto-regressive sequence-to-sequence task of machine translation. In order to adapt the architecture accordingly, a few major changes had to be made.

Firstly, since the output we are looking for is a set of binary class labels containing (probabilistic) information about the presence of different types of cardiac arrhythmia rather than a sequence of word-embeddings based on the original sequence as well as previous outputs, the *decoder* module of the original architecture has been abandoned.

Secondly, since we are not dealing with discrete words but quasi-continuous electrical signals, we need to find an alternative way of creating a form of embedding that can serve as input to the *encoder* module. The most intuitive way to do this is to use a stack of 1D-Convolutional layers followed by max-pooling,

where the cross-sectional slice of all feature maps at each (downsampled) time-step can be interpreted as an embedding vector for that part of the original sequence. This builds on the analogy between *words* and *beats*, as discussed in section 2.5.3. *Batch normalisation* and *dropout* layers are further added to improve performance, prevent overfitting as well as for the purposes described in section 2.5.4. For an in-depth discussion of the benefits of adding these layer-types see Ioffe and Szegedy (2015) and Srivastava et al. (2014) respectively.

Thirdly, we need to note that one of the defining characteristics of the *encoder* and *decoder* modules of the original architecture is their preserving of dimensions, with both their input and output tensors being of shape [batch-size,sequence-length,d_{model}] where d_{model} is the dimension of the embedding vector (i.e. in our case the number of filters in the final CNN-layer). However, as we have discussed in section 2.4.7, each vector of the output time-series now contains information regarding all the other time-steps (this is what scaled-dot-product Attention does), and we can hence "collapse" the time-dimension by averaging over it using a *global average pooling* layer. The resulting tensor is of dimension [batch-size,d_{model}], allowing the transformer - at least from a purely technical perspective- to process variable-sized input sequences at test time. This is also in line with the approach taken by Natarajan et al. (2020).

Lastly, in order to make use of the additional data provided by the Shaoxin People's Hospital ECG Database (see section 4.1), we propose three possible modifications to increase performance through additional inputs and outputs. The rationale behind these modifications is to *condition* the output of the *encoder* module on patient-specific data that would be easy to acquire in a clinical setup (i.e. age, gender,...) and to improve performance and robustness by giving the model the additional task of predicting beat/condition labels (compare table 4.4) for each signal respectively. The theoretical basis for the latter as a regularisation technique has been described by Caruana (1997) and has been explored in various studies (**c**)itehusken2003recurrent, which have found that so-called *multitask* models can also lead to better results than an otherwise identical benchmark based on a single task.

Figure 3.1 illustrates the layer structure in the form of a flow chart and also states the output dimensions associated with each layer. For added detail, the embedding and encoder modules are illustrated in figure 3.2 and 3.3

respectively. The three possible modifications, resulting from the three possible combinations of using alternative input and/or output illustrated in figure 3.1 (dashed lines), will henceforth be referred to *multi-in*, *multi-out* and *multi-in/out*. The *positional encoding* is calculated in the same way as proposed by Vaswani et al. (2017) with a maximum sequence length of 10000.

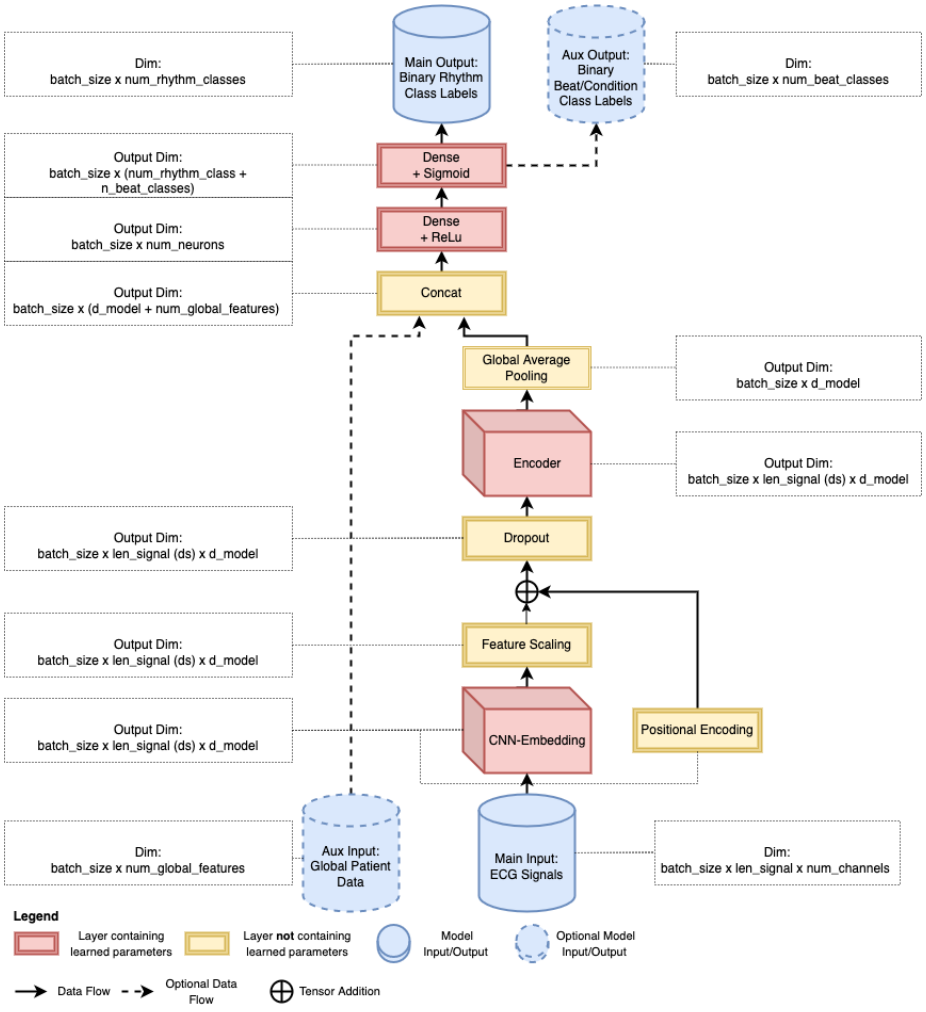


Figure 3.1: Diagram of the proposed architecture and it's potential modifications (dashed lines).

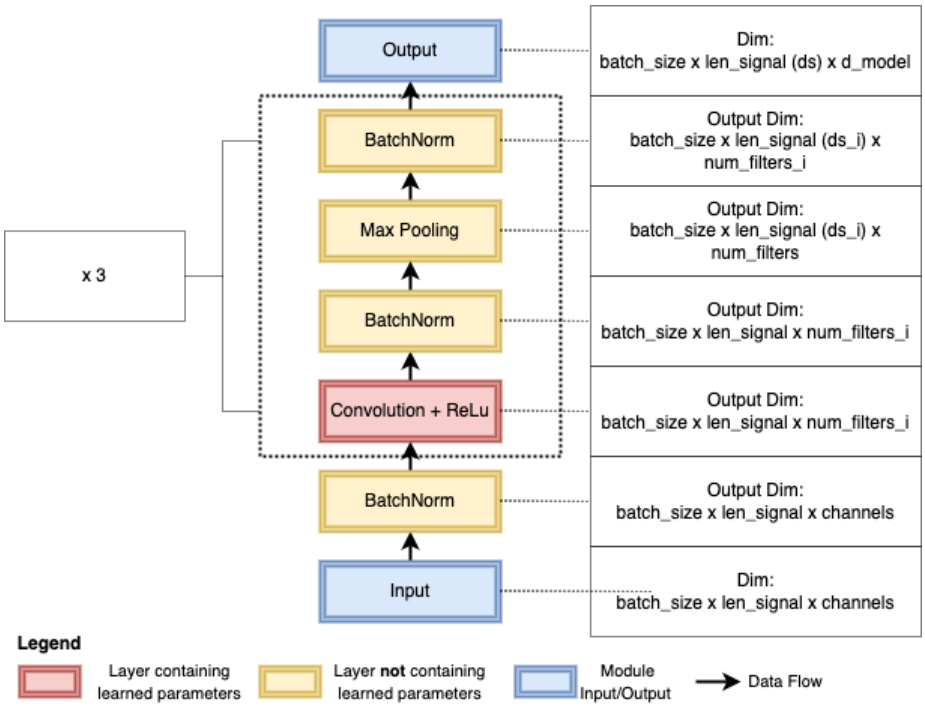


Figure 3.2: Diagram of *CNN-embedding* module.

3.1.1 Morphological feature extraction

Unlike in the original transformer model for NLP, ECG-data cannot be tokenised due to lack of a finite vocabulary. Instead, as mentioned above, a one-dimensional convolutional neural network is used to extract morphological features through a learned filter bank. This approach is in line with our favouring of feature-learning over the use of *hardcoded* morphological features, as discussed throughout chapter 2. The resulting feature maps have (approximately) beat-wise resolution[1] and the number of feature maps per time step is equal to the *model depth* of the transformer module.

3.1.2 Rhythmic feature extraction

In keeping with our natural language analogy, the extraction of *global* dependencies from the *localised* features of the CNN-embedding module is the domain of the *encoder* module. More specifically, the purpose of the multi-head attention

[1]That said, the downsampling factor has been treated as a hyper parameter during model selection, so other scales were tried but led to declining performance (compare section 6.1).

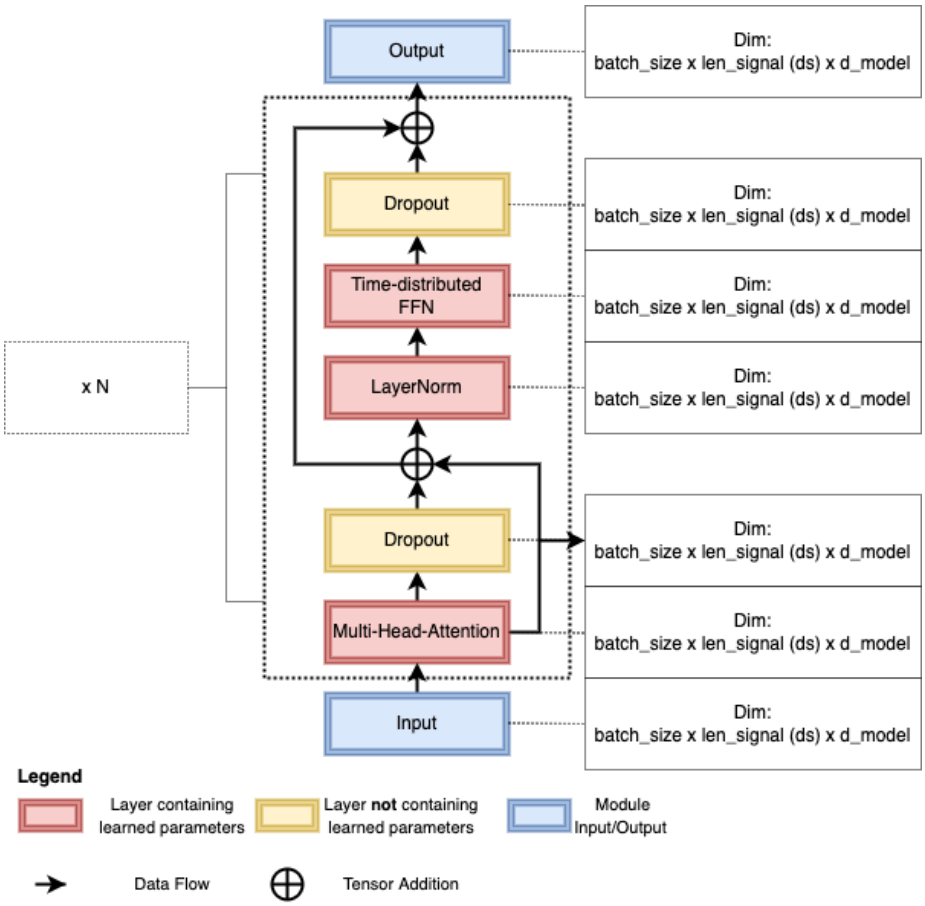


Figure 3.3: Diagram of the *encoder* module.

layers within the encoder stack is to "compare" the sequence to itself and establish relevant connections, which are then fed into the final classification layer.

3.1.3 Classification layer and loss function

Since the presence of individual classes of cardiac arrhythmia is not necessarily mutually exclusive (i.e. Asirvatham & Stevenson, 2016) and in order to mitigate the detrimental effect of the model encountering adversarial examples by giving the model more flexibility of outcome[2], the task was chosen to be presented in

[2]The idea is to not "force" the model to decide on a positive class when none is appropriate, thereby reducing systematic *model risk*

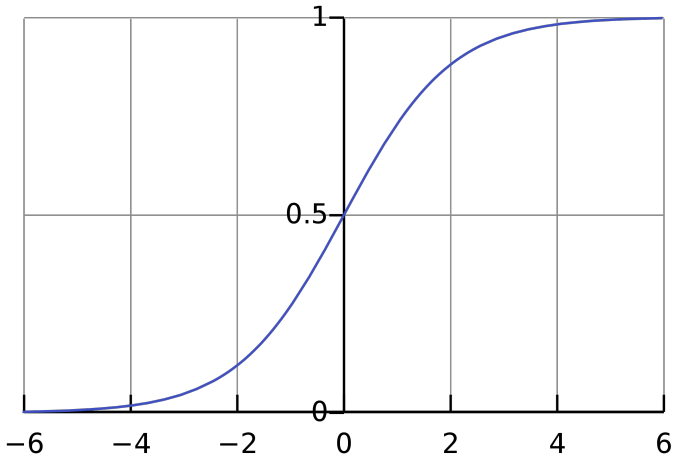


Figure 3.4: Graph of the sigmoid function. The horizontal grid line drawn at y=0.5 corresponds the standard decision boundary.

the form of *multi-label* classification.
Consequently, the activation function of the classification layer was chosen to be a *sigmoid* function rather than the *softmax* commonly used in *multi-class* classification tasks.

The functional equation of the *sigmoid* is as follows:

$$S(x) = \frac{1}{1 + e^{-x}} \tag{3.1}$$

One can easily see that $\lim_{x \to -\infty} S(x) = 0$ and $\lim_{x \to \infty} S(x) = 1$. Conceptually, the function hence bijectively projects the real line onto the $]-1, 1[$ interval, making the result interpretable in terms of a probability measure.

With four/seven target classes corresponding to the two different labelling regimes employed in our experiments (compare section 4.1.1), the final classification layer will comprise four/seven neurons respectively, i.e. one for each class. The corresponding loss function for the resulting multi-label classification problem is the *binary crossentropy* loss,

$$\mathcal{L}(y_i, \hat{y_i}) = -\frac{1}{n} \sum_{i=1}^{n} (y_i log(\hat{y_i}) + (1 - y_i) log(1 - \hat{y_i})), \tag{3.2}$$

where y_i and $\hat{y_i}$ are the ground truth label and the model output (resp) for the i-th class. With n being is the total number of classes, $n \in \{4, 7\}$ in our case.

3.2 Benchmark architectures

3.2.1 Simple-CNN-Benchmark

Since the main aim of this study is to establish whether or not the transformer-inspired architecture presented above, and more specifically the attention-based *encoder* module provides any marginal utility for the task at hand, the main common-sense benchmark that needs to be outperformed is simply the *CNN module* identical to the one used in the transformer, employed in isolation and followed by flattening of the feature maps and the same two-fold classification layer used in the proposed architecture. See figure 3.5 for a diagram illustrating this configuration.

3.2.2 DNN-LSTM-Benchmark proposed by Yildirim et al., 2020

The second benchmark that was used is based on the work of Yildirim et al. (2020), in which a DNN-LSTM network was proposed and trained on the Shaoxin People's Hospital ECG database. Since CNN-LSTM-type architectures, sometimes combined with an attention-mechanism, are very commonly and successfully used for ECG related tasks in recent publications (i.e. He et al., 2019;Zhang and Li, 2021) and therefore can be seen a state-of-the-art, the model was chosen as an additional benchmark against which to measure the proposed architecture.

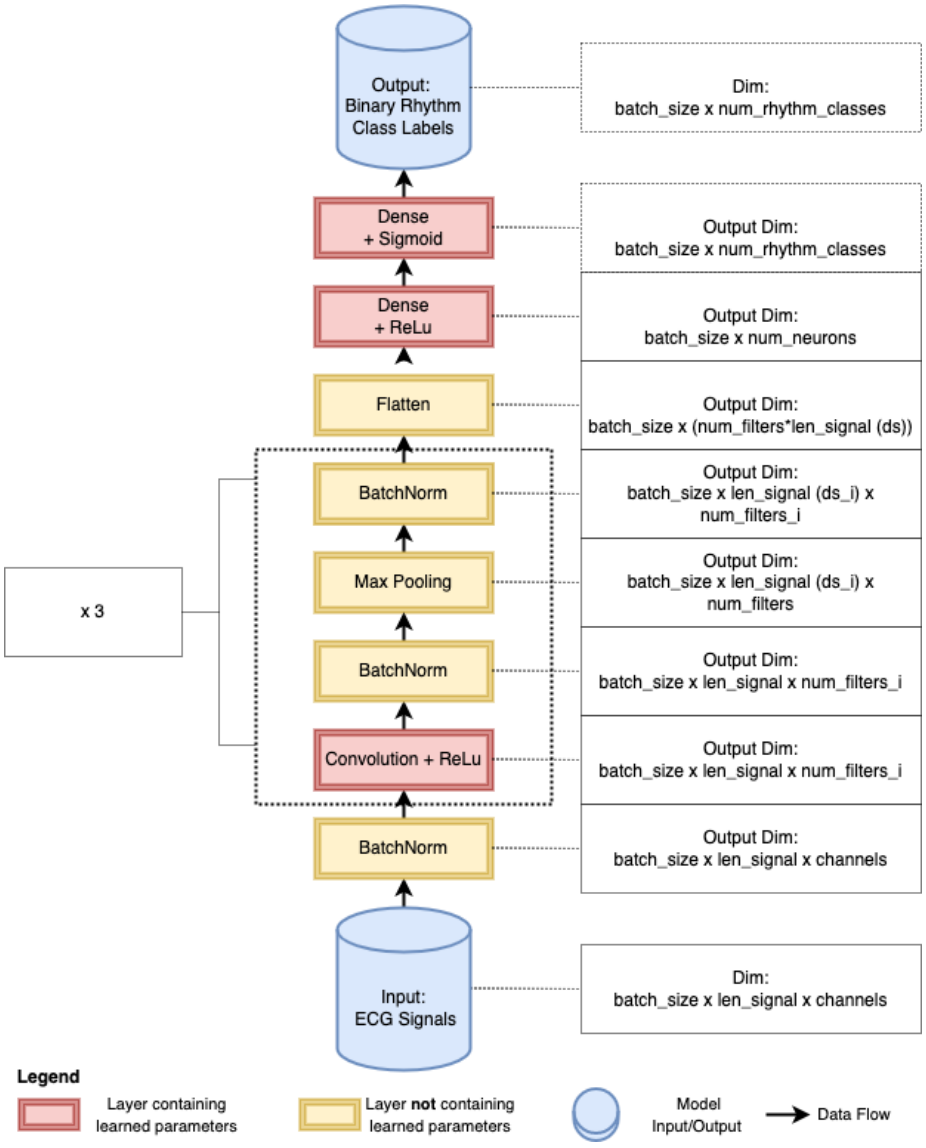


Figure 3.5: Diagram of the Simple-CNN-Benchmark.

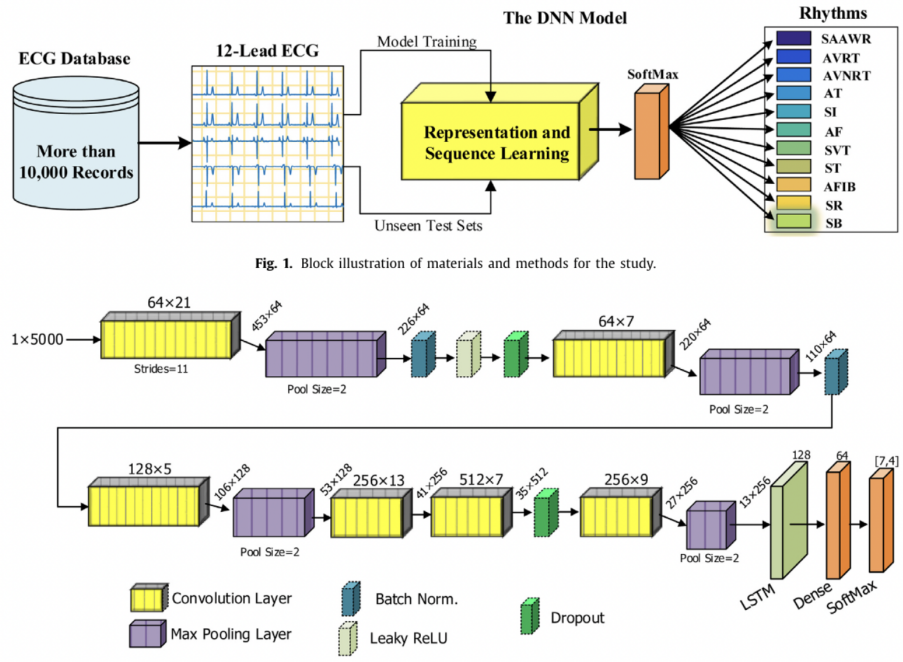


Figure 3.6: Diagram of the DNN-LSTM-Benchmark proposed by Yildirim et al., 2020. From Yildirim et al. (2020).

Chapter 4

The data

All the data used in this study is publicly available and hence the results obtained can be tested and verified in future investigations. The data was chosen based on the following criteria:

1. Relevant labelling.
2. Size - larger databases are better in order to increase performance and prevent overfitting (compare section 2.4.4).
3. Total number of patients - in order to avoid learning patient-specific features and improve generalisability.
4. Class balance - the diagnostic classes should be as balanced as possible within the limitations of clinical incidence.
5. Availability of suitable benchmarks in literature.

4.1 The Shaoxing People's Hospital ECG Database

The Shaoxing People's Hospital ECG Database was assembled by Zheng et al. and published in 2020. It comprises ECG data from more than 10646 unique patients and hence meets our **second criterion. Furthermore, the samples are of unique length and sampling frequency** at 10 seconds and 500 Hz respectively, fulfilling the **third criterion**. Extra effort has further

Acronym Name	Full Name	Frequency, n(%)	Age, Mean ± SD	Male, n(%)
SB	Sinus Bradycardia	3,889 (36.53)	58.34 ± 13.95	2,481 (58.48%)
SR	Sinus Rhythm	1,826 (17.15)	54.35 ± 16.33	1,024 (56.08%)
AFIB	Atrial Fibrillation	1,780 (16.72)	73.36 ± 11.14	1,041 (58.48%)
ST	Sinus Tachycardia	1,568 (14.73)	54.57 ± 21.06	799 (50.96%)
AF	Atrial Flutter	445 (4.18)	71.07 ± 13.5	257 (57.75%)
SI	Sinus Irregularity	399 (3.75)	34.75 ± 23.03	223 (55.89%)
SVT	Supraventricular Tachycardia	587 (5.51)	55.62 ± 18.53	308 (52.47%)
AT	Atrial Tachycardia	121 (1.14)	65.72 ± 19.3	64 (52.89%)
AVNRT	Atrioventricular Node Reentrant Tachycardia	16 (0.15)	57.88 ± 17.34	12 (75%)
AVRT	Atrioventricular Reentrant Tachycardia	8 (0.07)	57.5 ± 16.84	5 (62.5%)
SAAWR	Sinus Atrium to Atrial Wandering Rhythm	7 (0.07)	51.14 ± 31.83	6 (85.71%)
All	All	10,646 (100)	51.19 ± 18.03	5,956 (55.95%)

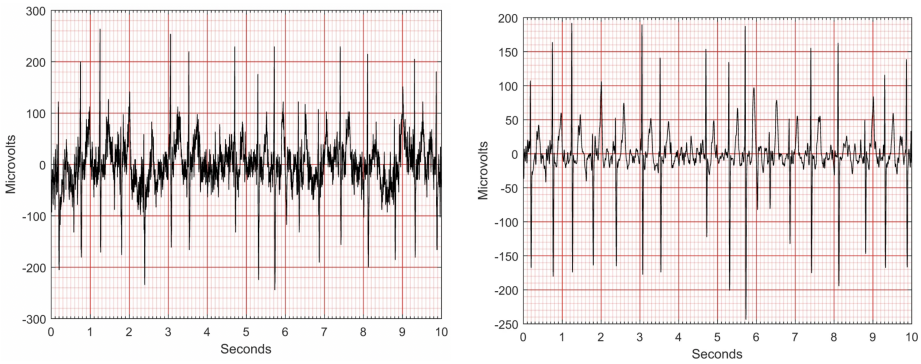
Table 4.1: Summary table of the rhythm classes present in the Shaoxing dataset with corresponding frequencies and baseline characteristics. Table from Zheng et al. (2020).

Merged from	Merged to	Total	Training data size (80%)	Testing data size (20%)
AFIB, AF	AFIB	3,889	3,111	778
SVT, AT, SAAWR, ST, AVNRT, AVRT	GSVT	2,307	1,846	461
SB	SB	2,225	1,780	455
SR, SI	SR	2,225	1,780	455
All	All	10,646	8,517	2,129

Table 4.2: Mapping of the Shaoxing rhythm classes to corresponding superclasses as proposed by Zheng et al. (2020). From Zheng et al., 2020.

Attributes	Type	Value Range	Description
FileName	String		ECG data file name (unique ID)
Rhythm	String		Rhythm Label
Beat	String		Other conditions Label
PatientAge	Numeric	0–999	Age
Gender	String	MALE/FEMAL	Gender
VentricularRate	Numeric	0–999	Ventricular rate in BPM
AtrialRate	Numeric	0–999	Atrial rate in BPM
QRSDuration	Numeric	0–999	QRS duration in msec
QTInterval	Numeric	0–999	QT interval in msec
QTCorrected	Numeric	0–999	Corrected QT interval in msec
RAxis	Numeric	–179~180	R axis
TAxis	Numeric	–179~181	T axis
QRSCount	Numeric	0–254	QRS count
QOnset	Numeric	16 Bit Unsigned	Q onset (In samples)
QOffset	Numeric	17 Bit Unsigned	Q offset (In samples)
TOffset	Numeric	18 Bit Unsigned	T offset (In samples)

Table 4.3: List of the additional global features associated with each record of the Shaoxing database. From Zheng et al. (2020).



(a) (b)

Figure 4.1: Example signal drawn from the Shaoxing database in (a) raw and (b) pre-processed form. From Zheng et al. (2020).

been invested into balancing the classes through a physician-verified regrouping, and hence the dataset meets our **fourth criterion** better than comparable datasets such as the MIT-BIH Cardiac Arrhythmia dataset (Zheng et al., 2020).

For each example, the database provides one *rhythm label* per example, comprising *sinus rhythm* and various types of arrhythmia (see table 4.1), thereby meeting the *first criterion*, as well as an optional *condition label* based on cardiac conditions or abnormal beats (see table 4.4). Furthermore, a total of 16 categories of patient-specific data are provided, including the baseline-characteristics of name and age as well as a number of global statistical properties of the respective ECG-signals (see fig 4.3).

The availability of a state-of-the-art benchmark through the work of Yildirim et al. (2020) further checks our **fifth criterion**.

The data is provided in a *raw* as well as a *pre-processed*, with the latter having undergone de-noising using a *butterworth-low-pass filter* at a cutoff of 50 Hz in order to remove high frequency noise unlikely to be of meaningful cardiac physiological origin as well as *LOESS-smoothing* to remove baseline wandering (Zheng et al., 2020). This will be further discussed in section 5.3.

Condition Acronym	Description
AVB	1st degree atrioventricular block
2AVB	2nd degree atrioventricular block
2AVB1	2nd degree atrioventricular block(Type one)
2AVB2	2nd degree atrioventricular block(Type two)
3AVB	3rd degree atrioventricular block
ABI	atrial bigeminy
ALS	Axis left shift
APB	atrial premature beats
AQW	abnormal Q wave
ARS	Axis right shift
AVB	atrioventricular block
CCR	countercolockwise rotation
CR	colockwise rotation
ERV	Early repolarization of the ventricles
FQRS	fQRS Wave
IDC	Interior differences conduction
IVB	Intraventricular block
JEB	junctional escape beat
JPS	J point shift
JPT	junctional premature beat
LBBB	left bundle branch block
LBBBB	left back bundle branch block
LFBBB	left front bundle branch block
LRRI	Long RR interval
LVH	left ventricle hypertrophy
LVHV	left ventricle high voltage
LVQRSAL	lower voltage QRS in all lead
LVQRSCL	lower voltage QRS in chest lead
LVQRSLL	lower voltage QRS in limb lead
MI	myocardial infarction
MIBW	myocardial infraction in back wall
MIFW	Myocardial infgraction in the front wall
MILW	Myocardial infraction in the lower wall
MISW	Myocardial infraction in the side wall
PRIE	PR interval extension
PWC	P wave Change
QTIE	QT interval extension
RAH	right atrial hypertrophy
RAHV	right atrial high voltage
RBBB	right bundle branch block
RVH	right ventricle hypertrophy
STDD	ST drop down
STE	ST extension
STTC	ST-T Change
STTU	ST tilt up
TWC	T wave Change
TWO	T wave opposite
UW	U wave
VB	ventricular bigeminy

Table 4.4: Comprehensive list of the optional condition- and beat labels included in the Shaoxing database.

Rhythms	Number of Total Samples	Number of Training Samples	Number of Testing Samples	Age, Mean ± Std	Number of Females	Number of Males
AF	438	406	32	71.14 ±13.47	182	256
AFIB	1,780	1,622	158	73.35 ±11.13	739	1,041
SI	397	355	42	34.88 ±23.00	175	222
SB	3,888	3,494	394	58.33 ±13.95	1,408	2,480
SR	1,825	1,632	193	54.37 ±16.29	1,024	801
ST	1,564	1,398	166	54.67 ±20.97	769	795
SVT	544	485	59	55.64 ±18.35	294	250
All	10,436	9,392	1,044	59.16 ±17.94	4,591	5,845

Table 4.5: Summary table of the *reduced* classes proposed by Yildirim et al., 2020 for the Shaoxing dataset. Table from Yildirim et al. (2020)

. .

Superclass	*Reduced* classes contained
AFIB	AFIB, AF
GSVT	SVT,ST
SB	SB
SR	SR, SI

Table 4.6: Mapping between the *superclasses* and *reduced* regime.

4.1.1 Labelling Regimes

Since the varying frequencies of incidence in the original labelling regime result in a large class imbalance (see figure 4.1), Zheng et al. (2020) have further suggested a four-fold labelling regime of *superclasses* comprising atrial fibrillation/atrial flutter (AFIB/AF), generalised supraventricular tachycardia (gSVT), sinus bradycardia (SB) and sinus rhythm/sinus irregularity (SR). Table 4.2 shows this mapping as well as the resulting frequencies of the superclasses and baseline characteristics.

In order to take full advantage of both the size and diversity of this database while remaining within the realm of practicability with regards to class imbalance, Yildirim et al. (2020) propose an alternative labelling regime by eliminating the rarest labels (namely AT, AVNRT, AVRT and SAAWR - compare table 4.1) altogether, which was appropriately named *reduced* regime. The resulting dataset provides a finer "diagnostic resolution" than the *superclasses* regime proposed by Zheng et al. (2020), at the cost of higher (yet still manageable) class imbalance. Summary statistics of the *reduced* regime are provided in table 4.5 while a mapping of *reduced* to *superclasses* is provided in table 4.6.

PTB-XL Diagnostic Class	Shaoxin *Superclass*
Atrial Flutter (AFL)	AFIB/AF
Atrial Fibrillation (AFIB)	AFIB/AF
Sinus Tachycardia (STACH)	gSVT
Supraventricular tachycardia (SVTAC)	gSVT
Paroxysmal supraventricular tachycardia (PSVT)	gSVT
Sinus Tachycardia (STACH)	gSVT
Sinus Bradycardia (SBRAD)	SB
Sinus Arrhythmia (SARRH)	SR
Sinus Rhythm (SR)	SR

Table 4.7: Mapping of PTB-XL rhythm annotations to Shaoxing *superclasses*.

4.2 The PTB-XL Database

The PTB-XL database (Wagner et al., 2020) is a comparably large database of clinical 12-lead ECG signals comprising 21837 examples from 18885 patients at a uniform length of 10 seconds and a sampling frequency of 500 Hz. According to the authors, the value of the dataset results from "the comprehensive collection of many different co-occurring pathologies, but also from a large proportion of healthy control samples.".

The database was chosen for validation, thanks to its similarities to the Shaoxing database in size, sample length and corresponding label strength. In order to make it compatible with classifiers trained on one of the above labelling regimes, we ignored all labels other than those with an (approximate) equivalent within one of the four superclasses defined by Zheng et al. (2020). The resulting class-mapping can be found in table 4.7.

Chapter 5

Methods

5.1 Structure and Overview of Experiments

Table 5.1 provides a comprehensive summary of the unique experimental configurations that form the empirical basis of this work. All models use the same training set drawn from the Shaoxin People's Hospital database as well as the same test set drawn from either the Shaoxing or PTB-XL database.

In order to account for the intrinsic randomness of the training process (i.e. through weight-initialisation and the use of SGD) as well as to establish a measure of uncertainty in the performance metrics of the resulting models, **ten** models were trained for each experimental configuration at 30 epochs each, using *early stopping* with a patience of 10 epochs and employing the $Adam$ algorithm (Kingma & Ba, 2014) with a LR of 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 1e-07$ for optimisation.

The raw results where then used to compute various performance metrics and characteristics as detailed in section 5.5 and the corresponding confidence intervals were established through the method detailed in section 5.7. While confusion matrices and ROC-curves for all experiments are provided in Appendix A, individual performance metrics are provided and compared in context in the tables and figures of sections 6.2-6.7.

Exp.	Architecture	Test Data	Balancing	Pre-Proc.	Labelling
1	Benchmark	Shaoxing	Over	Yes	Super
2	Benchmark	Shaoxing	None	Yes	Reduced
3	Benchmark	PTB-XL	Over	No	Super
4	Benchmark	Shaoxing (20s)	Over	Yes	Super
5	Benchmark	Shaoxing (50s)	Over	Yes	Super
6	Plain	Shaoxing	None	No	Super
7	Plain	Shaoxing	Under	No	Super
8	Plain	Shaoxing	Over	No	Super
9	Plain	Shaoxing	Over	Yes	Super
10	Plain	Shaoxing	None	Yes	Reduced
11	Plain	PTB-XL	Over	No	Super
12	Plain	Shaoxing (20s)	Over	Yes	Super
13	Plain	Shaoxing (50s)	Over	No	Super
14	Multi-In	Shaoxing	Over	Yes	Super
15	Multi-In	Shaoxing	None	Yes	Reduced
16	Multi-Out	Shaoxing	Over	Yes	Super
17	Multi-Out	Shaoxing	None	Yes	Reduced
18	Multi-In/Out	Shaoxing	Over	Yes	Super
19	Multi-In/Out	Shaoxing	None	Yes	Reduced

Table 5.1: Summary table of the experimental configuration for each experiment conducted.

5.2 Balancing

There are several reasons for why class-imbalance in the training set can be undesirable in machine learning, perhaps the most important and obvious being that an imbalanced data set will -on average- result in imbalanced mini-batches and thereby create an incentive for the optimizer to overemphasize performance on the majority class, as can be seen from equation 2.3.

While in *multi-label* classification problems, even given a perfectly balanced training set, each individual class will be in the minority compared to all the others combined, it has nevertheless been our hypothesis that a balanced data set will lead to a better and/or more consistent performance as well as better applicability of common metrics such as *accuracy* during the training process (compare section 5.5)[1].

The first balancing method consisted of *oversampling* of the training set by *randomly* duplicating examples until the number of examples for each class was equal to the number of unique examples of the majority class (*Sinus Bradycardia*). This resulted in a training set of 12444 examples. The second balancing method comprised *random undersampling* of the data by only using as many examples of each class as there are unique examples of the most underrepresented minority class. This naturally led to an overall loss of unique examples which hence became unavailable for training. The resulting training set comprised 6800 examples.In table 5.1, these methods are declared as "over" and "under" respectively.

For reasons of comparability and reproducibility, only the training data was manipulated while the test data was always maintained in its original, unbalanced form.

[1]To understand why this is the case, one has to remember that the test set has the same imbalances as the training set and hence if one was to skew performance towards the majority class by performing training on an unbalanced data-set, the result would be an overly optimist estimate.

5.3 Pre-Processing

Pre-processing techniques such as *de-noising*, *de-trending* and *feature engineering* are indispensable for many classification techniques including most traditional (or "shallow") Machine Learning techniques (compare section 2.4). Despite this, there is the argument that all forms of pre-processing and/or denoising rely on more or less arbitrary assumptions and (presumed) prior knowledge about the data. It follows that one should avoid such "ad-hoceries" in favour preserving information and "letting the model decide what is signal and what is noisy" (Jaynes, 2003, p.223-236). This argument is supported by the fact that there is growing evidence that *deep learning* models are much more robust to such disturbances since their hierarchically organised feature extraction layers can *learn* to separate the signal from the noise[2]

The main sources of poor signal quality in ECG signals are *baseline wander*, *muscle noise*, *power-line* interference and - in some cases- *sensor motion artifacts* (Kher, 2019). As stated in section 4.1, Zheng et al. (2020) employ LOESS-smoothing to remove baseline wander and a 50-Hz butterworth-low-pass filter to filter out muscle noise and power-line interference. Motion artifacts are more difficult to detect due to their idiosyncratic nature, however, these are more common in long-term ECG recordings such as those obtained in ICU (i.e. Bashar et al., 2019).

In this work, for reasons of reproducibility and comparability, pre-processing will by default mean use of the pre-processed data proivided by Zheng et al. (2020). This is indicated by a "yes" in the corresponding columns in table 5.1. Following the same reasoning, pre-preocessing was omitted for models to be subsequently tested on a test set drawn from the PTB-XL database.

Note that there are a total of 18 examples in the pre-processed Shaoxing database which were considered "faulty" due to missing or NaN-values. These were consequently removed from the data-set and replaced with randomly copies of other examples with the same class-label in order to ensure unchanged cardinality.

[2]Provided that overfitting is prevented through regularisation and appropriate model capacity (Goodfellow et al., 2016).

5.4 Use of Augmented Data Sets

In order to enable scrutiny of the models' ability to generalise to longer sequences and the potential presence of different types of cardiac arrhythmia in the same signal (resulting in *mixed labels*), we created two sets of *augmented* data based on the Shaoxing test set. The two sets have the same cardinality as the original test set and comprise sequences of 20s and 50s respectively. In table 5.1, these are denoted as "Shaoxing (20s)" and "Shaoxin (50s)" respectively.

Examples were created by random concatenation of elements of the original test set and linear combination of the associated label vectors. Naturally, due to the fact that the sampling was random, labels are mixed and include a maximum of two positive classes for the shorter sequences and four positive classes (i.e. the maximum) for the longer sequences.

The rationale behind this approach is that despite the resulting signals being discontinuous as hence "unnatural", the models should nevertheless be able to pick up characteristic features of the different diagnostic classes due to the position-invariant nature of the feature-extraction process.

5.5 Performance Metrics and Characteristics

Because all models in this study were designed to perform *multi-label* classification, i.e. multiple simultaneous *binary* classifications, we will limit our discussion to performance metrics for *binary classifiers*, i.e. classifiers designed to distinguish between *positives* and *negatives*.

In the following, the different types of performance metrics used in evaluating our experiments are briefly introduced.

5.5.1 Accuracy

The *accuracy* of a classifier on a task is the frequency of correct predictions compared to the total number of predictions made:

$$\text{accuracy} = \frac{\text{Number of correct predictions}}{\text{Number of predictions made}} \tag{5.1}$$

While being a good measure of overall performance, one can easily see that the utility of accuracy as a performance metric breaks down with increasing class-imbalance, as in the limit the classifier could achieve almost perfect accuracy by simply always predicting the dominant class (i.e. Géron, 2019).

Furthermore, to avoid confusion in the context of *multilabel classification*, it is important to note that we define *overall* accuracy as the number of correct label vectors divided by the total number of examples, as opposed to an average of the class-wise binary accuracies. Since partially correct predictions do not count under the former definition, it gives a more pessimistic estimate than the latter. Although it is not explicitly stated in the publication, based on our scrutiny of the results reported, we assume that Yildirim et al., 2020 have done the same.

5.5.2 Precision

An accuracy-like metric that only focuses on positive predictions is *precision* or *positive predictive value*, defined in equation 5.2.

$$\text{precision} = \frac{TP}{TP + FP} \tag{5.2}$$

One can see that since this metric is normalised by the total number of positive examples, the base rate of the positive class in the test-set/population does not distort the result. *Precision* can be thought of as the likelihood that a positive classification is in fact correct.

5.5.3 Sensitivity

A metric often used in conjunction with *precision* is *sensitivity*, sometimes called *recall* or *true positive rate* (TPR). It is defined in equation 5.3.

$$\text{sensitivity} = \frac{TP}{TP + FN} \tag{5.3}$$

Instead of normalising the true positive predictions by the total total number of positive predictions, it normalises by the total number of positives and therefore gives a measure of the likelihood that a positive example is identified as such.

5.5.4 Specificity

Another metric often used in conjunction with sensitivity is *specificity* or *true negative rate* (TNR), defined in equation 5.4.

$$\text{specificity} = \frac{TN}{TN + FP} \tag{5.4}$$

One can see that this metric complements *sensitivity* in the sense that it provides a measure of the probability that a member of the negative class is actually classified as such.

5.5.5 F_1-Score

The F_1-Score is a combined metric based on both *precision* and *recall* as specified in equation 5.5.

$$F_1 = \frac{2}{\frac{1}{precision} + \frac{1}{recall}} = \frac{TP}{TP + \frac{FN+FP}{2}} \tag{5.5}$$

As one can see from the denominator, F_1 gets maximised when *precision* and *recall* are both high. However, since high *precision* usually comes at the cost of low *recall* (see below), it can be said that the F_1-Score favours classifiers that have similar values for both (Géron, 2019).

5.5.6 Confusion Matrices

One way of visually summarising the performance of a classifier based on some of the metrics defined above are *confusion matrices*. In the case of a binary classifier this is a 2x2 matrix either comprising the absolute numbers TP, TF, FP and FN, or their respective rates TPR, TFR, FPR, FNR. Definitions of FPR and FNR are given in equation 5.6 and 5.7 respectively.

$$FPR = 1 - TNR = \frac{FP}{TN + FP} \tag{5.6}$$

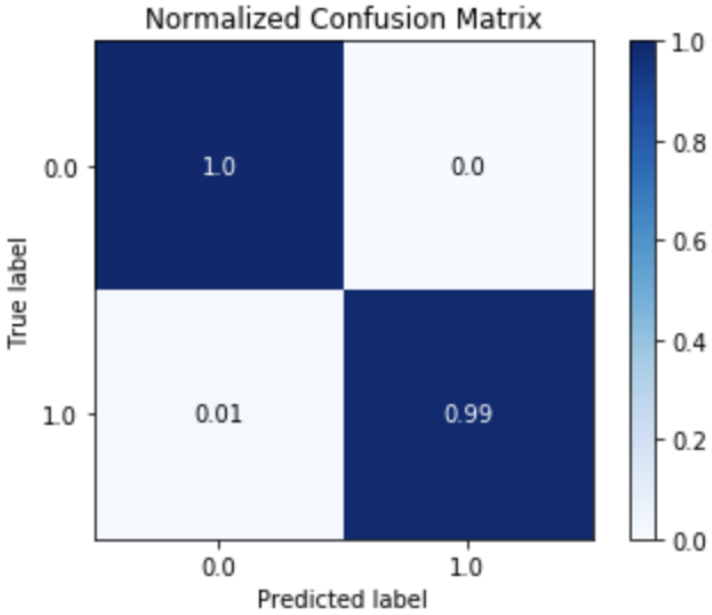


Figure 5.1: Schematic of a confusion matrix. By Kevin Jolly, last accessed 29th December 2021 from www.oreilly.com.

$$FNR = 1 - TPR = \frac{FN}{TP + FN} \tag{5.7}$$

A schematic of a confusion matrix based on *rates* (as opposed to absolute values) is given in figure 5.1.

Note that in order to present the outcomes of our multiple evaluation runs in the best possible way, we had to summarise over the individual results in order to create *average confusion matrices*. These naturally violate the row-wise unitarity property of normal confusion matrices, however this trade-off was accepted in favour of conciseness and expressiveness.

Furthermore, although we have chosen to frame the problem in a *multi-label* way, we have additionally included average *multi-class* confusion matrices for those experiments were ground truth labels were mutually exclusive.

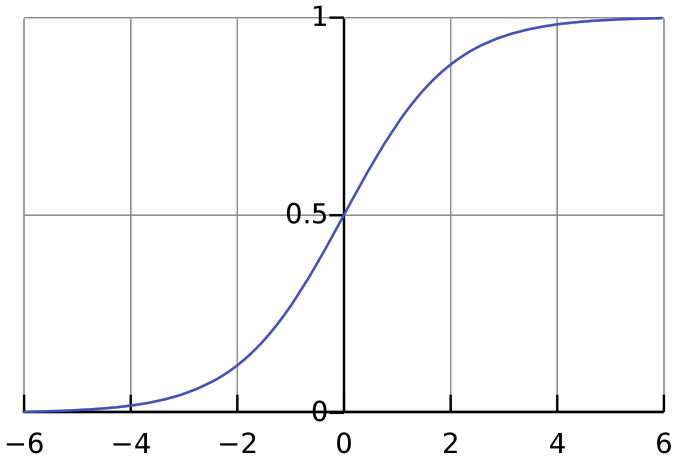


Figure 5.2: Graph of the sigmoid function. The horizontal grid line drawn at y=0.5 corresponds the standard decision boundary.

5.5.7 ROC, Sensitivity/Specificity Curves and AUC

Many machine learning classifiers such as ANNs obtain their classification results by means of a *decision function* with a continuous yet bounded outcome ($\in [0, 1]$) that is often interpreted as the probability measure associated with a *logit model*[3]. As desicussed in chapter 3, the commonly used decision function in binary classification is the *sigmoid* function (see equation 3.1).

When trying to decide which threshold to use on the output value in order to separate positive from negative predictions, one encounters the *precision/recall* and *sensitivity/specificity trade-off* - the fact that by manipulating the threshold upwards or downwards in order to improve one of the performance metrics, one tends to decrease the other[4]. The central horizontal grid line in figure 5.2 provides an illustration of such a threshold, corresponding to the value 0.5. This trade-off is comprehensively captured in the *Receiver Operating Characteristic* or ROC-curve of a classifier, which plots TPR against FPR for a number of different threshold values.

[3]In fact, the classification layer of a neural network can be viewed as performing ordinary logistic regression using the *learned* features of previous layers as input.

[4]It is easy to see why this has to be the case for by looking at the extreme cases of thresholds of either 1.0 (perfect precision) or 0.0 (perfect recall).

A very concise and comprehensive metric derived from ROC-curves is the *area under the curve (AUC)*, capturing the performance of a classifier over the whole spectrum of possible thresholds values by approximately "integrating" *TPR* over the corresponding *FPR* values.

All metrics displayed in the results tables where calculated for a default threshold of 0.5 and the corresponding AUC curves and confusion matrices of each set of classifiers can be found in Appendix A.

5.6 Model Selection Procedure

The task of *model-selection* in ANNs and other models with non-convex loss functions and large numbers of hyperparameters is non-trivial and - by its very nature - not perfectly rigorous, as one can rarely be certain of the non-existence of another set of hyperparameters outside of the search space which leads to better performance results given the same training and testing data. In fact, this *tuning* of HPs can be seen as an optimization problem in its own right, albeit without a differentiable gradient due to the discrete nature of most HPs.

There are a number of methods that each add a certain amount of rigour to the process of finding the "optimal" set of hyper parameters. These include brute-force methods such as *grid search* and *random search* (Bergstra & Bengio, 2012) as well as bandit-based (Li et al., 2017) and bayesian methods (i.e. Joy et al., 2016).

While the use of *grid search* might be the most obvious choice, it is important to note that it can quickly lead to a waste of computational resources due to some HPs having negligible impact on the result. Furthermore, the act of defining the grid itself can limit the search space and thereby arbitrarily exclude feasible solutions.

A less wasteful improvement over the *grid search* method has been proposed by Bergstra and Bengio (2012) in the form of *random search*. The main benefit of this method is that it decreases the likelihood of a disproportionate amount of computational time being spend on unproductive "dead ends" of the HP space. Yet, one still has to design the search-space carefully in order to al-

low the optimiser to find an adequate solution within a feasible number of trials.

Bandit-based and bayesian methods such as *hyperband search* and *bayesian optimization* both attempt to dynamically adapt the search space based on prior results, thereby incrementally "zeroing in" on an optimal set of hyper parameters. The cardinal problem of the former is the trade-off between *exploration* and *exploitation*, while the latter tends can be analytically intractable (Bishop & Nasrabadi, 2006) or - at the very least- computationally expensive, and they both seem to be most appropriate for very large highly irregular HP spaces, where only a small subset of the space leads to feasible solutions.

Like most deep learning architectures, our proposed architecture and its corresponding benchmark have a significant yet not excessive number of hyperparameters which - due to their substantial impact on model capacity and performance - need to be optimised prior to conducting further experiments. Based on the discussion above, the following three-step procedure for selecting optimal hyperparameters was consequently devised:

1. Initially, a number of unsystematic trials were conducted using a fixed train-validation-split (of the training set) in order to establish reasonable bounds to the search space.

2. Within the henceforth established bounds of the grid (see table 6.1.), 500 trials using the *random search* method were subsequently conducted - again at fixed training-validation split - and a maximum of 30 epochs per trial, while using an early stopping mechanism with a patience of 10 epochs.

3. After these trials were complete, the 6 most promising sets of HPs (see Appendix B) were chosen and subjected to a more rigorous 10-fold cross-validation procedure, which due to its use of multiple training and validation bootstraps and corresponding models allows not only for a more effective use of the training data but also for an establishment of confidence intervals (see section 5.7).

5.7 Confidence Intervals

As discussed in section 2.5.4, non-bayesian machine learning classifiers not based on ensembles are inherently ignorant of the model risk associated with their individual predictions. The same problem arises when trying to determine the confidence intervals associated with the performance metrics obtained for a single classifier.

One possible solution would be the use of bootstrapped sub-samples of both the training and the test set in order to gain an estimate of the associated variance in performance. However, this approach would also introduce a pessimistic bias through a reduction in size of the training set (Vanwinckelen & Blockeel, 2012) as well as require an artificially large test set, thereby further reducing the available training data.

The approximate, data-efficient solution chosen in this study is the following:

1. Use the inherent randomness of the training process to produce an estimate of the variance in model performance.
2. Employ *binomial proportion confidence intervals* (see Wallis, 2013) in order to gain an estimate of the uncertainty associated with our particular choice of test set.

With regards to 1., one needs to note that even when trained on identical data, there is no guarantee that two classifiers based on the same model architecture will find identical minima of the objective function and weight-configurations due to the non-convexity of the problem (Goodfellow et al., 2016) and the inherent randomness of the training process (i.e. due to random weight initialisation and random mini-batch sampling, both of which were used in the experiments here presented). This randomness, which us assumed to dominate the variability on model parameters, can be exploited in order to gain an uncertainty-estimate for the model performance on our particular test set.

2. is aimed at the uncertainty associated with the finiteness of the test set, and hinges on the observation that the performance metrics introduced in section 5.5 can all be interpreted "success/failure ratios" (or, in the case of

F_1,derived from such). We can assume their variance to approximately take the form of a Binomial distribution (Wallis, 2013) with variance given by equation 5.8,

$$V(\hat{p}) = \frac{\hat{p}(1-\hat{p})}{n}, \tag{5.8}$$

where $\hat{p}$ is the estimated metric and n is the number of examples in the test set.

Since both of these sources of uncertainty are clearly independent from each other, they can assumed to add in quadrature. Combined with the general expression for the standard error on the mean (see Barlow, 1993), this leads to the following equation as the expression for the uncertainty associated with the mean of each performance metric based on m training runs of the model, each tested on the same training set of n examples and yielding a result p_i:

$$SE(\bar{\hat{p}}) = \sqrt{\frac{1}{\sum\limits_{m} \frac{1}{Var(\hat{p}) + \frac{\hat{p}(1-\hat{p_i})}{n}}}} \tag{5.9}$$

,

Here, $Var(\hat{p})$ is just the sample variance of $\{p_i | i \in 1, ..., m\}$.

Chapter 6

Results and Discussion

In this chapter, we will first present the results of the model selection procedure presented in section 5.6, followed by a detailed presentation and discussion of the results obtained from the experiments defined in 5.1 in light of the research objectives (compare section 1.1). For added clarity, at the beginning of each chapter, elements of a set of *defining characteristics* are mapped to corresponding experiments from table 5.1 and, within that section, will serve as a working title for that particular experiment.[1]

With regards to the objectives as specified in section 1.1, section 6.2 and 6.3 are aimed at **objective 1**, section 6.4 at **objective 2**, section 6.5 at **objective 3**, section 6.6 at **objective 4** and section 6.7 at **objective 5** and section 6.8 at **objective 6**.

6.1 Establishing Hyperparameters

One significant finding that emerged from step one of our model selection procedure (see section 5.6) was that a down-sampling factor associated with the cumulative effect of successive pooling layers inside the CNN-embedding module of ~ 500, leading to an effective sampling-frequency of ~ 1 Hz for the *encoder* input (compare figures 3.1 and 3.3) was optimal - coinciding in order of magni-

[1] This mapping approach was deemed necessary due to the significant overlap between the experimental results relevant to each section.

tude with the frequency of the human cardiac cycle [2]. Deviations from this in both directions led to deteriorating performance and - in the case of decreased down-sampling - exploding training times.

Furthermore, since no statistically significant difference in performance could be identified based in the trials associated with **step 3**, the total number of *learnable parameters* as a result of a given choice of *hyperparameters* was chosen as an additional criterion for model selection, with less parameters at equal performance being considered more desirable[3].The HPs which were chosen as a result of this process are detailed in table 6.1.

Hyper Parameter	Grid Values	Best Value
N.o. filters in CNN-layer 1	$\{2^n \vert n \in \{4,5,6,7,8,9\}\}$	256
Filter-size in CNN-layer 1	{4,6,8,10,12,14,16,24,32}	32
Number of filters in CNN-layer 2	$\{2^n \vert n \in \{4,5,6,7,8,9\}\}$	128
Filter-size in CNN-layer 2	{4,6,8,10,12,14,16,24,32}	4
Number of filters in CNN-layer 3	$\{2^n \vert n \in \{4,5,6,7,8,9\}\}$	64
Filter-size in CNN-layer 3	{4,6,8,10,12,14,16,24,32}	16
Size of CNN-pooling-layer	{4,6,8,10}	8
N.o. *encoder*-layers	{1,2,3,4}	1
N.o. attention-heads in enc-layer	{1,2,4,6,8,10,16,32,64}	10
N.o. neurons in t.d. FC-layer	{1024,2058,5098}	2048
N.o. neurons in pre-final FC-layer	$\{2^n \vert n \in \{5,6,7,8\}\}$	32

Table 6.1: Best hyper parameter choices based on the model selection procedure.

Once the best optimal set of hyper parameters was established, the *dropout rate* was chosen through a series of unsystematic trials and fixed at 0.1.

[2]One possible explanation for why this might be the case lies in the analogy between words and beats (or *cycles*) that partly inspired the proposed architecture. Ideally, one might assume, all the relevant information concerning each beat is captured in the CNN-feature maps at each down-sampled time-step that then gets processed by the *encoder* in order to identify global patterns and/or predictive relationships. Of course this relationship can only be approximate due to natural intra- and inter-individual fluctuations in heart rate, particularly under pathological conditions. If true, the hypothesis would be an illustrative example of the effectiveness of *domain knowledge* in choosing effective priors in HP-space.

[3]This can be viewed as a naive implementation of *Occam's Razor*.

Defining Characteristic	Experiment Number
Unbalanced	Experiment 6
Undersampled	Experiment 7
Oversampled	Experiment 8

Table 6.2: Mapping of defining characteristics and experiment numbers for section 6.2

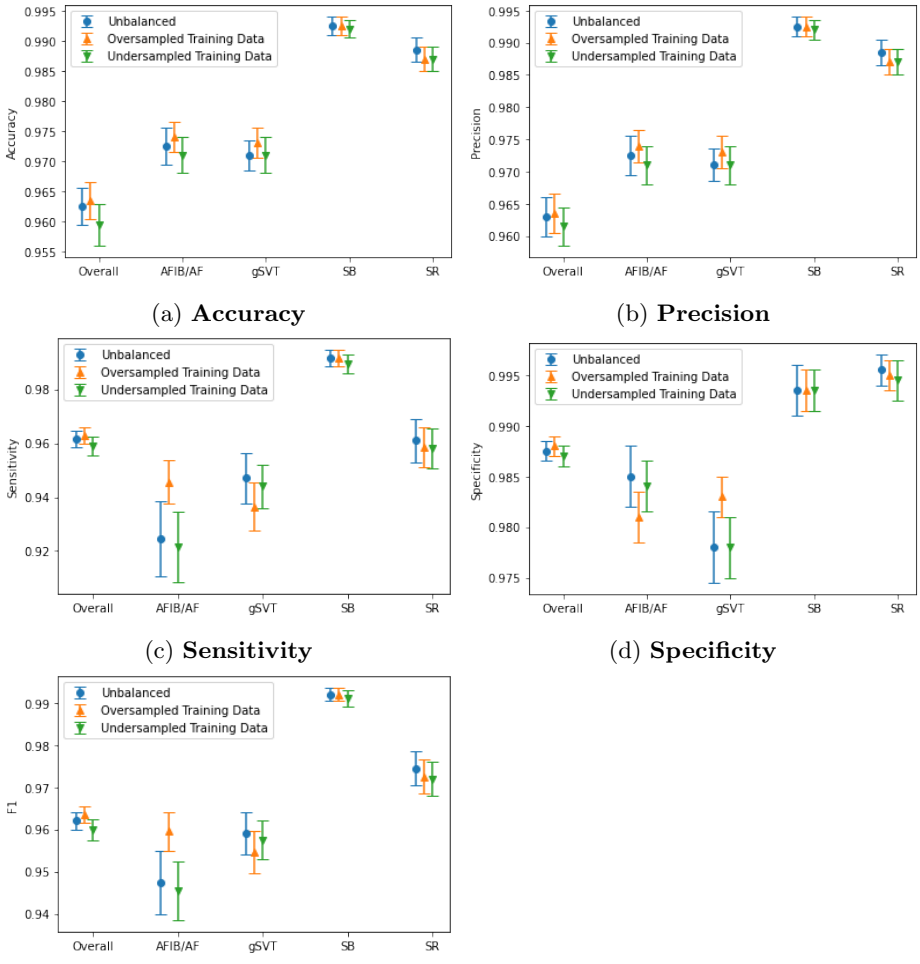
6.2 Comparing Models Based on Unbalanced Versus Balanced Training Data

In order to understand the effect of balancing the training data (see **objective 1**), models based on the proposed architecture were trained on our raw unbalanced training set drawn from the *Shaoxing* database as well as an oversampled and undersampled version of the same set. All experiments were conducted following the procedure outline in section 5.2 and the corresponding experiment numbers are 4,7 and 8 (see table 6.2).

The performance metrics are summarised in table 6.15 and class-wise as well as overall comparisons are made in figure 6.1 . ROC-curves and confusion matrices for all training runs are displayed in sections A.6, A.7 and A.8 of the Appendix A.

6.2.1 Discussion

One can see from table 6.15 as well as figure 6.1 that a statistically significant difference in the performance of either of the proposed methods of preparing the training set cannot be inferred. One might however conclude from the consistent trend in performance in the *overall* category that the oversampling methods leads to slightly better results than the unbalanced training set while the undersamping method has the opposite effect. The latter is hardly surprising since we are effectively withholding information from the model during training. What is interesting, furthermore, is that balancing the training data does not appear to result better relative performance on minority classes. In fact, with regards to the *generalised supraventricular tachycardia* class, is seems to have quite the opposite effect.



(a) **Accuracy** (b) **Precision**

(c) **Sensitivity** (d) **Specificity**

(e) $\mathbf{F}_1$

Figure 6.1: Mean values and 2-σ (95.4%) confidence intervals of performance metrics obtained for **models based on unbalanced, oversampled and undersampled training data**.

Diagnostic Class	Metric	Model: Unbalanced	Model: Raw/ Oversampled	Model: Raw/ Undersampled
Overall	Accuracy	0.9625 ±0.0030	0.9635 ±0.0030	0.9595 ±0.0035
	Precision	0.9630 ± 0.0030	0.9635 ± 0.0030	0.9635 ± 0.0030
	Recall	0.9615 ± 0.0030	0.9630 ± 0.0030	0.9630 ± 0.0030
	Specificity	0.9875 ± 0.0010	0.9880 ± 0.0010	0.9880 ± 0.0010
	F_1	0.9620 ± 0.0020	0.9635 ± 0.0020	0.9635 ± 0.0020
Atrial Fibrillation/ Atrial Flutter	Accuracy	0.9725 ± 0.0030	0.9740 ± 0.0025	0.9740 ± 0.0025
	Precision	0.9420 ± 0.0105	0.9300 ± 0.0085	0.9300 ±0.0085
	Recall	0.9245 ± 0.0140	0.9455 ± 0.0080	0.9455 ±0.0080
	Specificity	0.9850 ± 0.0030	0.9810 ± 0.0025	0.9810 ±0.0025
	F_1	0.9330 ± 0.0085	0.9380 ± 0.0060	0.9380 ±0.0060
Generalised Supraventricular Tachycardia	Accuracy	0.9710 ± 0.0025	0.9730 ± 0.0025	0.9730 ±0.0025
	Precision	0.9220 ± 0.0115	0.9390 ± 0.0075	0.9390 ±0.0075
	Recall	0.9470 ± 0.0095	0.9365 ± 0.0090	0.9365 ±0.0090
	Specificity	0.9780 ± 0.0035	0.9830 ± 0.0020	0.9830 ±0.0020
	F_1	0.9345 ± 0.0075	0.9375 ± 0.0060	0.9375 ±0.0060
Sinus Bradycardia	Accuracy	0.9925 ± 0.0015	0.9815 ± 0.0025	0.9925 ±0.0015
	Precision	0.9885 ± 0.0040	0.9815 ± 0.0025	0.9885 ±0.0035
	Recall	0.9985 ± 0.0015	0.9915 ± 0.0030	0.9915 ±0.0030
	Specificity	0.9935 ± 0.0025	0.9935 ± 0.0020	0.9935 ±0.0020
	F_1	0.9900 ± 0.0025	0.9900 ± 0.0020	0.9900 ±0.0020
Sinus Rhythm/ Sinus Irregularity	Accuracy	0.9885 ± 0.0020	0.9870 ± 0.0020	0.9870 ±0.0020
	Precision	0.9830 ± 0.0050	0.9800 ± 0.0055	0.9800 ±0.0055
	Recall	0.9610 ± 0.0080	0.9585 ± 0.0075	0.9585 ±0.0075
	Specificity	0.9955 ± 0.0015	0.9950 ± 0.0015	0.9950 ±0.0015
	F_1	0.9715 ± 0.0045	0.9690 ± 0.0050	0.9690 ±0.0050

Table 6.3: Mean metrics and 2-σ (95,4%) confidence intervals obtained for **models based on unbalanced, oversampled and undersampled training data**.

Despite the inconclusive experimental evidence and based on the theoretical argument in favour of oversampling, it was decided that all of the following experiments involving models based on the four *superclasses* would be conducted using **oversampled** training data.

6.3 Comparing models based on pre-processed versus raw input

In order to test the utility of pre-processing empirically for the proposed architecture (see **objective 1**), models based on training data consisting of *raw* and *pre-processed* signals were compared. (see sections 4.1 and 5.3 for details on the pre-processing methodology).

Based on the results of the previous section, models compared in this section were based on our oversampled training set drawn from the *Shaoxin* database and evaluated on the corresponding training set using the *superclasses* regime. As shown in table 6.4, the experiment numbers corresponding to the models based on raw and pre-processed data are 8 and 9 respectively.

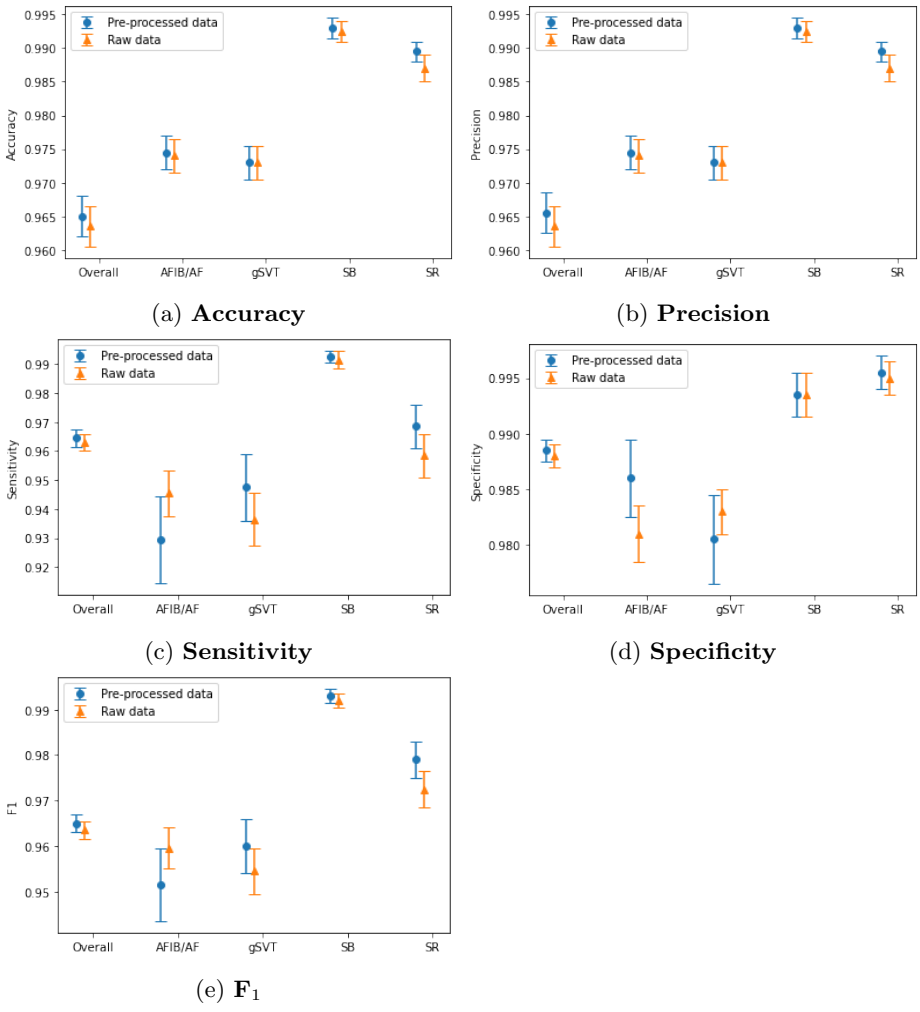
The resulting performance metrics are summarised and compared in table 6.5 as well as figure 6.2, while confusion matrices and corresponding ROC-curves are displayed in sections A.8 and A.9 of Appendix A.

6.3.1 Discussion

Based on the results, we cannot infer a statistically significant difference in performance between models based on raw and pre-processed data for the given architecture. While it appears that the models based on pre-processed data

Defining Characteristic	Experiment Number
Raw data	Experiment 8
Pre-processed data	Experiment 9

Table 6.4: Mapping of defining characteristics and experiment numbers for section 6.3.



(a) **Accuracy** (b) **Precision**

(c) **Sensitivity** (d) **Specificity**

(e) $\mathbf{F}_1$

Figure 6.2: Mean values and 2-σ (95.4%) confidence intervals of performance metrics obtained for **models based on raw and pre-processed data**.

Diagnostic Class	Metric	Model: Raw Data	Model: Pre-processed Data
	Accuracy	0.9635 ±0.0030	0.0.9650 ±0.0030
Overall	Precision	0.9635 ±0.0030	0.9655 ±0.0030
	Recall	0.9630 ±0.0030	0.9645 ±0.0030
	Specificity	0.9880 ±0.0010	0.9885 ±0.0010
	F_1	0.9635 ±0.0020	0.9650 ±0.0020
Atrial Fibrillation/ Atrial Flutter	Accuracy	0.9740 ± 0.0025	0.9745 ±0.0025
	Precision	0.9300 ±0.0085	0.9470 ±0.0120
	Recall	0.9455 ±0.0080	0.9295 ±0.0150
	Specificity	0.9810 ±0.0025	0.9860 ±0.0035
	F_1	0.9380 ±0.0060	0.9380 ±0.0095
Generalised Supraventricular Tachycardia	Accuracy	0.9730 ±0.0025	0.9730 ±0.0025
	Precision	0.9390 ±0.0075	0.9310 ±0.0135
	Recall	0.9365 ±0.0090	0.9475 ±0.0115
	Specificity	0.9830 ±0.0020	0.9805 ±0.0040
	F_1	0.9375 ±0.0060	0.9390 ±0.0090
Sinus Bradycardia	Accuracy	0.9925 ±0.0015	0.9930 ±0.0015
	Precision	0.9885 ±0.0035	0.9885 ±0.0035
	Recall	0.9915 ±0.0030	0.9925 ±0.0020
	Specificity	0.9935 ±0.0020	0.9935 ±0.0020
	F_1	0.9900 ±0.0020	0.9905 ±0.0020
Sinus Rhythm/ Sinus Irregularity	Accuracy	0.9870 ±0.0020	0.9895 ±0.0015
	Precision	0.9800 ±0.0055	0.9820 ±0.0060
	Recall	0.9585 ±0.0075	0.9685 ±0.0075
	Specificity	0.9950 ±0.0015	0.9955 ±0.0015
	F_1	0.9690 ±0.0050	0.9750 ±0.0050

Table 6.5: Mean metrics and 2-σ (95.4%) confidence intervals obtained for **models based on raw and pre-processed data**.

perform slightly better overall, the picture is less clear when looking at individual diagnostic classes. In particular, the large variance in precision, sensitivity and specificity for the AFIB and gSVT classes appears to be caused by the high number of misclassifications between the two classes, as illustrated by the confusion matrices for both models (see sections A.8 and A.9). One might assume this to be the case due to the morphological and rhythmic similarities between the two classes, yet the concern that low-pass filtering might cause the signal to lose vital physiological information in the high-frequency domain and thereby result in systematic underperformance seems to be unfounded based on the overall similar performance characteristics.

Consequently, and also because the only available third-party benchmark (provided by Yildirim et al., 2020) also uses the pre-processed data, the experiments for the subsequent sections (with an exception of section 6.7) where based on pre-processed data.

Defining Characteristic	Experiment Number
Plain,Superclasses Regime	Experiment 9
Plain, Reduced Regime	Experiment 10
Multi-input, Superclasses Regime	Experiment 14
Multi-input, Reduced Regime	Experiment 15
Multi-output, Superclasses Regime	Experiment 16
Multi-output, Reduced Regime	Experiment 17
Multi-in/out, Superclasses Regime	Experiment 18
Multi-in/out, Reduced Regime	Experiment 19

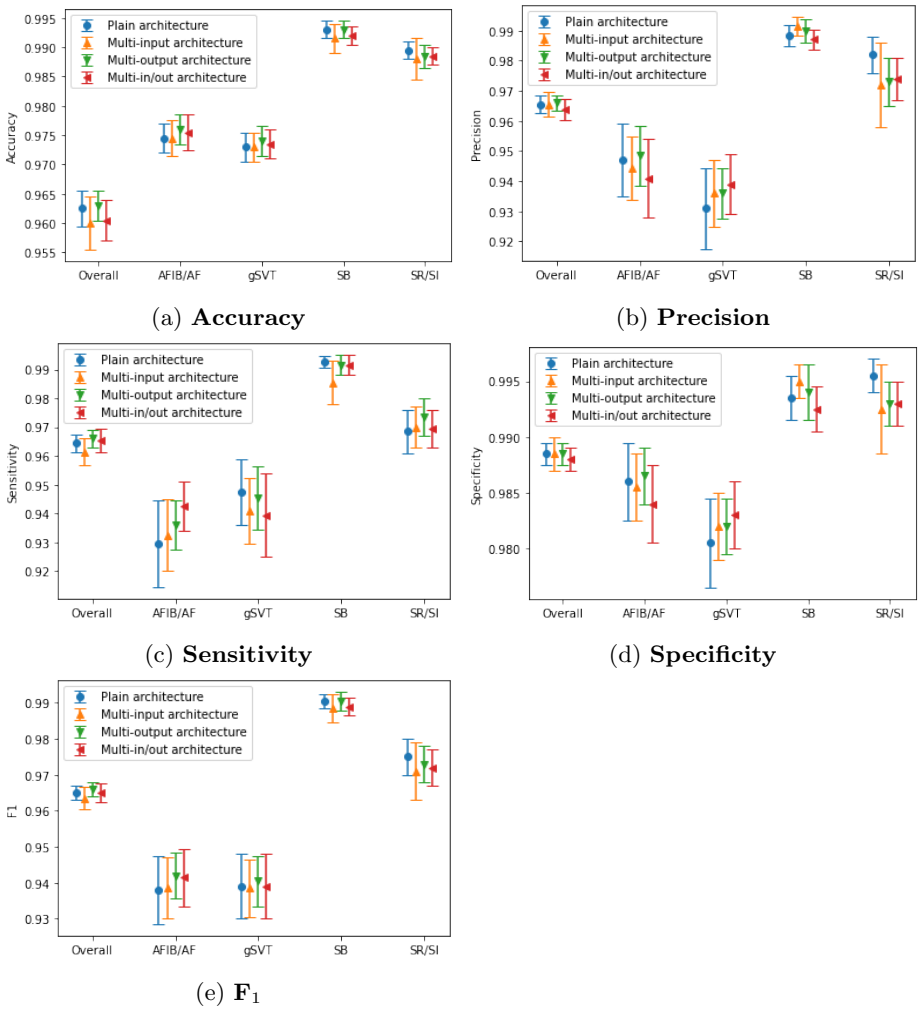
Table 6.6: Mapping of defining characteristics and experiment numbers for section 6.4

6.4 Evaluating Possible Modifications of the Proposed Architecture

In this section, the performance of the proposed architecture was compared to its modifications as discussed in section 3.1 (see **objective 2**. All models were trained using our de-noised and de-trended test set from the *Shaoxin* database (compare sections 4.1 and 5.3).

In order to further investigate whether one of the proposed modifications would be particularly useful with regards to certain types of arrhythmia, we made the same comparison for both the *superclasses* and *reduced classes* labeling regime. A mapping of the various combinations of architecture-modifications and labelling regimes the the corresponding experiments are detailed in table 6.6.

The performance metrics for the proposed model and potential modifications (i.e. multi-input, multi-output and multi-input/multi-output) are summarised in table 6.7 and 6.8, and class-wise as well as overall comparisons are drawn in figures 6.3 and 6.4. As before, the corresponding ROC-curves and mean confusion matrices for all ten training runs are displayed in sections A.9-A.10 as well as sections A.14 - A.19.



(a) **Accuracy** (b) **Precision**

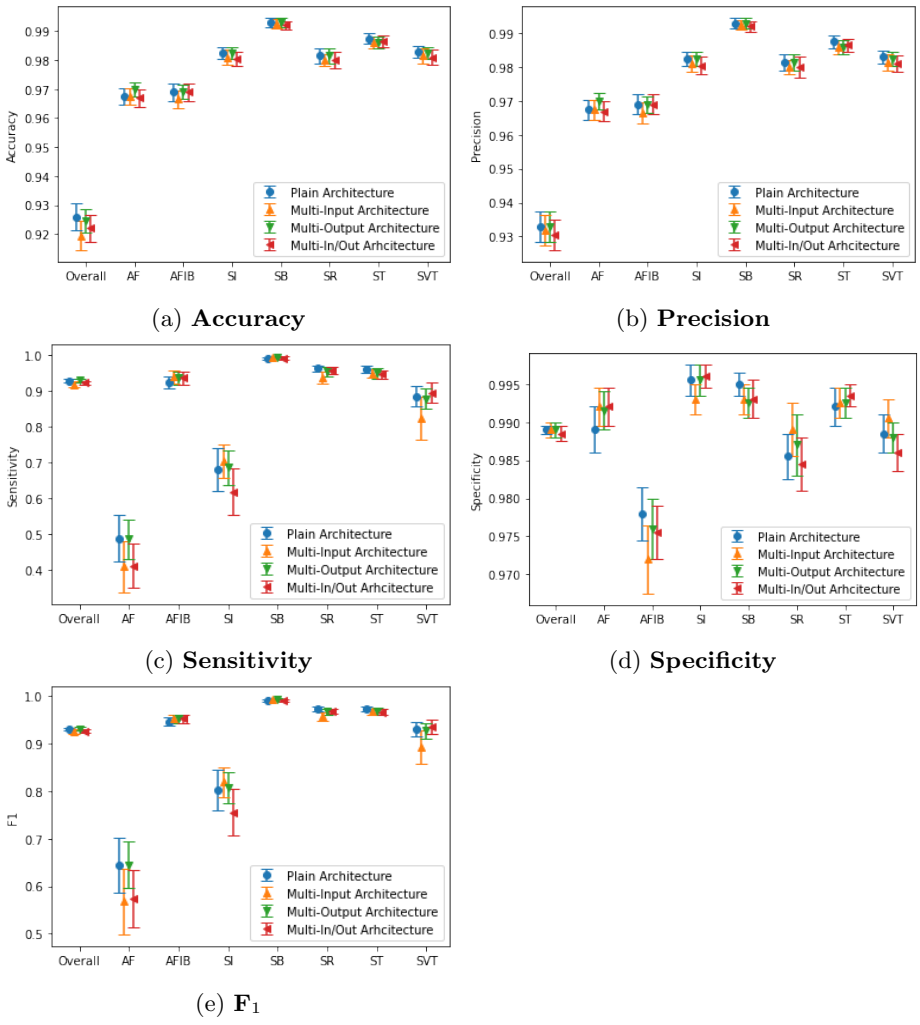
(c) **Sensitivity** (d) **Specificity**

(e) $\mathbf{F}_1$

Figure 6.3: Mean values and 2-σ (95.4%) confidence intervals of performance metrics obtained for **models based on different modifications of the proposed architecture within the *superclasses* labelling regime**.

Diagnostic Class	Metric	Model: Plain	Model: Multi- input	Model: Multi-output	Model: Multi-in-out
Overall	Accuracy	0.9650 ±0.0030	0.9600 ±0.0045	0.9630 ±0.0025	0.9605 ±0.0035
	Precision	0.9655 ±0.0030	0.9655 ±0.0040	0.9660 ±0.0025	0.9640 ±0.0035
	Recall	0.9645 ±0.0030	0.9615 ±0.0045	0.9660 ±0.0030	0.9655 ±0.0040
	Specificity	0.9885 ±0.0010	0.9885 ±0.0015	0.9885 ±0.0010	0.9880 ±0.0010
	F_1	0.9650 ±0.0020	0.9635 ±0.0030	0.9660 ±0.0020	0.9650 ±0.0025
Atrial Fibrillation/ Atrial Flutter	Accuracy	0.9745 ±0.0025	0.9745 ±0.0030	0.9760 ±0.0025	0.9755 ±0.0030
	Precision	0.9470 ±0.0120	0.9445 ±0.0105	0.9485 ±0.0100	0.9410 ±0.0130
	Recall	0.9295 ±0.0150	0.9325 ±0.0125	0.9360 ±0.0085	0.9425 ±0.0085
	Specificity	0.9860 ±0.0035	0.9855 ±0.0030	0.9865 ±0.0025	0.9840 ±0.0035
	F_1	0.9380 ±0.0095	0.9385 ±0.0085	0.9420 ±0.0065	0.9415 ±0.0080
Generalised Supraventricular Tachycardia	Accuracy	0.9730 ±0.0025	0.9730 ±0.0025	0.9740 ±0.0025	0.9735 ±0.0025
	Precision	0.9310 ±0.0135	0.9360 ±0.0110	0.9360 ±0.0085	0.9390 ±0.0100
	Recall	0.9475 ±0.0115	0.9410 ±0.0115	0.9455 ±0.0110	0.9395 ±0.0145
	Specificity	0.9805 ±0.0040	0.9820 ±0.0030	0.9820 ±0.0025	0.9830 ±0.0030
	F_1	0.9390 ±0.0090	0.9385 ±0.0080	0.9405 ±0.0070	0.9390 ±0.0090
Sinus Bradycardia	Accuracy	0.9930 ±0.0015	0.9915 ±0.0025	0.9930 ±0.0015	0.9920 ±0.0015
	Precision	0.9885 ±0.0035	0.9915 ±0.0030	0.9900 ±0.0040	0.9870 ±0.0035
	Recall	0.9925 ±0.0020	0.9855 ±0.0075	0.9915 ±0.0035	0.9915 ±0.0035
	Specificity	0.9935 ±0.0020	0.9950 ±0.0015	0.9940 ±0.0025	0.9925 ±0.0020
	F_1	0.9905 ±0.0020	0.9885 ±0.0040	0.9905 ±0.0025	0.9890 ±0.0025
Sinus Rhythm/ Sinus Irregularity	Accuracy	0.9895 ±0.0015	0.9880 ±0.0035	0.9885 ±0.0020	0.9885 ±0.0015
	Precision	0.9820 ±0.0060	0.9720 ±0.0140	0.9730 ±0.0080	0.9740 ±0.0070
	Recall	0.9685 ±0.0075	0.9700 ±0.0070	0.9735 ±0.0065	0.9695 ±0.0065
	Specificity	0.9955 ±0.0015	0.9925 ±0.0040	0.9930 ±0.0020	0.9930 ±0.0020
	F_1	0.9750 ±0.0050	0.9710 ±0.0080	0.9730 ±0.0050	0.9720 ±0.0050

Table 6.7: Mean metrics and 2-σ (95.4%) confidence intervals obtained for **models based on different modifications of the proposed architecture based on the *superclasses* labelling regime**.



(a) **Accuracy**

(b) **Precision**

(c) **Sensitivity**

(d) **Specificity**

(e) $\mathbf{F}_1$

Figure 6.4: Mean values and 2-σ (95.4%) confidence intervals of performance metrics obtained for **models based on different modifications of the proposed architecture within the *reduced* labelling regime**.

Diagnostic Class	Metric	Model: Plain	Model: Multi- input	Model: Multi-output	Model: Multi-in-out
Overall	Accuracy	0.9260 ±0.0045	0.9195 ±0.0050	0.9245 ±0.0040	0.9220 ±0.0045
	Precision	0.9330 ±0.0045	0.9320 ±0.0045	0.9330 ±0.0045	0.9305 ±0.0045
	Recall	0.9285 ±0.0050	0.9180 ±0.0080	0.9290 ±0.0050	0.9230 ±0.0050
	Specificity	0.9890 ±0.0005	0.9890 ±0.0010	0.9890 ±0.0010	0.9885 ±0.0010
	F_1	0.9310 ±0.0030	0.9250 ±0.0045	0.9310 ±0.0035	0.9265 ±0.0035
Atrial Flutter	Accuracy	0.9675 ±0.0030	0.9675 ±0.0030	0.9700 ±0.0025	0.9670 ±0.0030
	Precision	0.9675 ±0.0030	0.9675 ±0.0030	0.9700 ±0.0025	0.9670 ±0.0030
	Recall	0.4890 ±0.0655	0.4100 ±0.0715	0.4870 ±0.0555	0.4120 ±0.0615
	Specificity	0.9890 ±0.0030	0.9920 ±0.0025	0.9915 ±0.0025	0.9920 ±0.0025
	F_1	0.6445 ±0.0575	0.5680 ±0.0695	0.6450 ±0.0490	0.5730 ±0.0600
Atrial Fibrillation	Accuracy	0.9690 ±0.0030	0.9665 ±0.0030	0.9690 ±0.0025	0.9690 ±0.0030
	Precision	0.9690 ±0.0030	0.9665 ±0.0030	0.9690 ±0.0025	0.9690 ±0.0030
	Recall	0.9240 ±0.0165	0.9390 ±0.0175	0.9355 ±0.0185	0.9360 ±0.0175
	Specificity	0.9780 ±0.0035	0.9720 ±0.0045	0.9760 ±0.0040	0.9755 ±0.0035
	F_1	0.9460 ±0.0085	0.9525 ±0.0090	0.9520 ±0.0095	0.9520 ±0.0090
Sinus Irregularity	Accuracy	0.9825 ±0.0020	0.9810 ±0.0025	0.9825 ±0.0020	0.9805 ±0.0025
	Precision	0.9825 ±0.0020	0.9810 ±0.0025	0.9825 ±0.0020	0.9805 ±0.0025
	Recall	0.6810 ±0.0610	0.7055 ±0.0465	0.6865 ±0.0475	0.6190 ±0.0655
	Specificity	0.9955 ±0.0020	0.9930 ±0.0020	0.9955 ±0.0020	0.9960 ±0.0015
	F_1	0.8015 ±0.0425	0.8195 ±0.0315	0.8070 ±0.0330	0.7555 ±0.0490
Sinus Bradycardia	Accuracy	0.9930 ±0.0015	0.9925 ±0.0015	0.9930 ±0.0015	0.9920 ±0.0015
	Precision	0.9930 ±0.0015	0.9925 ±0.0015	0.9930 ±0.0015	0.9920 ±0.0015
	Recall	0.9890 ±0.0035	0.9920 ±0.0025	0.9940 ±0.0020	0.9905 ±0.0050
	Specificity	0.9950 ±0.0015	0.9930 ±0.0020	0.9925 ±0.0020	0.9930 ±0.0025
	F_1	0.9910 ±0.0020	0.9925 ±0.0015	0.9935 ±0.0015	0.9915 ±0.0025
Sinus Rhythm	Accuracy	0.9815 ±0.0025	0.9800 ±0.0020	0.9815 ±0.0025	0.9800 ±0.0030
	Precision	0.9815 ±0.0025	0.9800 ±0.0020	0.9815 ±0.0025	0.9800 ±0.0030
	Recall	0.9620 ±0.0095	0.9360 ±0.0165	0.9535 ±0.0130	0.9575 ±0.0095
	Specificity	0.9855 ±0.0030	0.9890 ±0.0035	0.9870 ±0.0040	0.9845 ±0.0035
	F_1	0.9720 ±0.0050	0.9575 ±0.0090	0.9670 ±0.0070	0.9685 ±0.0050
Sinus Tachycardia	Accuracy	0.9875 ±0.0020	0.9860 ±0.0020	0.9860 ±0.0020	0.9865 ±0.0020
	Precision	0.9875 ±0.0020	0.9860 ±0.0020	0.9860 ±0.0020	0.9865 ±0.0020
	Recall	0.9595 ±0.0100	0.9485 ±0.0130	0.9490 ±0.0150	0.9465 ±0.0115
	Specificity	0.9920 ±0.0025	0.9925 ±0.0020	0.9925 ±0.0020	0.9935 ±0.0015
	F_1	0.9730 ±0.0050	0.9670 ±0.0070	0.9670 ±0.0075	0.9660 ±0.0060
Supra-ventricular Tachycardia	Accuracy	0.9830 ±0.0020	0.9815 ±0.0025	0.9825 ±0.0020	0.9810 ±0.0025
	Precision	0.9830 ±0.0020	0.9815 ±0.0025	0.9825 ±0.0020	0.9810 ±0.0025
	Recall	0.8850 ±0.0270	0.8230 ±0.0595	0.8780 ±0.0290	0.8945 ±0.0275
	Specificity	0.9885 ±0.0025	0.9905 ±0.0025	0.9880 ±0.0020	0.9860 ±0.0025
	F_1	0.9310 ±0.0150	0.8925 ±0.0350	0.9270 ±0.0165	0.9355 ±0.0150

Table 6.8: Mean metrics and 2-σ (95.4%) confidence intervals obtained for **models based on different modifications of the proposed architecture based on the *reduced* labelling regime**.

6.4.1 Discussion

One can see no significant difference in performance between the different variations of the proposed architecture. The only trend that might be inferred from the results is that the models taking ECG signals as well as global features as input performed slightly worse than those that do not, albeit not necessarily in a statistically significant way. Nevertheless, given these results, all modifications of the original proposed architecture can -at least within the scope of this study- be considered a likely waste of computational and data collection resources and were hence discarded in subsequent experiments.

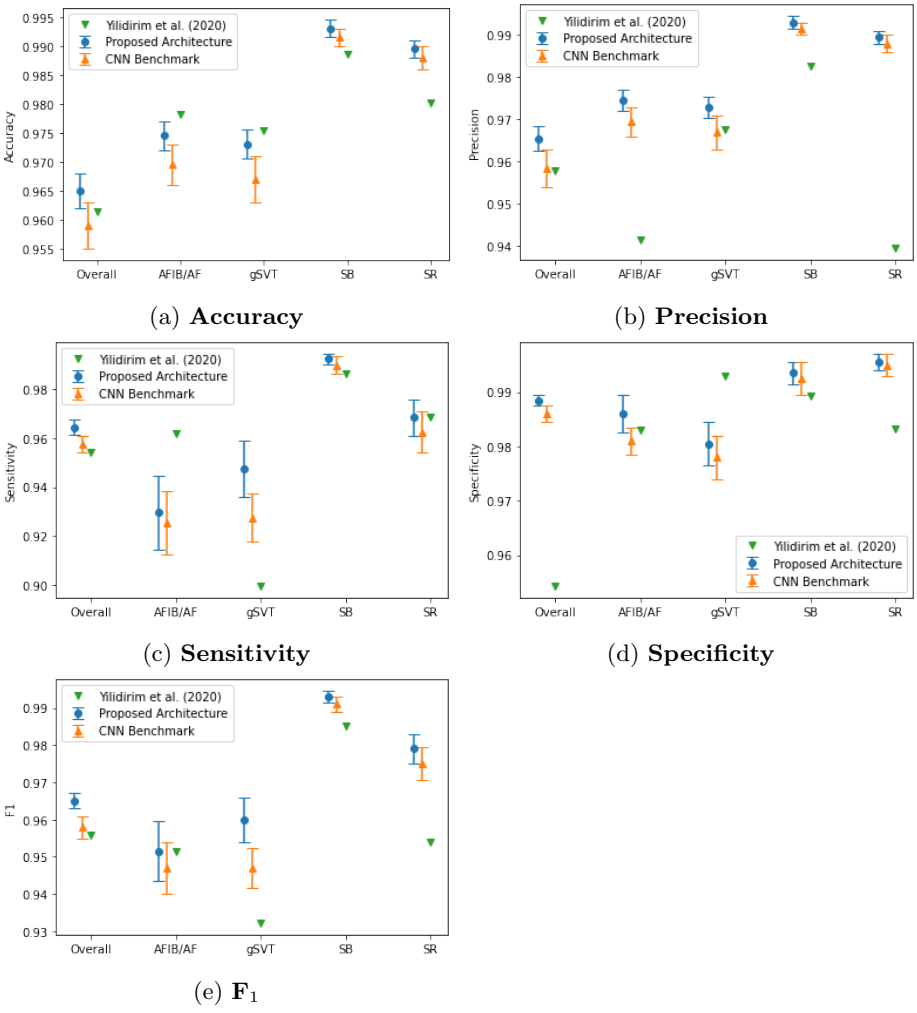
6.5 Validating the Proposed Architecture against the Chosen Benchmarks

In order to decide whether or not the proposed architecture marks an improvement in performance over alternative approaches (see **objective 3**, in this section, its performance compared to the benchmarks discussed in section 3.2.

As before, both the models based on the proposed architecture and those based on the CNN-benchmark were trained on a pre-processsed, balanced (by oversampling) training set drawn from the *Shaoxing* database. They were then evaluated on the corresponding Shaoxing test set. Results were obtained for both the *superclasses* and the *reduced classes* labelling regime and summarised in table 6.10 and 6.11 respectively. Furthermore, class-wise as well as overall comparisons were made for each metric in figures 6.5 and 6.6. Furthermore, average confusion matrices as well as ROC-curves are provided in sections A.1-A.2 as well as A.9 - A.10 of Appendix A. A mapping of the relevant combinations

Defining Characteristic	Experiment Number
CNN-Benchmark, Superclasses Regime	Experiment 1
CNN-Benchmark, Reduced Regime	Experiment 2
Proposed Architecture, Superclasses Regime	Experiment 9
Proposed Architecture, Reduced Regime	Experiment 10

Table 6.9: Mapping of defining characteristics and experiment numbers for section 6.5

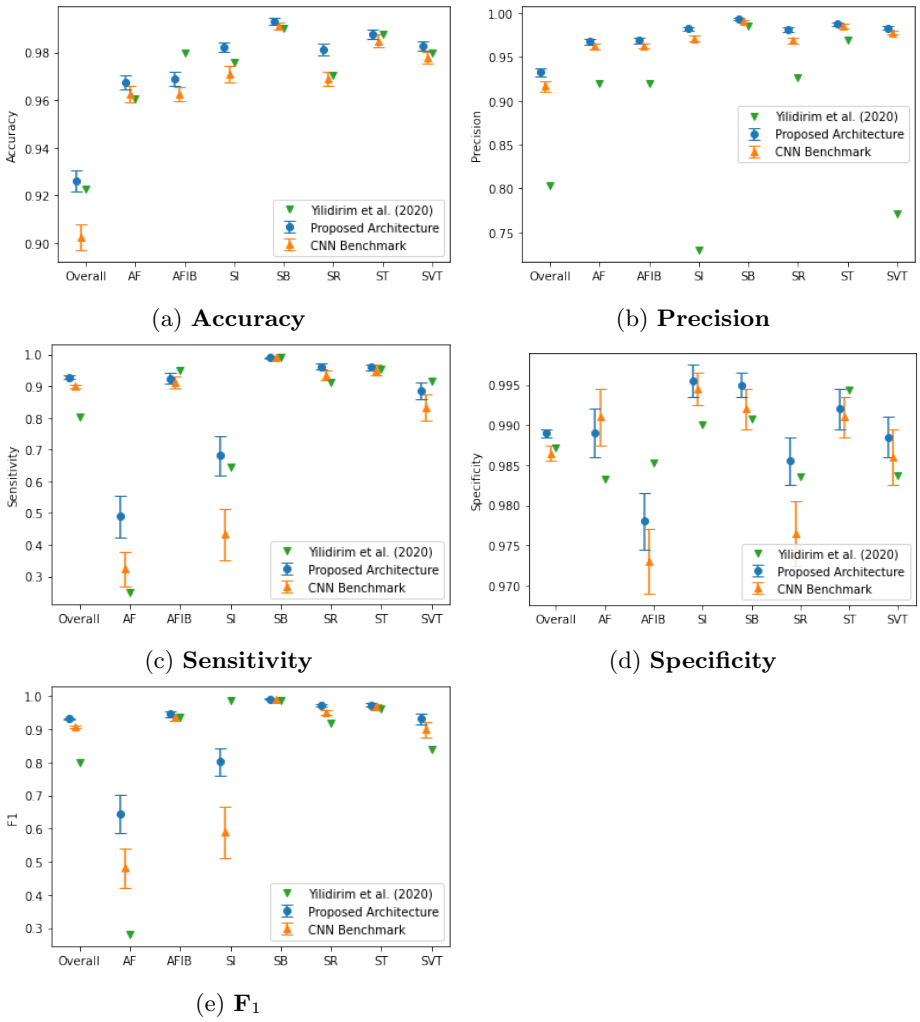


(a) **Accuracy** (b) **Precision**

(c) **Sensitivity** (d) **Specificity**

(e) $\mathbf{F}_1$

Figure 6.5: Mean values and 2-σ (95.4%) confidence intervals of performance metrics obtained for **models based on the proposed architecture as well as its benchmarks under the *superclasses* labelling regime**.



(a) **Accuracy** (b) **Precision**

(c) **Sensitivity** (d) **Specificity**

(e) $\mathbf{F}_1$

Figure 6.6: Mean values and 2-σ (95.4%) confidence intervals of performance metrics obtained for **models based on the proposed architecture as well as its benchmarks under the *reduced* labelling regime**.

Diagnostic Class	Metric	Architecture: Proposed	Architecture: CNN Benchmark.	Architecture: Yildirim et al.
Overall	Accuracy	0.9650 ±0.0030	0.9590 ±0.0040	0.9613
	Precision	0.9655 ±0.0030	0.9585 ±0.0045	0.9578
	Recall	0.9645 ±0.0030	0.9575 ±0.0035	0.9543
	Specificity	0.9885 ±0.0010	0.9860 ±0.0015	0.9543
	F_1	0.9650 ±0.0020	0.9580 ±0.0030	0.9557
Atrial Fibrillation/ Atrial Flutter	Accuracy	0.9745 ±0.0025	0.9695 ±0.0035	0.9782
	Precision	0.9745 ±0.0025	0.9695 ±0.0035	0.9416
	Recall	0.9295 ±0.0150	0.9255 ±0.0130	0.9617
	Specificity	0.9860 ±0.0035	0.9810 ±0.0025	0.9830
	F_1	0.9515 ±0.0080	0.9470 ±0.0070	0.9515
Generalised Supraventricular Tachycardia	Accuracy	0.9730 ±0.0025	0.9670 ±0.0040	0.9754
	Precision	0.9730 ±0.0025	0.9670 ±0.0040	0.9675
	Recall	0.9475 ±0.0115	0.9275 ±0.0100	0.8994
	Specificity	0.9805 ±0.0040	0.9780 ±0.0040	0.9930
	F_1	0.9600 ±0.0060	0.9470 ±0.0055	0.9322
Sinus Bradycardia	Accuracy	0.9930 ±0.0015	0.9915 ±0.0015	0.9886
	Precision	0.9930 ±0.0015	0.9915 ±0.0015	0.9825
	Recall	0.9925 ±0.0020	0.9900 ±0.0035	0.9865
	Specificity	0.9935 ±0.0020	0.9925 ±0.0030	0.9893
	F_1	0.9930 ±0.0015	0.9910 ±0.0020	0.9850
Sinus Rhythm/ Sinus Irregularity	Accuracy	0.9895 ±0.0015	0.9880 ±0.0020	0.9801
	Precision	0.9895 ±0.0015	0.9880 ±0.0020	0.9396
	Recall	0.9685 ±0.0075	0.9625 ±0.0085	0.9688
	Specificity	0.9955 ±0.0015	0.9950 ±0.0020	0.9832
	F_1	0.9790 ±0.0040	0.9750 ±0.0045	0.9540

Table 6.10: Mean metrics and 2-σ (95.4%) confidence intervals obtained for **models based on the proposed architecture and its benchmarks based on the *superclasses* labelling regime**.

Diagnostic Class	Metric	Architecture: Proposed	Architecture: CNN Benchmark.	Architecture: Yildirim et al.
Overall	Accuracy	0.9260 ±0.0045	0.9025 ±0.0055	0.9224
	Precision	0.9330 ±0.0045	0.9165 ±0.0060	0.8031
	Recall	0.9285 ±0.0050	0.9000 ±0.0060	0.8015
	Specificity	0.9890 ±0.0005	0.9865 ±0.0010	0.9872
	F_1	0.9310 ±0.0030	0.9080 ±0.0045	0.8004
Atrial Flutter	Accuracy	0.9675 ±0.0030	0.9625 ±0.0035	0.9607
	Precision	0.9675 ±0.0030	0.9625 ±0.0035	0.9202
	Recall	0.4890 ±0.0655	0.3240 ±0.0540	0.2500
	Specificity	0.9890 ±0.0030	0.9910 ±0.0035	0.9832
	F_1	0.6445 ±0.0575	0.4805 ±0.0600	0.2807
Atrial Fibrillation	Accuracy	0.9690 ±0.0030	0.9625 ±0.0030	0.9798
	Precision	0.9690 ±0.0030	0.9625 ±0.0030	0.9202
	Recall	0.9240 ±0.0165	0.9105 ±0.0190	0.9493
	Specificity	0.9780 ±0.0035	0.9730 ±0.0040	0.9853
	F_1	0.9460 ±0.0085	0.9355 ±0.0100	0.9345
Sinus Irregularity	Accuracy	0.9825 ±0.0020	0.9710 ±0.0035	0.9760
	Precision	0.9825 ±0.0020	0.9710 ±0.0035	0.7297
	Recall	0.6810 ±0.0610	0.4325 ±0.0815	0.6428
	Specificity	0.9955 ±0.0020	0.9945 ±0.0020	0.9900,
	F_1	0.8015 ±0.0425	0.5890 ±0.0770	0.9873
Sinus Bradycardia	Accuracy	0.9930 ±0.0015	0.9910 ±0.0015	0.9904
	Precision	0.9930 ±0.0015	0.9910 ±0.0015	0.9848
	Recall	0.9890 ±0.0035	0.9900 ±0.0035	0.9898
	Specificity	0.9950 ±0.0015	0.9920 ±0.0025	0.9907
	F_1	0.9910 ±0.0020	0.9905 ±0.0020	0.9873
Sinus Rhythm	Accuracy	0.9815 ±0.0025	0.9690 ±0.0030	0.9703
	Precision	0.9815 ±0.0025	0.9690 ±0.0030	0.9263
	Recall	0.9620 ±0.0095	0.9330 ±0.0150	0.9119
	Specificity	0.9855 ±0.0030	0.9765 ±0.0040	0.9835
	F_1	0.9720 ±0.0050	0.9505 ±0.0080	0.9190
Sinus Tachycardia	Accuracy	0.9875 ±0.0020	0.9850 ±0.0025	0.9875
	Precision	0.9875 ±0.0020	0.9850 ±0.0025	0.9693
	Recall	0.9595 ±0.0100	0.9500 ±0.0175	0.9518
	Specificity	0.9920 ±0.0025	0.9910 ±0.0025	0.9943
	F_1	0.9730 ±0.0050	0.9670 ±0.0090	0.9604
Supra-ventricular Tachycardia	Accuracy	0.9830 ±0.0020	0.9780 ±0.0025	0.9798
	Precision	0.9830 ±0.0020	0.9780 ±0.0025	0.7714
	Recall	0.8850 ±0.0270	0.8335 ±0.0405	0.9152
	Specificity	0.9885 ±0.0025	0.9860 ±0.0035	0.9837
	F_1	0.9310 ±0.0150	0.8990 ±0.0235	0.8372

Table 6.11: Mean metrics and 2-σ (95.4%) confidence intervals obtained for **models based on the proposed architecture and its benchmarks based on the *reduced* labelling regime**.

of architectures and labelling regimes to the corrsepconding experiment numbers is provided in table 6.9.

6.5.1 Discussion

We notice that while the proposed architecture outperforms the naive CNN-benchmark as well as the DNN-LSTM-benchmark proposed by Yildirim et al. (2020) overall and with regards to most classes in both labelling regimes, there are a few classes for which the picture is not so clear. In particular, the AFIB/AF and gSVT classes in the *superclass* regime as well as the SI, and SVT classes in the *reduced* regime show a high variety of different model outperformances. Specifically, we notice that the Yildirim et al. (2020) model outperforms the proposed models as well as the benchmark in the ST class and that the proposed architecture shows a larger positive performance differential with regards to the benchmark for AFIB/AF and gSVT (a potential explanation will be discussed in section 6.8).

With regards to the *reduced classes* regime one has to further keep in mind that there is a considerable class imbalance in the *reduced classes* labelling regime and that we would therefore expect a strong underperformance in the minority classes. Looking at *multiclass* confusion matrices in section A.2 and A.10, we notice that most of the misclassified examples result from one of the following:

- AF classified as AFIB
- SI classified as SR or (less prominently) SB
- SVT being missclassied ad ST or AF

These findings not only makes sense with regards to the morphological and rhythmic patterns discussed in section 2.2.2, but also in light of the fact that AF, SI and SVT are the most underrepresented classes in this labelling regime (compare table 4.5. The obvious conclusion is that when dealing with classes that are inherently difficult to distinguish -i.e. because the belong to the same "family" of arrhythmia or, as in the case of SVT and AFIB, one is on fact a subset of the other- adjusting the class balance of the training set or even oversampling the relevant classes could significantly boost performance.

Defining Characteristic	Experiment Number
Benchmark, Shaoxing	Experiment 1
Benchmark, Shaoxing 20s	Experiment 4
Benchmark, Shaoxing 50s	Experiment 5
Model, Shaoxing	Experiment 9
Model, Shaoxing 20s	Experiment 12
Model, Shaoxing 50s	Experiment 13

Table 6.12: Mapping of defining characteristics and experiment numbers for section 6.6

6.6 Validating the Proposed Architecture on Augmented Data

In order to scrutinise the architectures' ability to generalise to sequences longer than those seen during training (see **objective 4**), models based on the proposed architecture as well as the benchmark, trained on the original balanced (oversampled) and pre-processed training data drawn from the *Shaoxing* database under the *superclasses* labelling regime were evaluated on the two "augmented" test sets described in section 5.4.

The resulting performance metrics for sequences of length 20 and 50 are summarised in tables 6.13 and 6.14, while figure 6.7 illustrates the change in overall and class-wise performance with varying sequence lengths. ROC-curves and mean confusion matrices are further available in sections A.4-A.5 and A.12-A.13. A mapping of the relevant combinations of architecture and test data to the corresponding experiment numbers is provided in table 6.12.

6.6.1 Discussion

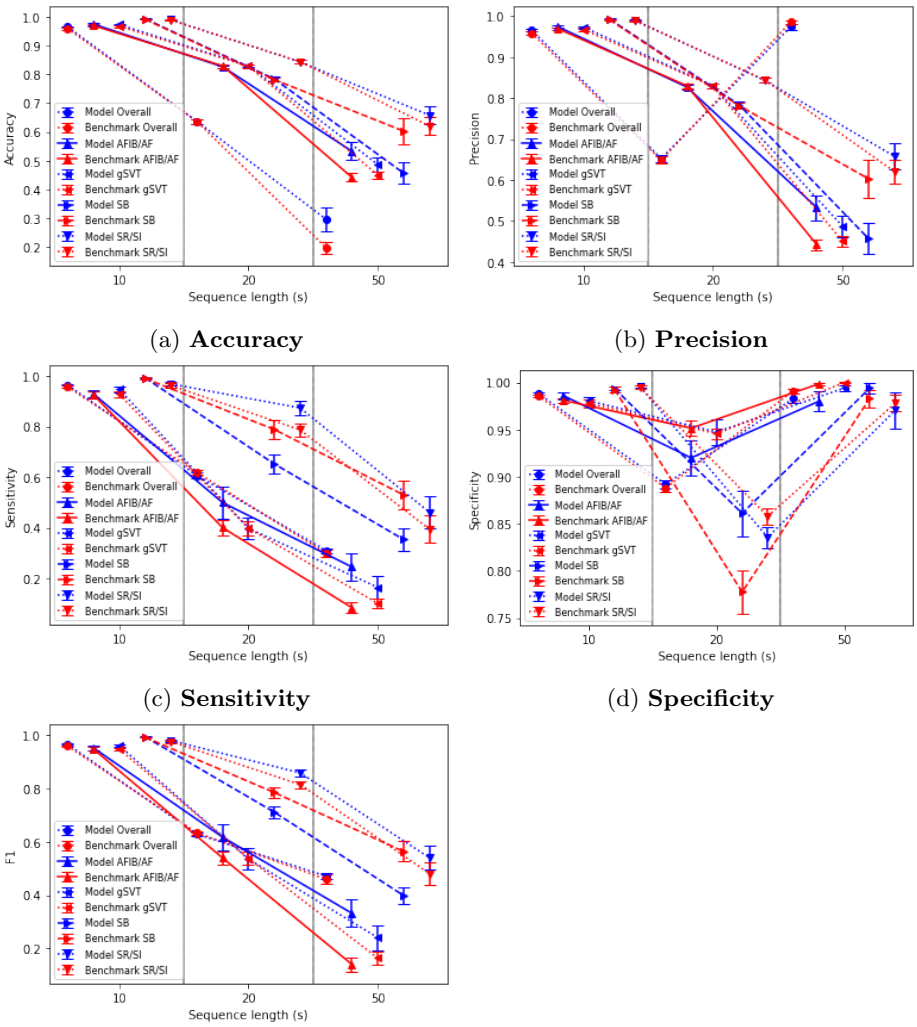
The most immediate and unsurprising observation is that the performance of classifiers based on both the proposed and the benchmark architecture generally deteriorate with increasing sequence length (and hence, degree of mixed labelling). The obvious exception here is specificity (see figure 6.7d), which increases at 50s back to the level achieved on the original test set. However, it is important note that this is is most likely due to the classifiers having made a slightly different trade-off between sensitivity and specificity (at threshold 0.5).

Diagnostic Class	Metric	Architecture: Bechmark	Architecture: Proposed
Overall	Accuracy	0.5890 ±0.0080	0.5975 ±0.0095
	Precision	0.6485 ±0.0085	0.6525 ±0.0095
	Recall	0.6190 ±0.0095	0.6075 ±0.0115
	Specificity	0.6190 ±0.0095	0.8920 ±0.0045
	F_1	0.6335 ±0.0065	0.6290 ±0.0075
Atrial Fibrillation/ Atrial Flutter	Accuracy	0.8295 ±0.0060	0.8265 ±0.0055
	Precision	0.8295 ±0.0060	0.8265 ±0.0055
	Recall	0.4005 ±0.0285	0.4995 ±0.0635
	Specificity	0.9515 ±0.0080	0.9200 ±0.0190
	F_1	0.5390 ±0.0255	0.6170 ±0.0490
Generalised Supraventricular Tachycardia	Accuracy	0.8285 ±0.0055	0.8300 ±0.0060
	Precision	0.8285 ±0.0055	0.8300 ±0.0060
	Recall	0.3985 ±0.0275	0.3980 ±0.0440
	Specificity	0.9455 ±0.0060	0.9475 ±0.0135
	F_1	0.5370 ±0.0250	0.5350 ±0.0400
Sinus Bradycardia	Accuracy	0.7815 ±0.0060	0.7840 ±0.0070
	Precision	0.7815 ±0.0060	0.7840 ±0.0070
	Recall	0.7885 ±0.0365	0.6525 ±0.0375
	Specificity	0.7775 ±0.0230	0.8610 ±0.0245
	F_1	0.7840 ±0.0185	0.7105 ±0.0225
Sinus Rhythm/ Sinus Irregularity	Accuracy	0.8440 ±0.0055	0.8430 ±0.0060
	Precision	0.8440 ±0.0055	0.8430 ±0.0060
	Recall	0.7870 ±0.0250	0.8735 ±0.0285
	Specificity	0.8580 ±0.0085	0.8355 ±0.0110
	F_1	0.8140 ±0.0135	0.8575 ±0.0140

Table 6.13: Mean values and 2-σ (95.4%) confidence intervals of performance metrics obtained for **models based on the proposed architecture as well as the benchmark on synthetic sequences of length 20s as compared to performance on the original test set**.

Diagnostic Class	Metric	Architecture: Bechmark	Architecture: Proposed
Overall	Accuracy	0.1980 ±0.0210	0.2970 ±0.0425
	Precision	0.9855 ±0.0035	0.9740 ±0.0070
	Recall	0.2995 ±0.0160	0.3105 ±0.0100
	Specificity	0.9910 ±0.0025	0.9830 ±0.0050
	F_1	0.4595 ±0.0185	0.4710 ±0.0115
Atrial Fibrillation/ Atrial Flutter	Accuracy	0.4425 ±0.0135	0.5330 ±0.0305
	Precision	0.4425 ±0.0135	0.5330 ±0.0305
	Recall	0.0845 ±0.0200	0.2455 ±0.0550
	Specificity	0.9985 ±0.0010	0.9800 ±0.0105
	F_1	0.1405 ±0.0275	0.3320 ±0.0515
Generalised Supraventricular Tachycardia	Accuracy	0.4505 ±0.0135	0.4870 ±0.0260
	Precision	0.4505 ±0.0135	0.4870 ±0.0260
	Recall	0.1015 ±0.0200	0.1650 ±0.0430
	Specificity	1.0000 ±0.0005	0.9940 ±0.0035
	F_1	0.1640 ±0.0265	0.2410 ±0.0470
Sinus Bradycardia	Accuracy	0.6030 ±0.0460	0.4570 ±0.0370
	Precision	0.6030 ±0.0460	0.4570 ±0.0370
	Recall	0.5300 ±0.0560	0.3540 ±0.0445
	Specificity	0.9830 ±0.0095	0.9940 ±0.0055
	F_1	0.5635 ±0.0375	0.3985 ±0.0315
Sinus Rhythm/ Sinus Irregularity	Accuracy	0.6205 ±0.0300	0.6580 ±0.0320
	Precision	0.6205 ±0.0300	0.6580 ±0.0320
	Recall	0.3950 ±0.0530	0.4615 ±0.0635
	Specificity	0.9780 ±0.0085	0.9705 ±0.0195
	F_1	0.4805 ±0.0405	0.5400 ±0.0450

Table 6.14: Mean values and 2-σ (95.4%) confidence intervals of performance metrics obtained for **models based on the proposed architecture as well as the benchmark on synthetic sequences of length 50s as compared to performance on the original test set**.



(a) **Accuracy**

(b) **Precision**

(c) **Sensitivity**

(d) **Specificity**

(e) $\mathbf{F}_1$

Figure 6.7: Mean values and 2-σ (95.4%) confidence intervals of performance metrics obtained for **models based on the proposed architecture as well as the benchmark on synthetic sequences of different length as compared to performance on the original test set**.

In conjunction with the confusion matrices, which indicate that both benchmark and proposed architecture classifiers become increasingly more prone to false negatives as the sequence lengths increase, this again suggests the conclu-

sion that the classifiers might have in fact learned a notion of mutual exclusivity of classes.

What is reassuring, however, is that based on the available information, there appears to be significant skill involved in the predictions made by all classifiers. Furthermore, there appears to be a gradual decline of performance with sequence length (compared to the original test set) as well as, perhaps more importantly, a widening differential in the performance of proposed-architecture classifiers and benchmark classifiers in figures figure 6.7 (performance on *Sinus Bradycardia* being the obvious exception), which supports our hypothesis that -given appropriate and representative training data-, transformer-based models have superior generalisation capabilities.

6.7 Validating the proposed architecture on the PTB-XL Database

In order to determine our proposed architecture's ability to generalise to data drawn from a different source (see **objective 5**), models based on our proposed architecture as well as the benchmark were trained on raw data from the Shaoxing training set and then tested on a test set drawn from the PTB-XL database. The reason for the lack of pre-processing was mainly the fact that the PTX-XL data is only available in raw form. Had we introduced our own method, we would have made the results less useful for later comparison by other researchers.

Additionally, since we did not see much difference in performance between models trained on the raw- and pre-processed classfiers to begin with (compare section 6.3), it is our belief that any potential performance gains would have been far outweighed by the results loss of generality.

Defining Characteristic	Experiment Number
Benchmark, Shaoxing	Experiment 3
Proposed Architecture, Shaoxing	Experiment 11

Table 6.15: Mapping of defining characteristics and experiment numbers for section 6.4

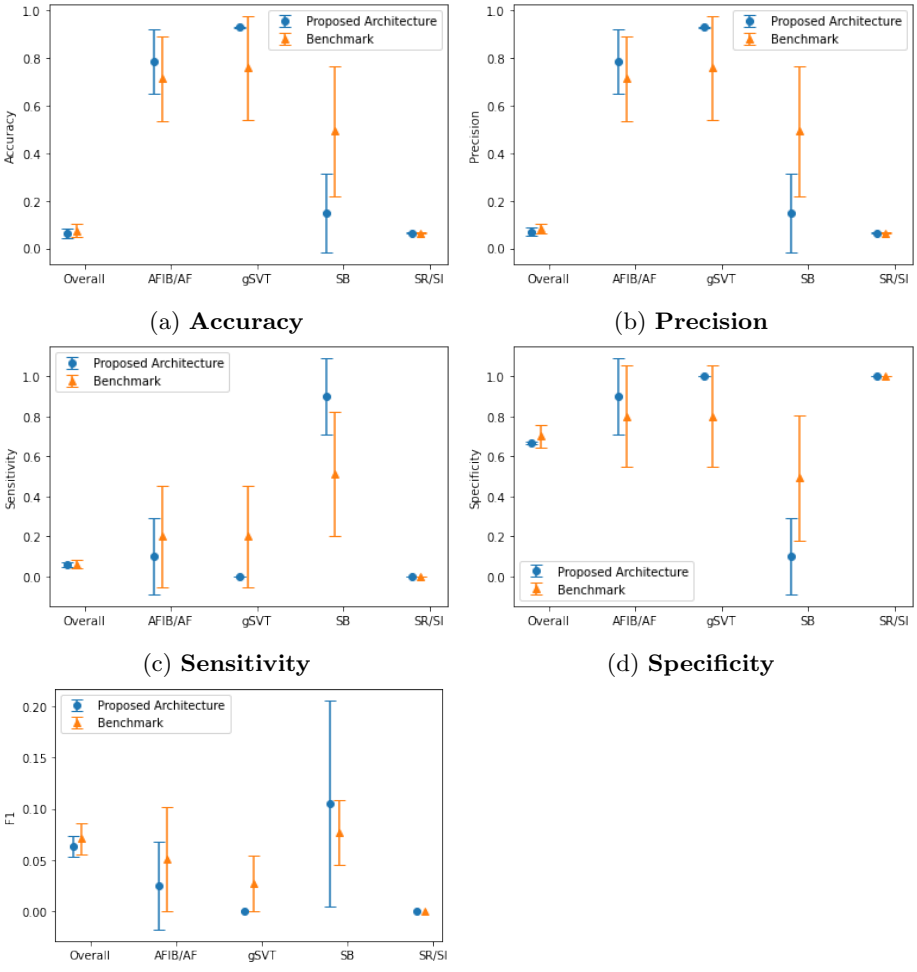
Diagnostic Class	Metric	Architecture: Benchmark	Model: Proposed
Overall	Accuracy	0.0750 ±0.0260	0.0620 ±0.0200
	Precision	0.0825 ±0.0210	0.0705 ±0.0160
	Recall	0.0620 ±0.02050.0580 ± 0.0135	
	Specificity	0.7010 ±0.0580	0.6660 ±0.0065
	F_1	0.0710 ±0.0155	0.0635 ±0.0105
Atrial Fibrillation/ Atrial Flutter	Accuracy	0.7120 ±0.1790	0.7830 ±0.1340
	Precision	0.7120 ±0.1790	0.7830 ±0.13400
	Recall	0.2000 ±0.2530	0.1000 ±0.1895
	Specificity	0.8000 ±0.2530	0.9000 ±0.1895
	F_1	0.0510 ±0.0505	0.0255 ±0.0430
Generalised Supraventricular Tachycardia	Accuracy	0.7565 ±0.2165	0.9275 ±0.0025
	Precision	0.7565 ±0.2165	0.9275 ±0.0025
	Recall	0.2000 ±0.2530	0.0000 ±0.0000
	Specificity	0.8000 ±0.2530	1.0000 ±0.0000
	F_1	0.0270 ±0.0270	0.0000 ±0.0000
Sinus Bradycardia	Accuracy	0.4930 ±0.2730	0.1495 ±0.1660
	Precision	0.4930 ±0.2730	0.1495 ±0.1660
	Recall	0.5105 ±0.3105	0.9000 ±0.1895
	Specificity	0.4920 ±0.3115	0.1000 ±0.1895
	F_1	0.0770 ±0.0315	0.1050 ±0.1000
Sinus Rhythm/ Sinus Irregularity	Accuracy	0.0650 ±0.0020	0.0650 ±0.00205
	Precision	0.0650 ±0.0020	0.0650 ±0.0020
	Recall	0.0000 ±0.0000	0.0000 ±0.0000
	Specificity	1.0000 ±0.0000	1.0000 ±0.0000
	F_1	0.0000 ±0.0000	0.0000 ±0.0000

Table 6.16: Mean metrics obtained for **models based on the proposed architecture as well as the CNN-benchmark on the PTB-XL database**.

After converting the labels according to table 4.7., we drew a stratified sample of 5000 examples in order to validate our model. Table 6.16 gives a comprehensive overview of the performance metrics obtained, and figure 6.8 makes class-wise comparisons for each metric. ROC-curves and mean confusion matrices are available in sections A.3 and A.11. The experiment numbers corresponding to the benchmark and the proposed architecture are 3 and 11 respectively (see table 6.15).

6.7.1 Discussion

We can see that the overall performance for both model and benchmark is very poor and in fact, if we consider the fact that there is a total of 16 possible label-configurations, close to random guessing. This conclusion is supported by the ROC-curves, which indicate that there is nearly no skill in the classfiers' predictions for the AFIB/AF class, and very little for the other three classes.



(a) **Accuracy** (b) **Precision**

(c) **Sensitivity** (d) **Specificity**

(e) $\mathbf{F}_1$

Figure 6.8: Mean values and 2-σ confidence intervals of performance metrics obtained for **models based on the proposed architecture as well as the CNN-benchmark on the PTB-XL dataset**.

Yet, based purely on accuracy (figure 6.8a), one might be tempted to conclude that both types of models are actually quite good at detecting AFIB, and gSVT. However, looking at the other metrics we must realise that while precision and specificity are high, both of these metrics are on the same side of the trade-off and the corresponding sensitivity is nearly zero[4]. The correct interpretation is of course that the classifiers simply do not classify any examples (or nearly none) as AFIB or gSVT, reaping high precision and accuracy as a result.

A look at the confusion matrices further reveals that the classifier has simply classified the vast majority of examples as non-AFIB/AF, all as non-gSVT and non-SR/SI and the vast majority as SB. One possible explanation for this result might be a poor mapping in the design of the experiment and/or the inability of the classifier to deal with mixed-label examples which are not part of the training set, and it can also not be excluded that there are significant systematic differences between the populations from which the two databases were sampled. However, neither of these hypothesis provide an explanation for why both types of classifiers show such an extremely one-sided bias.

Although the strong proclivity of the classifiers to predicit a positive label for SB seems at odds with this , another possible explanation for the poor performance could be the aforementioned hypothesis that due to the lack of mixed labels in the training set, the classifiers have actually *learned* that classes are mutually exclusive. As a result, in a mixed-multi-label situation, most classes would get rejected simply because the classifier has a higher conviction on another class.

6.8 Evaluating Model Interpretability through Attention Mapping

As stated before, part of the guiding principles that informed the development of the proposed architecture were the desiderata for medical machine learning laid out in section 2.5.4. In particular, one reason for choosing to investigate the

[4]In fact, one of the error-bars erroneously indicates a value of below zero, which reminds us that *binomial proportion confidence intervals* are a mere approximation and can produce artifacts called *overshoot* at the edges of bounded quantities (Newcombe, 1998).

transformer-based architecture was the presence of an attention-mechanism operating directly on CNN-features instead of the output of LSTM- or GRU-layers like in architectures inspired by the encoder-decoder architecture (i.e. Zhang and Li, 2021). The hypothesis is that this leads to a potentially high degree of potential insight into the relationships *learned* by the model through joint analysis of CNN-filter-banks and attention scores. While a full-scaled analysis of the interpretability of models based on the proposed architecture is beyond the scope of this work, this section is intended to provide an anecdotal insight into the interpretability of models based on the proposed architecture (see **objective 6**) and thereby illustrate the potential for further research in this area.

For this purpose, one example from our test set drawn from the *Shaoxing* database was chosen for each class of the *superclasses* labelling regime based on the fact that these have been classified successfully by an overwhelming majority of the classifiers here discussed[5]. Then, the first realisation of the ten classifiers trained on a balanced (oversampled) training set drawn from the *Shaoxin* database using the *superclasses* labelling regime (i.e. Experiment 9 in table 5.1; ROC curves and confusion matrices available in section A.9 of Appendix A) was used to make a prediction on each of the examples. The corresponding matrices *attention scores*, i.e. $\text{softmax}(Q^T K/\sqrt{d_{keys}})$ (compare equation 2.6) for each of the ten *attention heads* were then extracted and displayed in figures 6.9 - 6.12.

6.8.1 Discussion

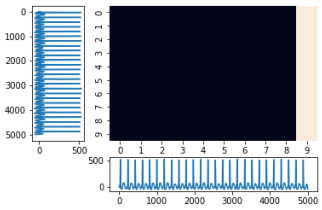
Although the resulting patterns are much less expressive than what we might have hoped for based on the results presented by Vaswani et al. (2017) in the context of *machine translation*, we can nevertheless identify some subtle differences between the different examples.

Without claiming generality, we notice that the pattern for SR/SI and SB, as well as to some degree gSVT are more "simple" in the sense that there appears to be no or few distinctive connections drawn between different parts of the sequence and if there are, they appear less nuanced than in the case of AFBI/AF. It would be interesting to show these to a clinical expert in order

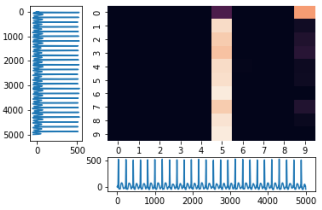
[5]The underlying assumption is that these represent "iconic" examples of their respecitve classes

to decide whether the patterns are reasonably interpretable within a clinically valid diagnostic framework.

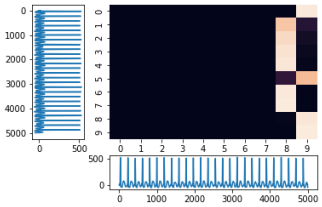
Based on the results of the previous experiments, it is our own (albeit speculative) hypothesis that in many classifications under the given labelling regimes, the *encoder* is not actually needed and the information contained in the *feature maps* obtained by the CNN module are simply "passed through" to the classifier. This would certainly explain the more differentiated patterns emerging for AFIB/AF and (to a lesser degree) gSVT and is supported by the observation that the proposed architecture's outperformance against the benchmarks is most pronounced for these classes (compare section 6.5).



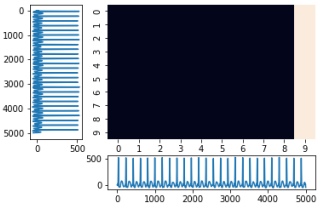
(a) Head 1



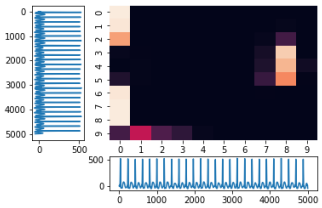
(b) Head 2



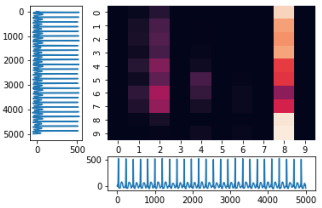
(c) Head 3



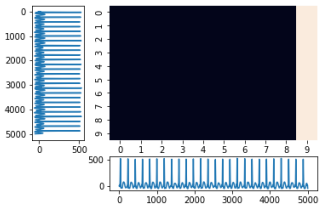
(d) Head 4



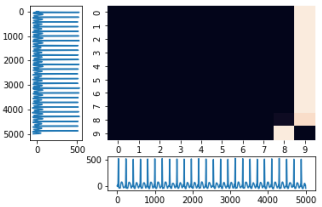
(e) Head 5



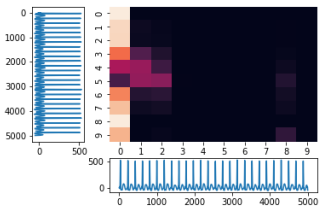
(f) Head 6



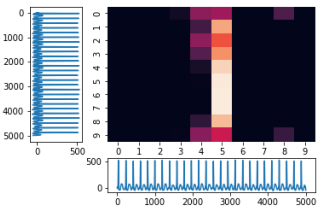
(g) Head 7



(h) Head 8

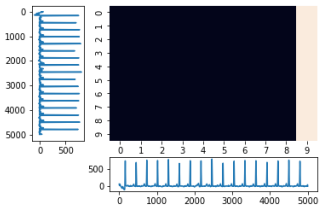


(i) Head 9

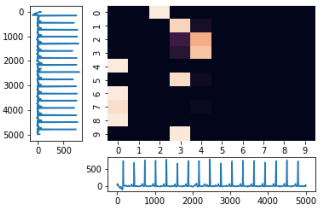


(j) Head 10

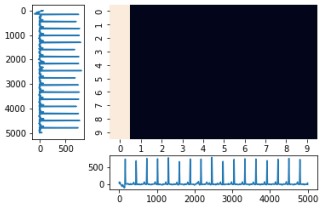
Figure 6.9: Attention scores for AFIB/AF.



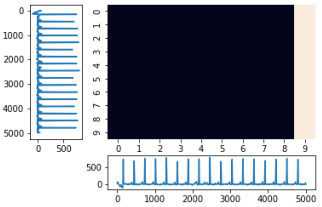
(a) Head 1



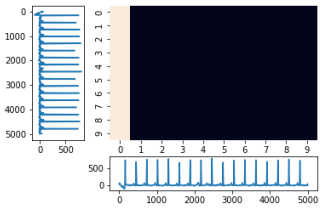
(b) Head 2



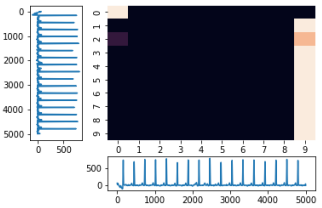
(c) Head 3



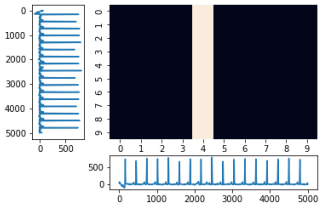
(d) Head 4



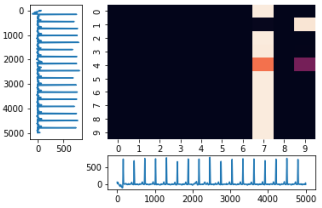
(e) Head 5



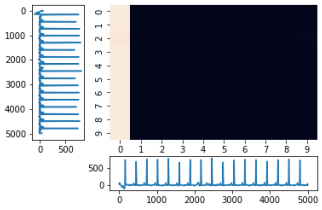
(f) Head 6



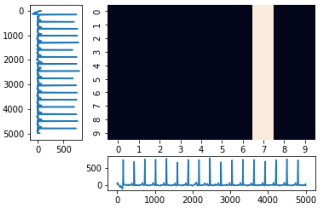
(g) Head 7



(h) Head 8

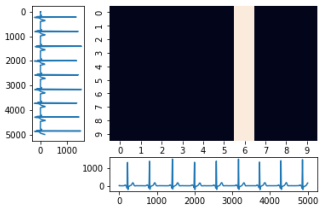


(i) Head 9

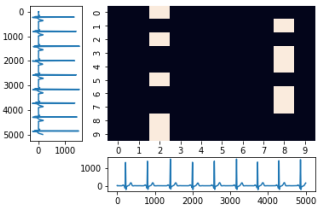


(j) Head 10

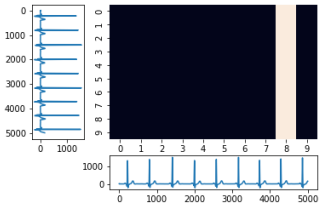
Figure 6.10: Attention scores for gSVT



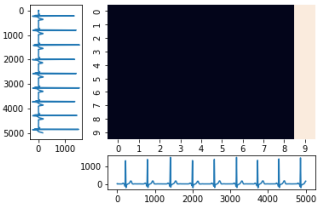
(a) Head 1



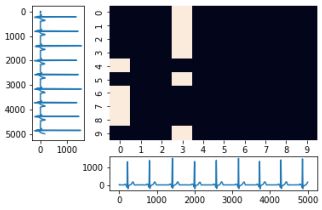
(b) Head 2



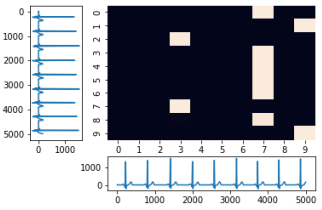
(c) Head 3



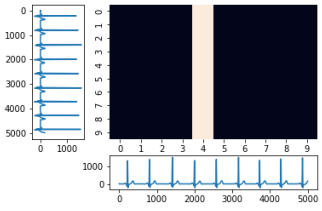
(d) Head 4



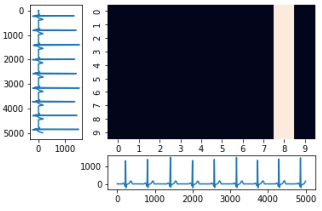
(e) Head 5



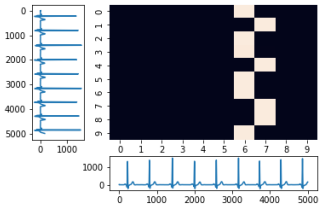
(f) Head 6



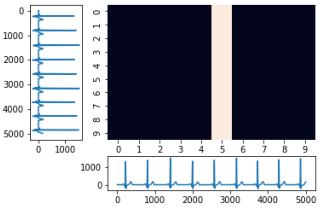
(g) Head 7



(h) Head 8

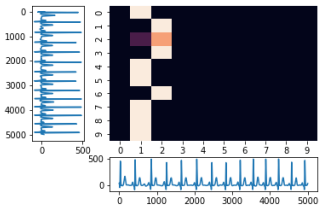


(i) Head 9

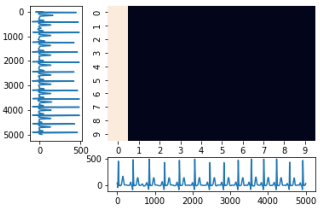


(j) Head 10

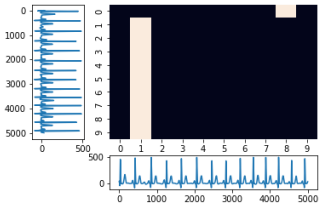
Figure 6.11: Attention scores for SB.



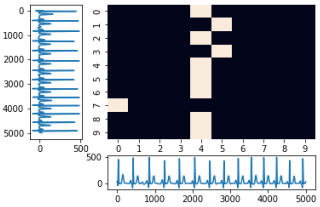
(a) Head 1



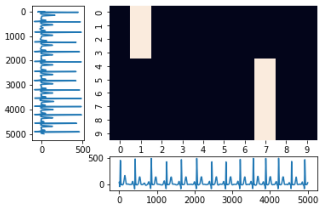
(b) Head 2



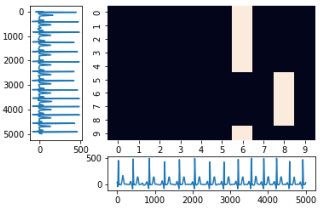
(c) Head 3



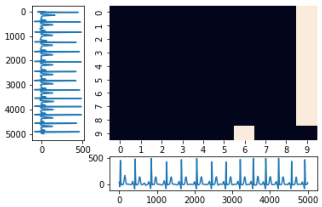
(d) Head 4



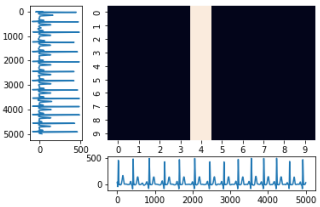
(e) Head 5



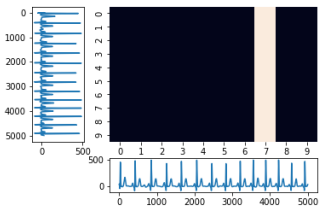
(f) Head 6



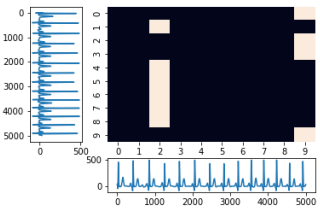
(g) Head 7



(h) Head 8



(i) Head 9



(j) Head 10

Figure 6.12: Attention scores for SR.

Chapter 7

Conclusion and Outlook

Based on the the experiments discussed in the previous section, we can first and foremost conclude that our proposed algorithm benefits from both pre-processing and a training set over-sampled to the majority class, if only marginally and not necessarily in a statistically significant way (**objective 1**).

Our evaluations of the proposed and benchmark architectures on the *reduced classes* labelling regime further showed the detrimental effect of the presence of *minority classes* on performance, particularly when these are easily confusable with other classes based on the nature of their features (compare section 6.5). If balancing cannot be achieved due to varying clinical incidence, the use of advanced data-augmenting techniques such as *generative adversarial networks* (Goodfellow et al., 2020) for creating synthetic examples might be indicated.

While the proposed modifications were deemed largely a waste of computational resources (**objective 2**), our experiments show that the transformer-inspired architecture proposed in this work can compete with other state-of-the-art *deep learning* architectures such as a CNN-LSTM and a plain CNN and at times outperform them (**objective 3**).

With regards to the ability to generalise to longer sequences and mixed labels (**objective 4**), it has become extraordinarily clear that in order to achieve acceptable performance in these areas, one has to include similar experiences in the training data. Despite this hardly surprising result, it is interesting

(although again not exceptionally surprising) to see that model performance decreases gradually with increasing sequence length. This suggests that at sequence lengths closer to the length of the training sequences, the models are still able to apply some of the predictive power acquired during training.

Another reassuring result based on the results presented in section 6.6 is the fact that at a sequence length of 50 seconds, the models based on the proposed architecture have outperformed those based on the benchmark architecture significantly, and by a larger margin than for shorter sequence lengths. This suggests that our baseline assumption of the superior ability of transformer-type architectures to generalise and handle long sequences might be correct, which is further supported by our informal observation that during training, the *generalisation gap* between performance on the training and test set was generally larger for the benchmark, indicating greater overfitting of the latter. We hope that in the future, this result can be replicated and expanded upon.

An alternative, more rigorous approach to the data augmentation technique outlined in section 5.4 and adopted in creating the test data used in experiments 4,5,12 and 13 would have been the use of *deep generative models* such as the aforementioned *generative adversarial networks* or *variational autoencoders* (Kingma, Welling, et al., 2019). This could form the basis of future work.

With regards to the proposed architecture's ability to generalise to data drawn from other sources (**objective 5**), the results were indeed disappointing. However, due to the poor performance of both the benchmark and the proposed architecture, it cannot be concluded that the proposed architecture is uniquely inferior in this regard. Rather, as discussed before, one might hypothesise that the cause for the poor performance, besides a lack of generalisation, are to be found in an incorrect mapping of diagnostic classes, mixed labels, vastly differing levels of signal quality[1], systematic differences in the underlying populations and/or the presence of unseen artifacts in the PTB-XL database. In any case, further research is needed to answer these questions.

Regarding the *interpretability* desideratum as laid out in section 2.5.4 and

[1]In fact, the signal quality labels that are part of the original PTB-XL annotations had to be ignored due to the constraints of the Shaxing labelling regime.

manifested in **objective 6**, we note that the proposed architecture does indeed provide insight into the relationships used for classification, although to a lesser degree than was expected. It cannot be concluded from our analysis whether the patterns of the sort shown in figures 6.9 - 6.12 from models based on our proposed architecture have any clinical significance, and in order to come to a conclusion, a representative set of these images would have to be shown to a panel of experienced clinicians in order to be evaluated with regards to their validity.

In any case, as hypothesised in section 6.8.1, it is possible that the *encoder module* is not actually needed for many of the individual classification tasks we confronted the models with. This hypothesis is supported by the very thin (yet in some cases statistically significant) margin of outperformance shown by the proposed architecture against the benchmarks. One might conclude that the given classification tasks (not including those on external examples on long sequences) are in a sense too "easy", and that the *bayes error*[2] can be approached to a reasonable degree by simpler classifiers such as out CNN-benchmark. It would therefore be interesting to see how the proposed architecture would perform on longer sequences and/or finer labelling given appropriate training data. Another interesting question for further research along this line would be to look more closely at the attention scores associated with those examples that tended to be misclassified by the benchmark but classified correctly by the models based on the proposed architecture, in order uncover potential underlying patterns.

In closing, we would like to stress that the results presented in sections 6.7 and 6.6 are very good examples of the importance of *model risk* and the need for robust *confidence estimates* when pushing classifiers towards the edge of their ability to generalise. The expanding error-bars in figure 6.7 are testimony to this fact. An important question to answer would be, for example, whether methods of *approximate bayesian inference* (i.e. *monte carlo dropout*) and/or ensemble techniques (as discussed in section 2.5.4) would have been able to detect the exploding *model risk* that emerged with sequences that were longer and label combinations that were much unlike those seen during training.

[2]This refers to the theoretical maximum performance a classifier can reach for a given task (Goodfellow et al., 2016).

Bibliography

Acharya, U. R., Oh, S. L., Hagiwara, Y., Tan, J. H., Adam, M., Gertych, A., & San Tan, R. (2017). A deep convolutional neural network model to classify heartbeats. *Computers in biology and medicine*, *89*, 389–396.

Alday, E. A. P., Gu, A., Shah, A. J., Robichaux, C., Wong, A.-K. I., Liu, C., Liu, F., Rad, A. B., Elola, A., Seyedi, S., et al. (2020). Classification of 12-lead ecgs: The physionet/computing in cardiology challenge 2020. *Physiological measurement*, *41*(12), 124003.

Al-Shoshan, A., & Al-Shoshan, A. (2019). Biomedical application of the evolutionary higher-order spectrum. *Proceedings of 32nd International Conference on*, *63*, 11–20.

Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., & Mané, D. (2016). Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*. https://arxiv.org/pdf/1606.06565.pdf

Antoniades, C., Asselbergs, F. W., & Vardas, P. (2021). The year in cardiovascular medicine 2020: Digital health and innovation. *European Heart Journal*.

Antzelevitch, C., & Burashnikov, A. (2011). Overview of basic mechanisms of cardiac arrhythmia. *Cardiac electrophysiology clinics*, *3*(1), 23–45.

Asirvatham, S. J., & Stevenson, W. G. (2016). Multiple and concurrent arrhythmia. *Circulation: Arrhythmia and Electrophysiology*, *9*(7), e003612.

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Barlow, R. J. (1993). *Statistics: A guide to the use of statistical methods in the physical sciences* (Vol. 29). John Wiley & Sons.

Bashar, S. K., Ding, E., Walkey, A. J., McManus, D. D., & Chon, K. H. (2019). Noise detection in electrocardiogram signals for intensive care unit patients. *IEEE Access*, *7*, 88357–88368.

Baydogan, M. G., Runger, G., & Tuv, E. (2013). A bag-of-features framework to classify time series. *IEEE transactions on pattern analysis and machine intelligence*, *35*(11), 2796–2802.

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of machine learning research*, *13*(2).

Bishop, C. M., & Nasrabadi, N. M. (2006). *Pattern recognition and machine learning* (Vol. 4). Springer.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Buerger, C., von Berg, J., Franz, A., Klinder, T., Lorenz, C., & Lenga, M. (2020). Combining deep learning and model-based segmentation for labeled spine ct segmentation. *Medical Imaging 2020: Image Processing*, *11313*, 307–314.

Caruana, R. (1997). Multitask learning. *Machine learning*, *28*(1), 41–75.

Che, Z., Purushotham, S., Khemani, R., & Liu, Y. (2016). Interpretable deep models for icu outcome prediction. *AMIA annual symposium proceedings*, *2016*, 371.

Chowdhury, R. H., Reaz, M. B., Ali, M. A. B. M., Bakar, A. A., Chellappan, K., & Chang, T. G. (2013). Surface electromyography signal processing and classification techniques. *Sensors*, *13*(9), 12431–12466.

Clayton, R., Bernus, O., Cherry, E., Dierckx, H., Fenton, F. H., Mirabella, L., Panfilov, A. V., Sachse, F. B., Seemann, G., & Zhang, H. (2011). Models of cardiac tissue electrophysiology: Progress, challenges and open questions. *Progress in biophysics and molecular biology*, *104*(1-3), 22–48.

Clifford, G. D., Azuaje, F., McSharry, P., et al. (2006). *Advanced methods and tools for ecg data analysis* (Vol. 10). Artech house Boston.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dixon, M. F., Halperin, I., & Bilokon, P. (2020). *Machine learning in finance*. Springer.

Drezner, J. A., Fischbach, P., Froelicher, V., Marek, J., Pelliccia, A., Prutkin, J. M., Schmied, C. M., Sharma, S., Wilson, M. G., Ackerman, M. J., et al. (2013). Normal electrocardiographic findings: Recognising physiolog-

ical adaptations in athletes. *British journal of sports medicine*, *47*(3), 125–136.

Durán, J. M., & Jongsma, K. R. (2021). Who is afraid of black box algorithms? on the epistemological and ethical basis of trust in medical ai. *Journal of Medical Ethics*, *47*(5), 329–335.

Faal, M., & Almasganj, F. (2021). Obstructive sleep apnea screening from unprocessed ecg signals using statistical modelling. *Biomedical Signal Processing and Control*, *68*, 102685.

Faziludeen, S., & Sankaran, P. (2016). Ecg beat classification using evidential k-nearest neighbours. *Procedia Computer Science*, *89*, 499–505.

Ferguson, C., Inglis, S. C., Newton, P. J., Middleton, S., Macdonald, P. S., & Davidson, P. M. (2014). Atrial fibrillation: Stroke prevention in focus. *Australian Critical Care*, *27*(2), 92–98.

Gal, Y. (2016). *Uncertainty in deep learning* (Doctoral dissertation). University of Cambridge.

Gal, Y., & Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. *international conference on machine learning*, 1050–1059.

Géron, A. (2019). *Hands-on machine learning with scikit-learn, keras, and tensorflow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media.

Gertsch, M. (2008). *The ecg manual: An evidence-based approach*. Springer London. https://books.google.de/books?id=5EMsiv-vy-0C

Gertsch, M. (2004). *The ecg: A two-step approach to diagnosis*. Springer Science & Business Media.

Gomes, P. R., Soares, F. O., Correia, J., & Lima, C. (2009). Cardiac arrhythmia classification using wavelets and hidden markov models–a comparative approach. *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 4727–4730.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning* [http://www.deeplearningbook.org]. MIT Press.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, *63*(11), 139–144.

Hannun, A. Y., Rajpurkar, P., Haghpanahi, M., Tison, G. H., Bourn, C., Turakhia, M. P., & Ng, A. Y. (2019). Cardiologist-level arrhythmia detec-

tion and classification in ambulatory electrocardiograms using a deep neural network. *Nature medicine*, *25*(1), 65–69.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

He, R., Liu, Y., Wang, K., Zhao, N., Yuan, Y., Li, Q., & Zhang, H. (2019). Automatic cardiac arrhythmia classification using combination of deep residual network and bidirectional lstm. *IEEE Access*, *7*, 102119–102135.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735–1780.

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural networks*, *2*(5), 359–366.

Hüllermeier, E., & Waegeman, W. (2021). Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning*, *110*(3), 457–506.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, *abs/1502.03167*. http://arxiv.org/abs/1502.03167

Jaynes, E. T. (2003). *Probability theory: The logic of science*. Cambridge university press.

Joy, T. T., Rana, S., Gupta, S., & Venkatesh, S. (2016). Hyperparameter tuning for big data using bayesian optimisation. *2016 23rd International Conference on Pattern Recognition (ICPR)*, 2574–2579.

Kara, V. (2021). *In silico investigations into the use of electrocardiographic imaging to detect regions of ventricular ischemia* (Doctoral dissertation). University of Manchester.

Kawaji, T., Ogawa, H., Hamatani, Y., Kato, M., Yokomatsu, T., Miki, S., Abe, M., & Akao, M. (2021). Association of inverted t wave during atrial fibrillation rhythm with subsequent cardiac events. *Heart*.

Kher, R. (2019). Signal processing techniques for removing noise from ecg signals. *J. Biomed. Eng. Res*, *3*(101), 1–9.

Khosravi, A., Nahavandi, S., Creighton, D., & Atiya, A. F. (2011). Comprehensive review of neural network-based prediction intervals and new advances. *IEEE Transactions on neural networks*, *22*(9), 1341–1356.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kingma, D. P., Welling, M. et al. (2019). An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, *12*(4), 307–392.

Kusumoto, F. (2020). *Ecg interpretation: From pathophysiology to clinical application*. Springer Nature.

Kusumoto, F., & Bernath, P. (2011). *Ecg interpretation for everyone: An on-the-spot guide*. John Wiley & Sons.

Lakshminarayanan, B., Pritzel, A., & Blundell, C. (2016). Simple and scalable predictive uncertainty estimation using deep ensembles. *arXiv preprint arXiv:1612.01474*.

LeCun, Y. et al. (1989). Generalization and network design strategies. *Connectionism in perspective*, *19*, 143–155.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436–444.

LeCun, Y. A., Bottou, L., Orr, G. B., & Müller, K.-R. (2012). Efficient backprop. *Neural networks: Tricks of the trade* (pp. 9–48). Springer.

Ledos, J., Des Francs, X. C., Strauss, A., Le Beux, P., Auvert, B., & Fontaine, D. (1988). An expert system for interpretation and diagnosis of ecg signals. *Expert systems and decision support in medicine* (pp. 191–195). Springer.

Li, C., Zheng, C., & Tai, C. (1995). Detection of ecg characteristic points using wavelet transforms. *IEEE Transactions on biomedical Engineering*, *42*(1), 21–28.

Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., & Talwalkar, A. (2017). Hyperband: A novel bandit-based approach to hyperparameter optimization. *The Journal of Machine Learning Research*, *18*(1), 6765–6816.

Lian, J., Wang, L., & Muessig, D. (2011). A simple method to detect atrial fibrillation using rr intervals. *The American journal of cardiology*, *107*(10), 1494–1497.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Luong, M.-T., Pham, H., & Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Maji, C., Sengupta, P., Batabyal, A., & Chaudhuri, H. (2020). Nonlinear and statistical analysis of ecg signals from arrhythmia affected cardiac system through the emd process. *arXiv preprint arXiv:2002.03840*.

Mukhoti, J., Dokania, P. K., Torr, P. H., & Gal, Y. (2020). On batch normalisation for approximate bayesian inference. *arXiv preprint arXiv:2012.13220*.

Murphy, K. P. (2022). *Probabilistic machine learning: An introduction*. MIT press.

Natarajan, A., Chang, Y., Mariani, S., Rahman, A., Boverman, G., Vij, S., & Rubin, J. (2020). A wide and deep transformer neural network for 12-lead ecg classification. *2020 Computing in Cardiology*, 1–4.

Newcombe, R. G. (1998). Two-sided confidence intervals for the single proportion: Comparison of seven methods. *Statistics in medicine*, *17*(8), 857–872.

Pandey, S. K., Janghel, R. R., & Vani, V. (2020). Patient specific machine learning models for ecg signal classification. *Procedia Computer Science*, *167*, 2181–2190.

Parvaneh, S., Rubin, J., Babaeizadeh, S., & Xu-Wilson, M. (2019). Cardiac arrhythmia detection using deep learning: A review. *Journal of electrocardiology*, *57*, S70–S74.

Patel, V. L., Arocha, J. F., & Kaufman, D. R. (1999). Expertise and tacit knowledge in medicine. *Tacit knowledge in professional practice* (pp. 89–114). Psychology Press.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, *1*(8), 9.

Rawshani, A. (2021). *Clinical ecg interpretation*.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). Model-agnostic interpretability of machine learning. *arXiv preprint arXiv:1606.05386*.

Rothman, D. (2021). *Transformers for natural language processing: Build innovative deep neural network architectures for nlp with python, pytorch, tensorflow, bert, roberta, and more*. Packt Publishing Ltd.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, *323*(6088), 533–536.

Salyer, S. W. (2007). *Essential emergency medicine: For the healthcare practitioner*. Elsevier Health Sciences.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-

based localization. *Proceedings of the IEEE international conference on computer vision*, 618–626.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*(56), 1929–1958. http://jmlr.org/papers/v15/srivastava14a.html

Tortora, G. J., & Derrickson, B. H. (2018). *Principles of anatomy and physiology*. John Wiley & Sons.

Turakhia, M. P., Desai, M., Hedlin, H., Rajmane, A., Talati, N., Ferris, T., Desai, S., Nag, D., Patel, M., Kowey, P., et al. (2019). Rationale and design of a large-scale, app-based study to identify cardiac arrhythmias using a smartwatch: The apple heart study. *American heart journal*, *207*, 66–75.

Valle-Perez, G., Camargo, C. Q., & Louis, A. A. (2018). Deep learning generalizes because the parameter-function map is biased towards simple functions. *arXiv preprint arXiv:1805.08522*.

Vanwinckelen, G., & Blockeel, H. (2012). On estimating model accuracy with repeated cross-validation. *BeneLearn 2012: Proceedings of the 21st Belgian-Dutch conference on machine learning*, 39–44.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Vuksanovic, B., & Alhamdi, M. (2013). Ar-based method for ecg classification and patient recognition. *International Journal of Biometrics and Bioinformatics (IJBB)*, *7*(2), 74.

Wagner, P., Strodthoff, N., Bousseljot, R.-D., Kreiseler, D., Lunze, F. I., Samek, W., & Schaeffter, T. (2020). Ptb-xl, a large publicly available electrocardiography dataset. *Scientific Data*, *7*(1), 1–15.

Waks, J. W., & Josephson, M. E. (2014). Mechanisms of atrial fibrillation–reentry, rotors and reality. *Arrhythmia & electrophysiology review*, *3*(2), 90.

Wallis, S. (2013). Binomial confidence intervals and contingency tests: Mathematical fundamentals and the evaluation of alternative methods. *Journal of Quantitative Linguistics*, *20*(3), 178–208.

Whittaker, D. G. (2018). *Pathophysiology and pharmacology of short qt syndrome gene mutations in the human atria: Insights from multi-scale*

computational modelling. The University of Manchester (United Kingdom).

Wolpert, D. H. (1996). The lack of a priori distinctions between learning algorithms. *Neural computation*, *8*(7), 1341–1390.

Xi, Z., & Panoutsos, G. (2018). Interpretable machine learning: Convolutional neural networks with rbf fuzzy logic classification rules. *2018 International Conference on Intelligent Systems (IS)*, 448–454.

Xing, D., Rozenblit, J. W., Bernau, S., & Ott, P. (2014). Cardiac arrhythmia visualization in a virtual heart for electrophysiology education. *Proceedings of the 2014 Summer Simulation Multiconference*, 1–6.

Yildirim, O. (2018). A novel wavelet sequence based on deep bidirectional lstm network model for ecg signal classification. *Computers in biology and medicine*, *96*, 189–202.

Yildirim, O., Talo, M., Ciaccio, E. J., San Tan, R., & Acharya, U. R. (2020). Accurate deep neural network model to detect cardiac arrhythmia on more than 10,000 individual subject ecg records. *Computer methods and programs in biomedicine*, *197*, 105740.

Zhang, H., & Li, Z. (2021). Automatic detection for multi-labelled cardiac arrhythmia based on frame blocking pre-processing and residual networks. *Frontiers in cardiovascular medicine*, *8*, 135.

Zhao, B., Lu, H., Chen, S., Liu, J., & Wu, D. (2017). Convolutional neural networks for time series classification. *Journal of Systems Engineering and Electronics*, *28*(1), 162–169.

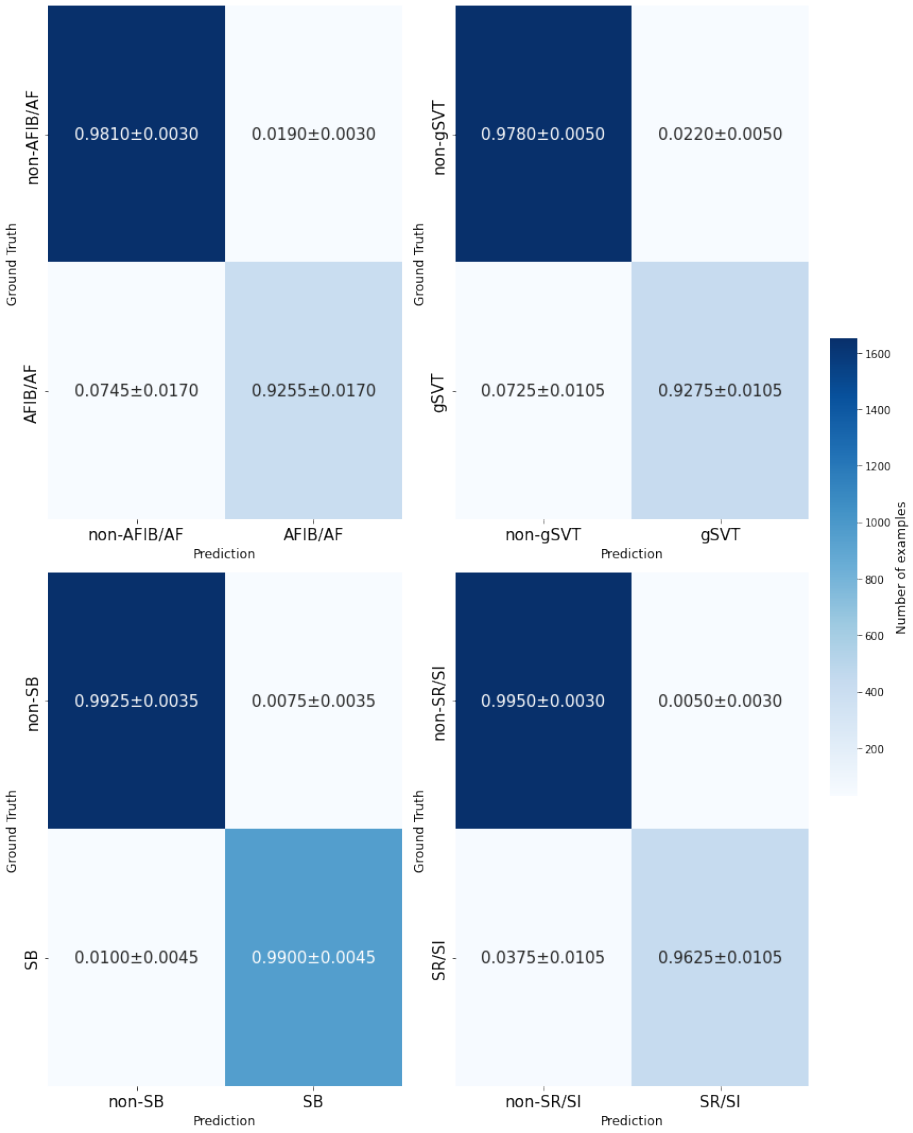
Zheng, J., Zhang, J., Danioko, S., Yao, H., Guo, H., & Rakovski, C. (2020). A 12-lead electrocardiogram database for arrhythmia research covering more than 10,000 patients. *Scientific data*, *7*(1), 1–8.

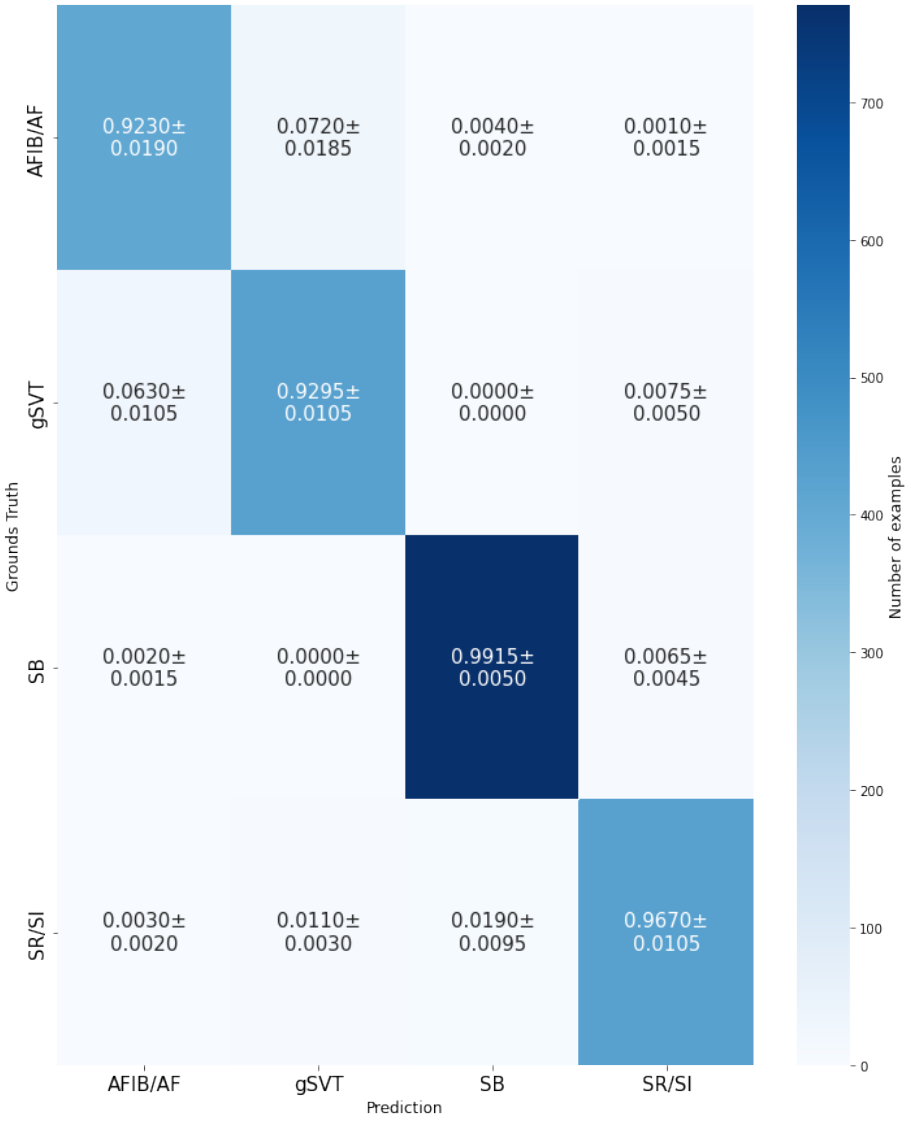
Appendices

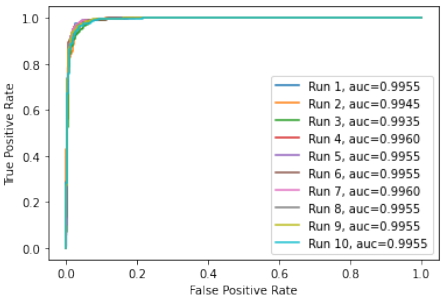
Appendix A

Confusion Matrices and ROC Curves

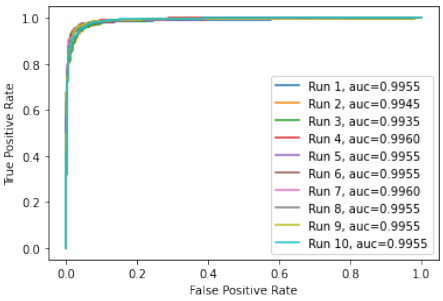
A.1 Experiment 1



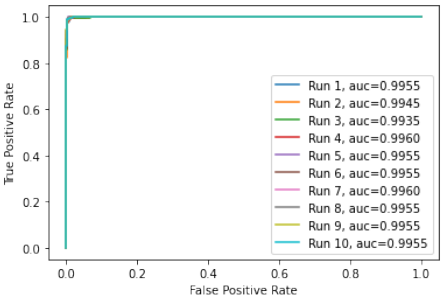

AFIB/AF
gSVT
SB
SR/SI
Grounds Truth
0.9230± 0.0190
0.0720± 0.0185
0.0040± 0.0020
0.0010± 0.0015
0.0630± 0.0105
0.9295± 0.0105
0.0000± 0.0000
0.0075± 0.0050
0.0020± 0.0015
0.0000± 0.0000
0.9915± 0.0050
0.0065± 0.0045
0.0030± 0.0020
0.0110± 0.0030
0.0190± 0.0095
0.9670± 0.0105
AFIB/AF
gSVT
SB
SR/SI
Prediction
Number of examples
700
600
500
400
300
200
100
0


True Positive Rate
False Positive Rate
Run 1, auc=0.9955
Run 2, auc=0.9945
Run 3, auc=0.9935
Run 4, auc=0.9960
Run 5, auc=0.9955
Run 6, auc=0.9955
Run 7, auc=0.9960
Run 8, auc=0.9955
Run 9, auc=0.9955
Run 10, auc=0.9955

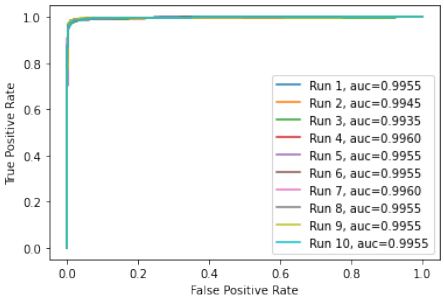
AFIB


True Positive Rate
False Positive Rate
Run 1, auc=0.9955
Run 2, auc=0.9945
Run 3, auc=0.9935
Run 4, auc=0.9960
Run 5, auc=0.9955
Run 6, auc=0.9955
Run 7, auc=0.9960
Run 8, auc=0.9955
Run 9, auc=0.9955
Run 10, auc=0.9955

SI

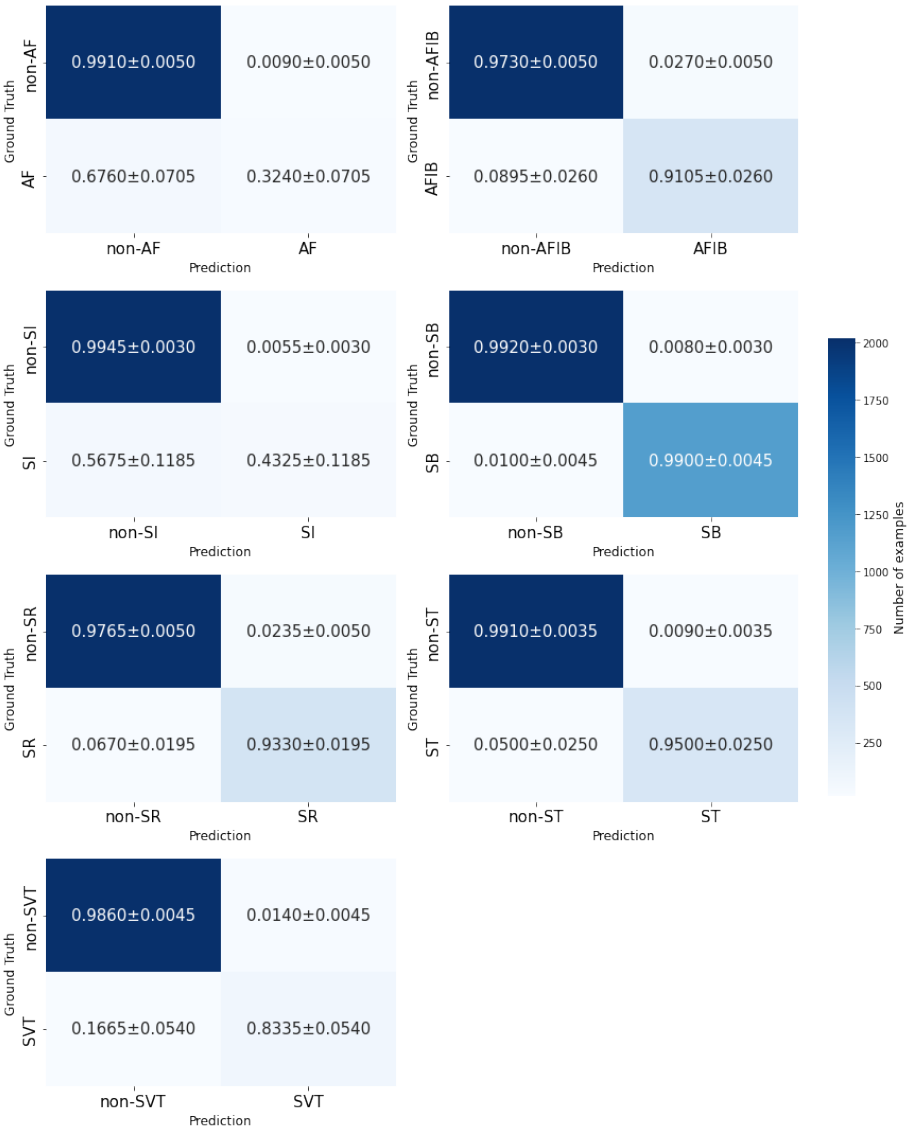

True Positive Rate
False Positive Rate
Run 1, auc=0.9955
Run 2, auc=0.9945
Run 3, auc=0.9935
Run 4, auc=0.9960
Run 5, auc=0.9955
Run 6, auc=0.9955
Run 7, auc=0.9960
Run 8, auc=0.9955
Run 9, auc=0.9955
Run 10, auc=0.9955

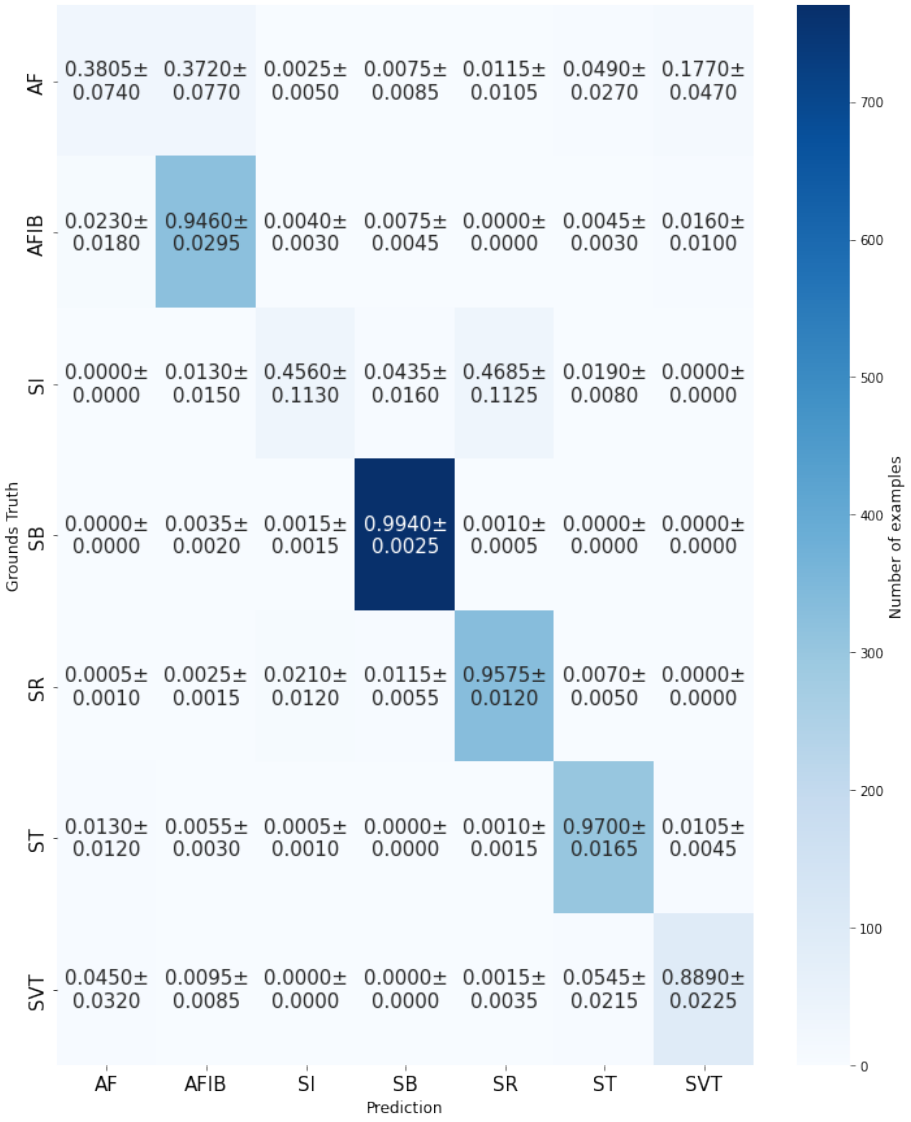
SB

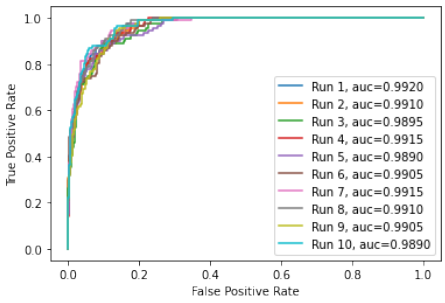

True Positive Rate
False Positive Rate
Run 1, auc=0.9955
Run 2, auc=0.9945
Run 3, auc=0.9935
Run 4, auc=0.9960
Run 5, auc=0.9955
Run 6, auc=0.9955
Run 7, auc=0.9960
Run 8, auc=0.9955
Run 9, auc=0.9955
Run 10, auc=0.9955

SR

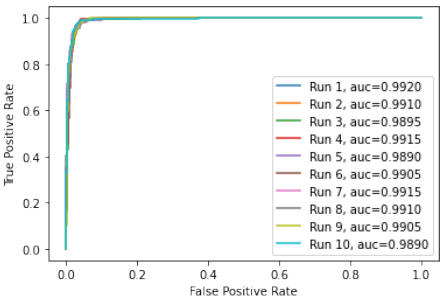
A.2 Experiment 2



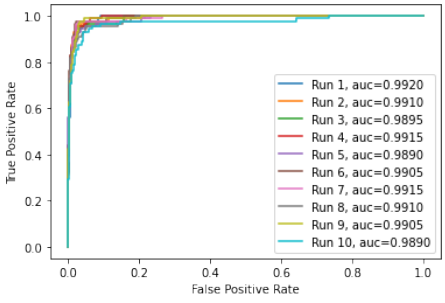

AF 0.3805± 0.0740 0.3720± 0.0770 0.0025± 0.0050 0.0075± 0.0085 0.0115± 0.0105 0.0490± 0.0270 0.1770± 0.0470
AFIB 0.0230± 0.0180 0.9460± 0.0295 0.0040± 0.0030 0.0075± 0.0045 0.0000± 0.0000 0.0045± 0.0030 0.0160± 0.0100
SI 0.0000± 0.0000 0.0130± 0.0150 0.4560± 0.1130 0.0435± 0.0160 0.4685± 0.1125 0.0190± 0.0080 0.0000± 0.0000
SB 0.0000± 0.0000 0.0035± 0.0020 0.0015± 0.0015 0.9940± 0.0025 0.0010± 0.0005 0.0000± 0.0000 0.0000± 0.0000
SR 0.0005± 0.0010 0.0025± 0.0015 0.0210± 0.0120 0.0115± 0.0055 0.9575± 0.0120 0.0070± 0.0050 0.0000± 0.0000
ST 0.0130± 0.0120 0.0055± 0.0030 0.0005± 0.0010 0.0000± 0.0000 0.0010± 0.0015 0.9700± 0.0165 0.0105± 0.0045
SVT 0.0450± 0.0320 0.0095± 0.0085 0.0000± 0.0000 0.0000± 0.0000 0.0015± 0.0035 0.0545± 0.0215 0.8890± 0.0225
Grounds Truth
AF AFIB SI SB SR ST SVT
Prediction
Number of examples
0 100 200 300 400 500 600 700



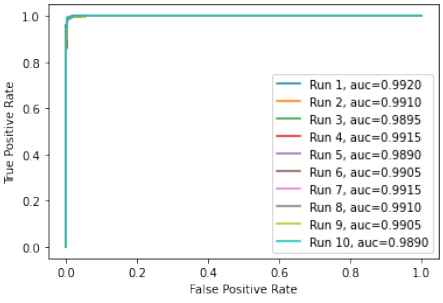
AF



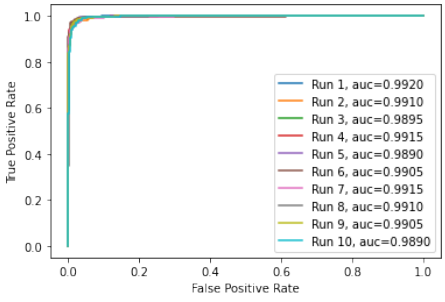
AFIB



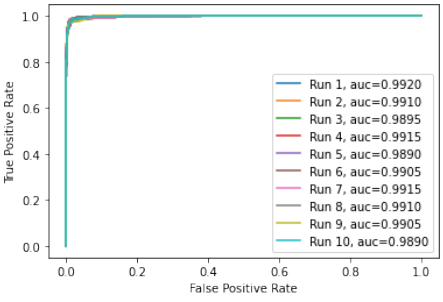
SI



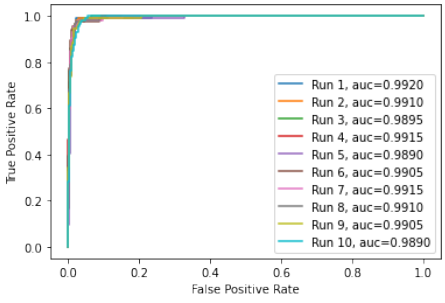
SB



SR

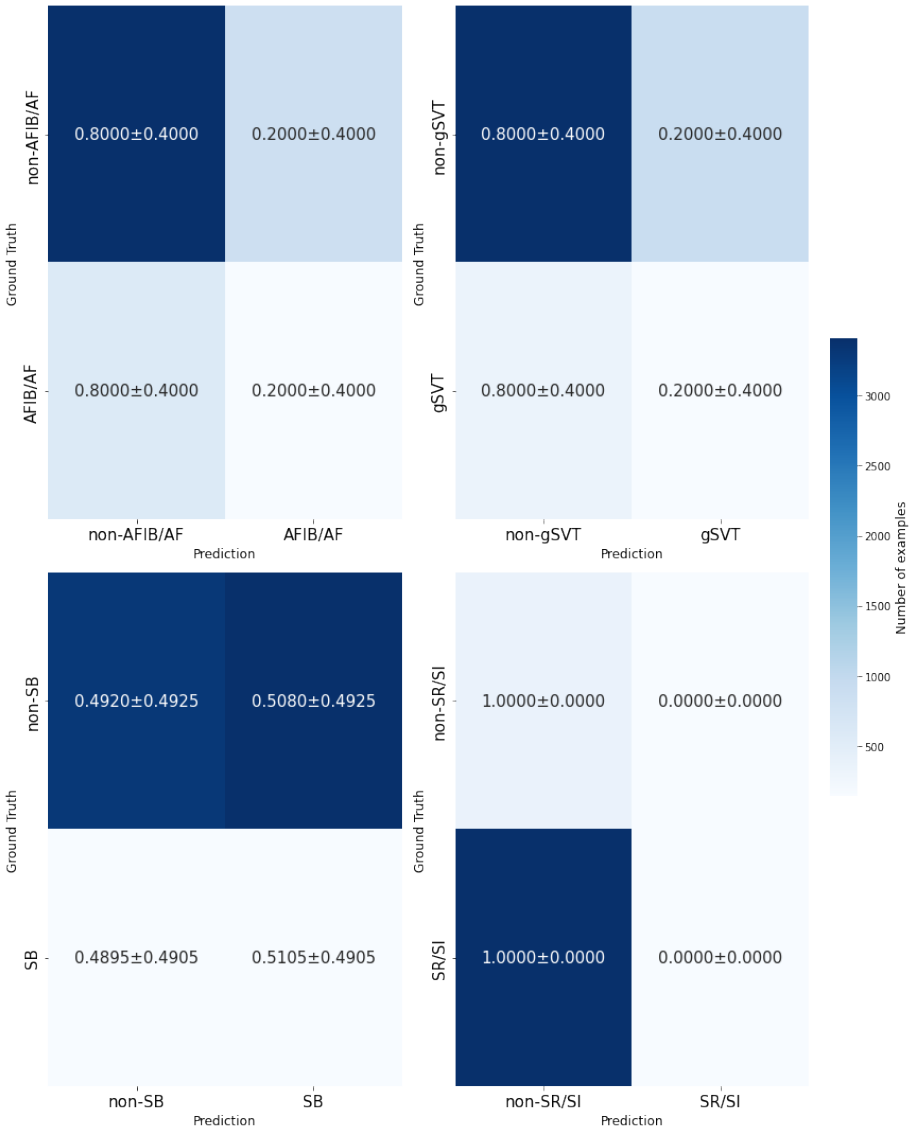


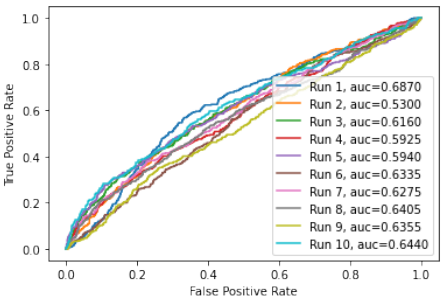
ST



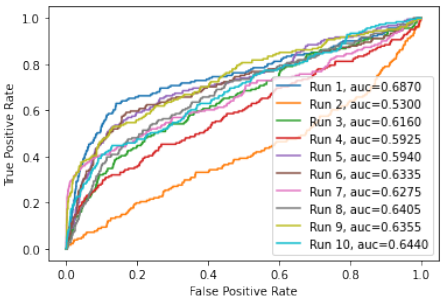
SVT

A.3 Experiment 3

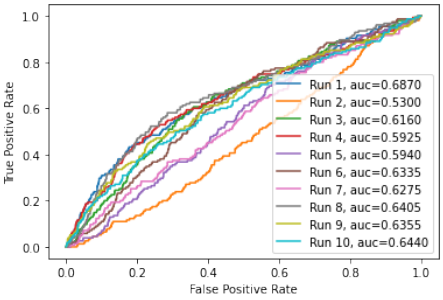




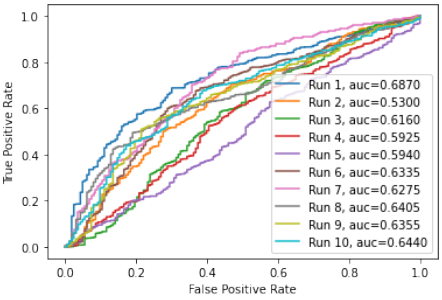
AF/AFIB



gSVT

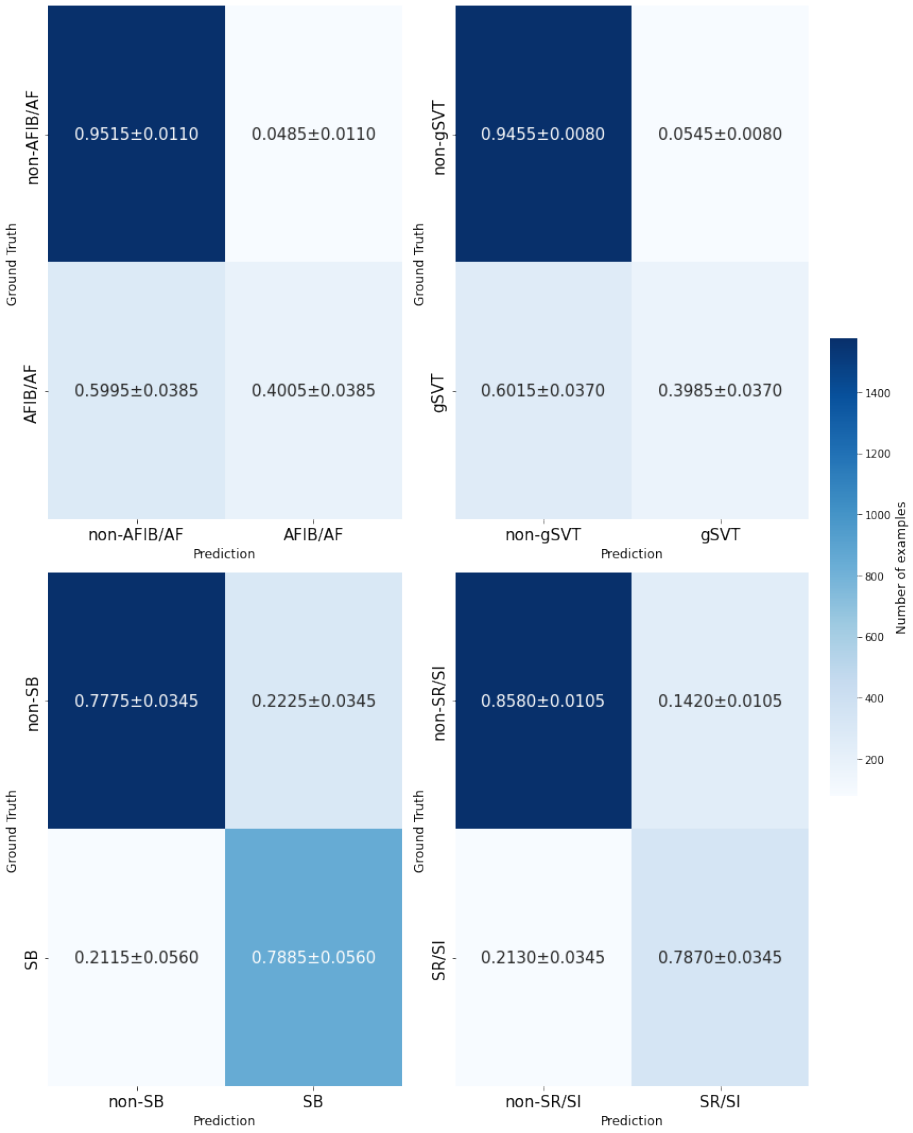


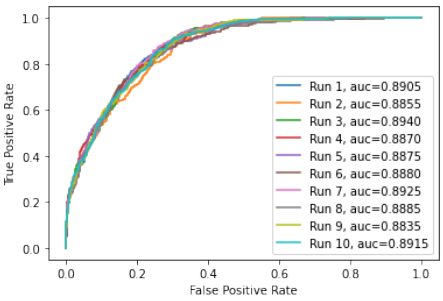
SB



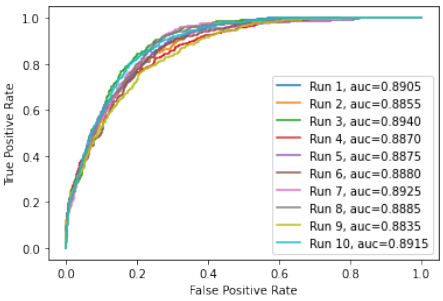
SI/SR

A.4 Experiment 4

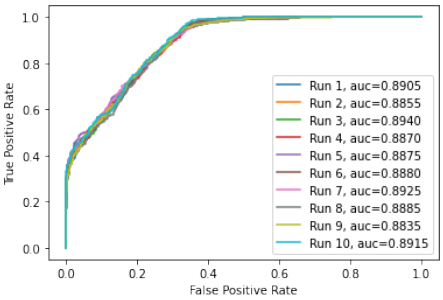




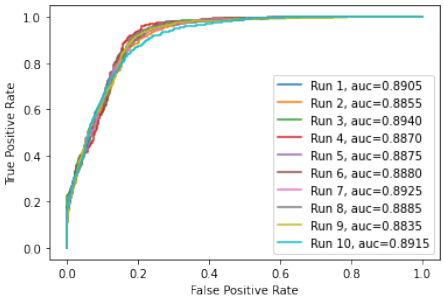
AF/AFIB



gSVT

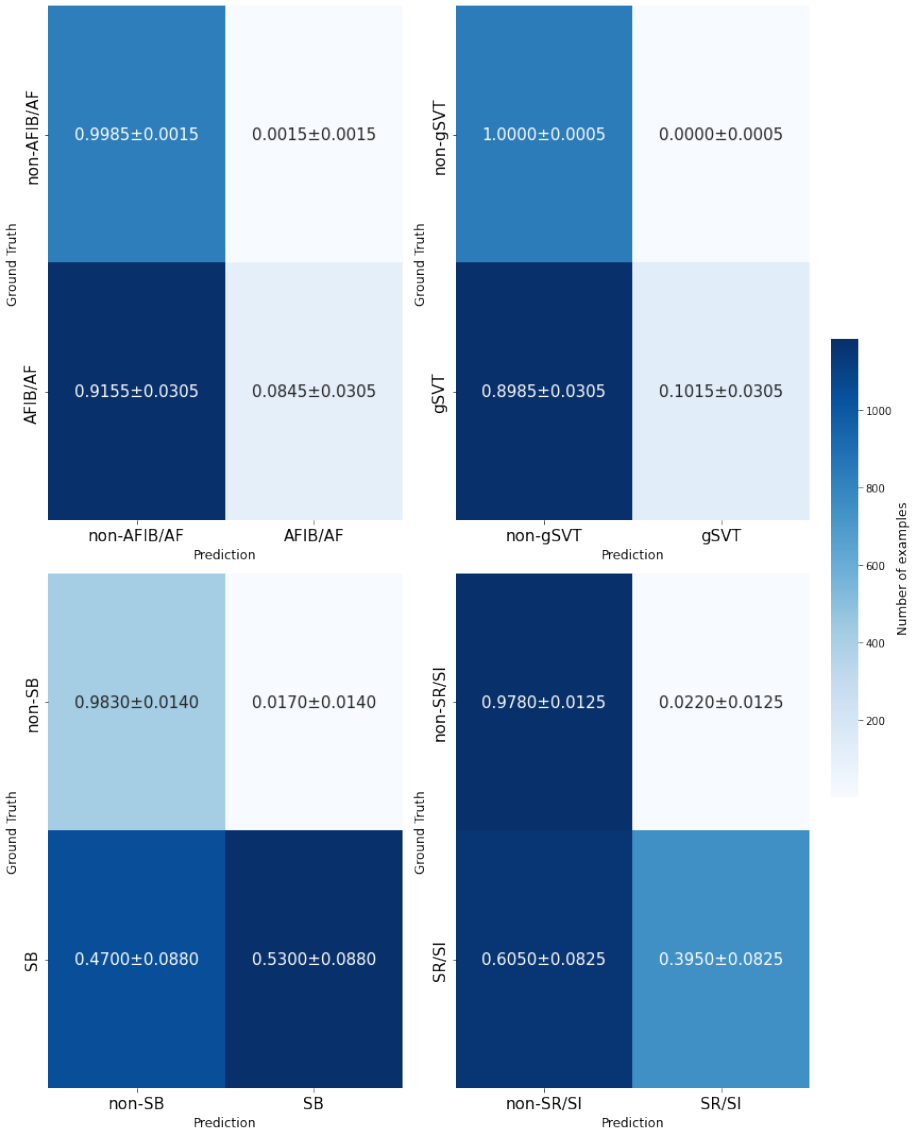


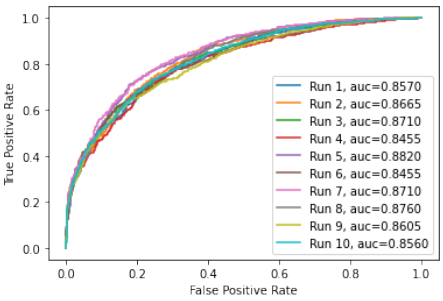
SB



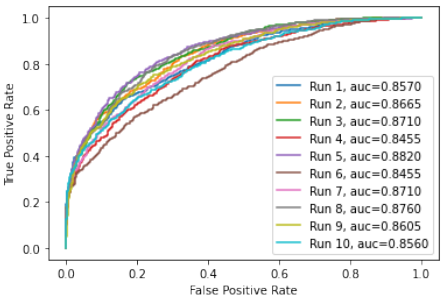
SI/SR

A.5 Experiment 5

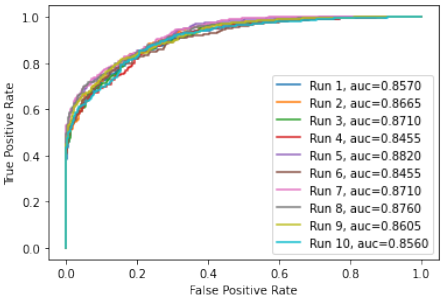




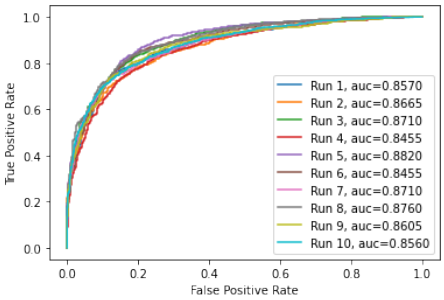
AF/AFIB



gSVT

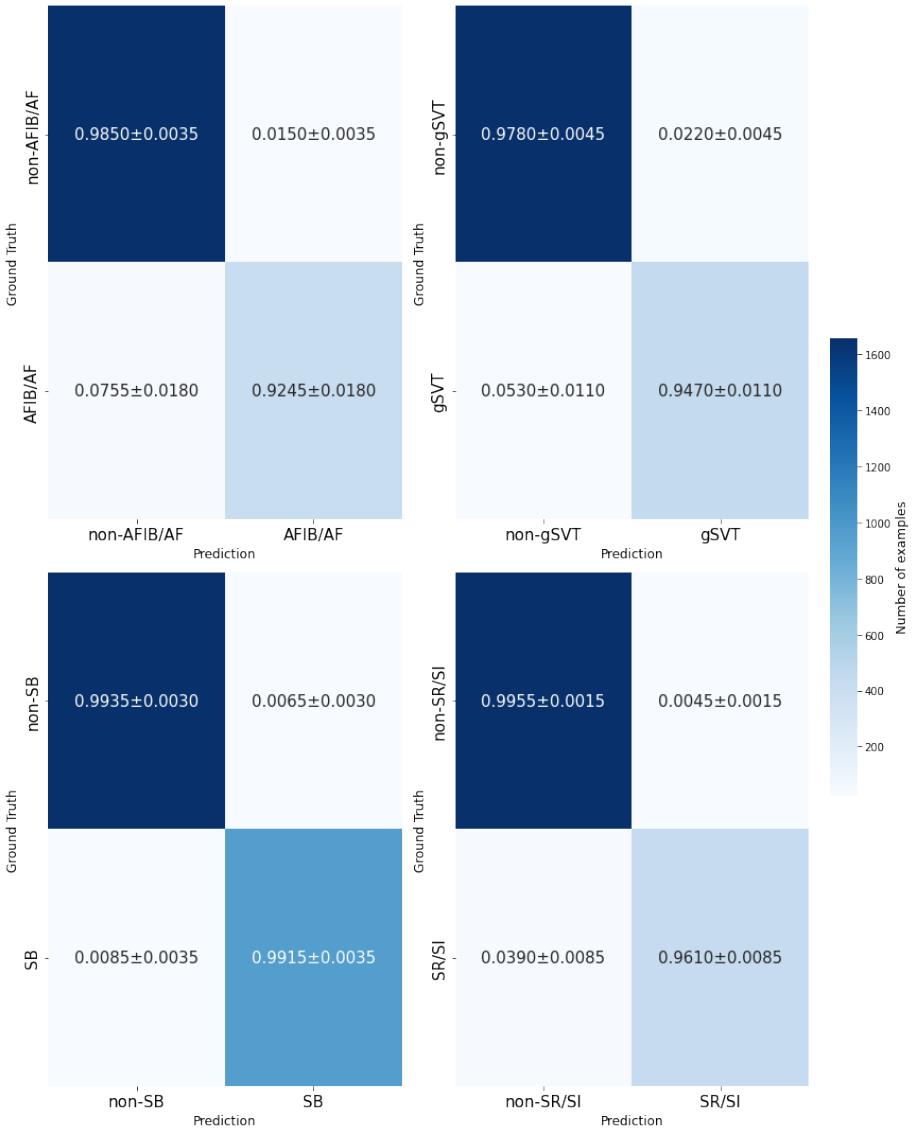


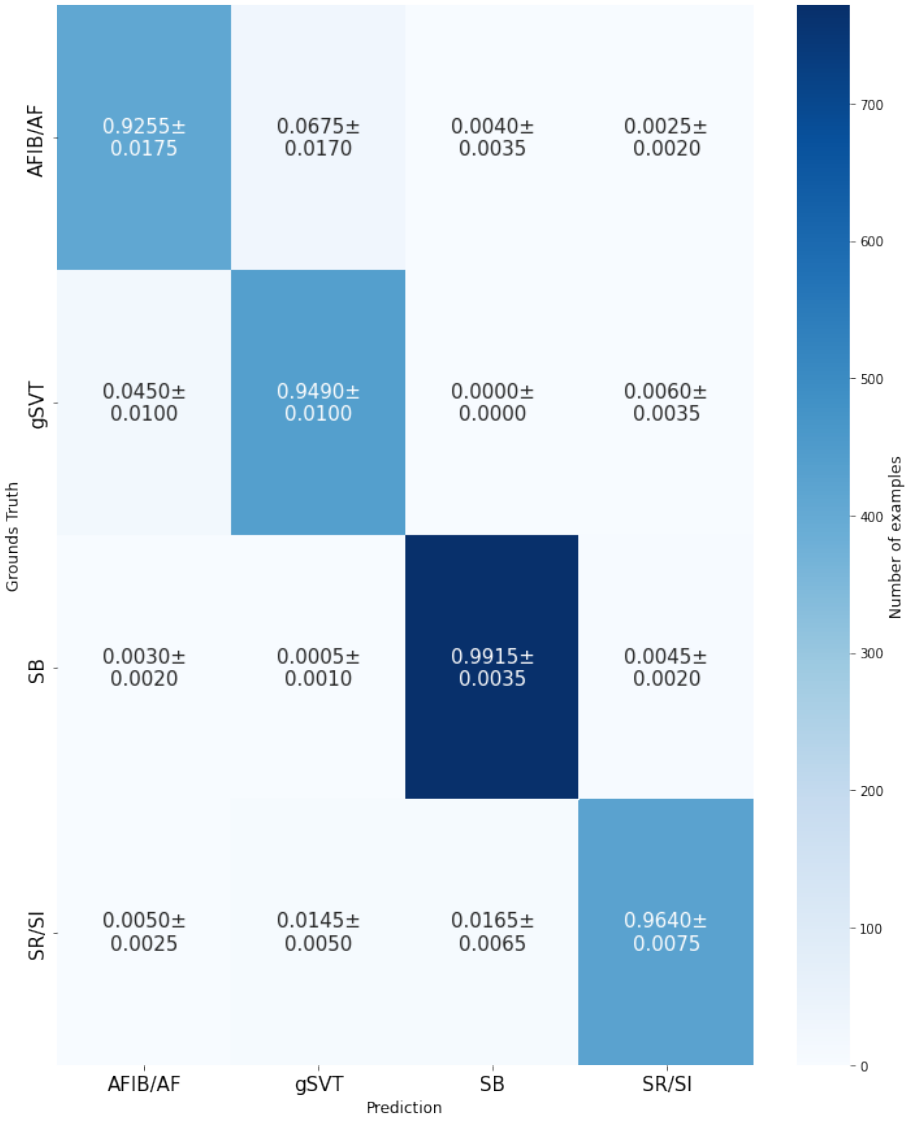
SB

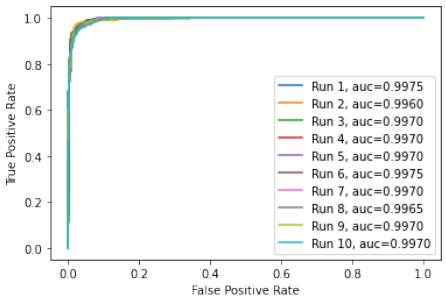


SI/SR

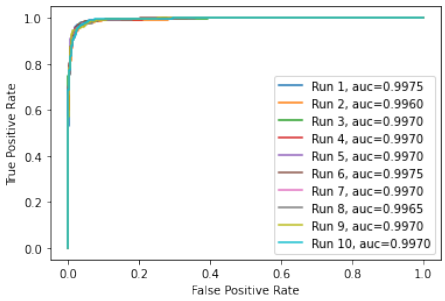
A.6 Experiment 6



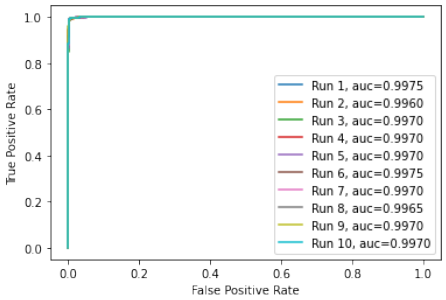

AFIB/AF
gSVT
SB
SR/SI
Grounds Truth
0.9255± 0.0175
0.0675± 0.0170
0.0040± 0.0035
0.0025± 0.0020
0.0450± 0.0100
0.9490± 0.0100
0.0000± 0.0000
0.0060± 0.0035
0.0030± 0.0020
0.0005± 0.0010
0.9915± 0.0035
0.0045± 0.0020
0.0050± 0.0025
0.0145± 0.0050
0.0165± 0.0065
0.9640± 0.0075
AFIB/AF
gSVT
SB
SR/SI
Prediction
Number of examples
700
600
500
400
300
200
100
0



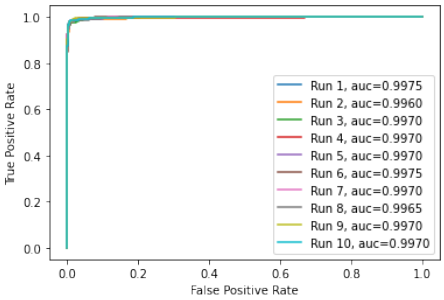
AF/AFIB



gSVT

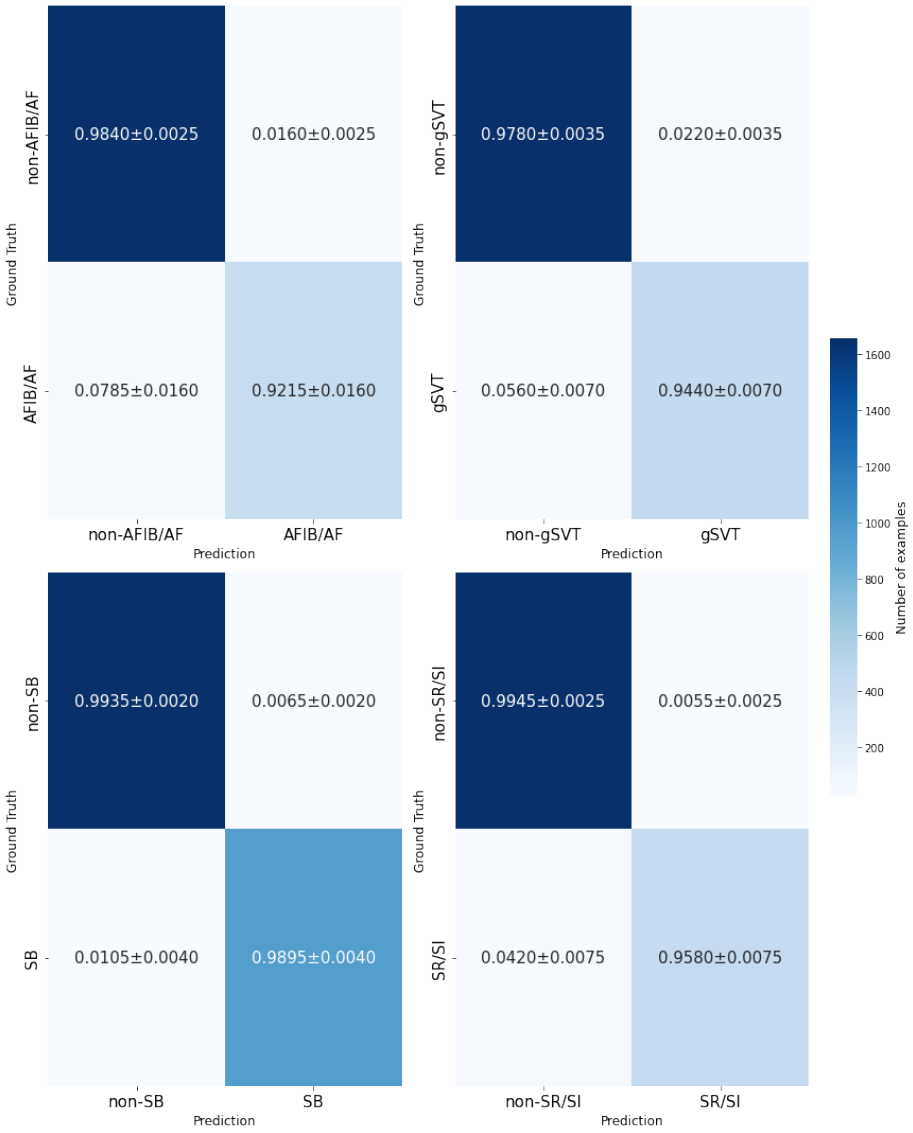


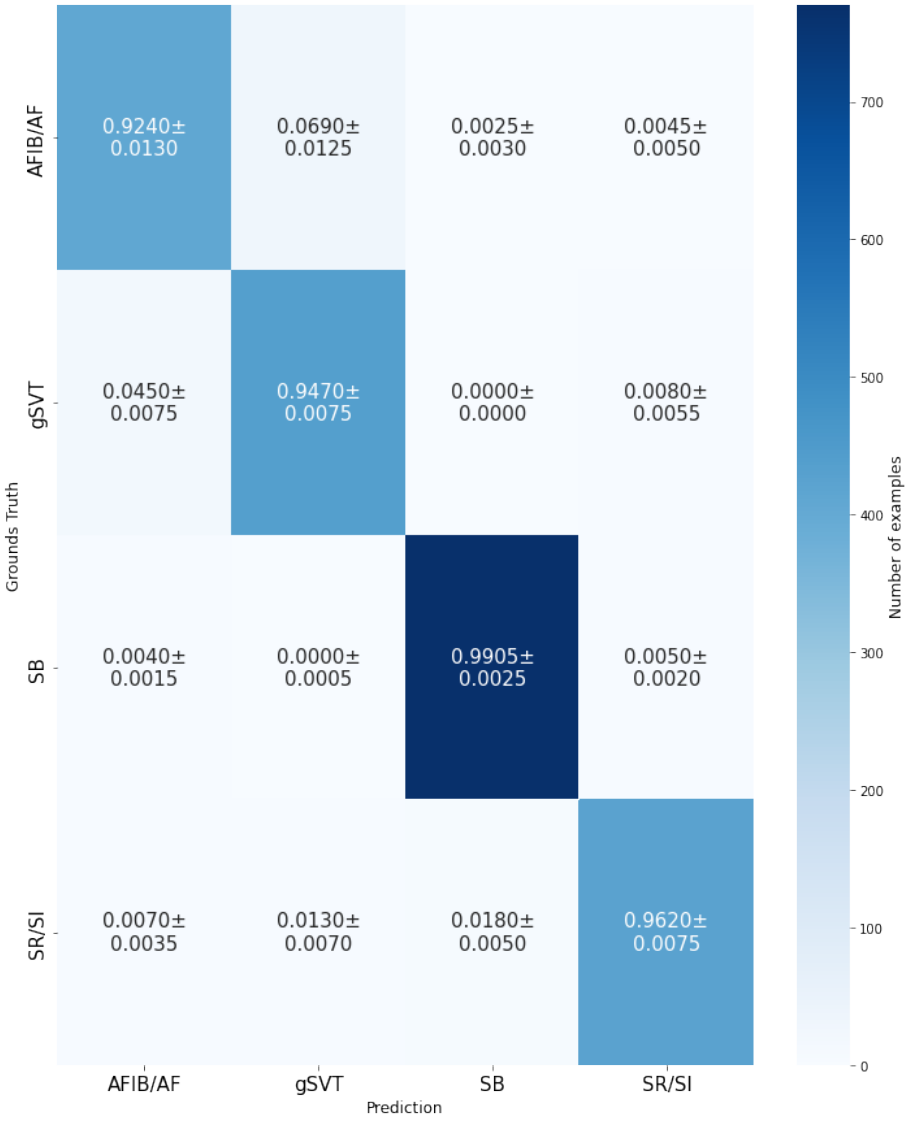
SB

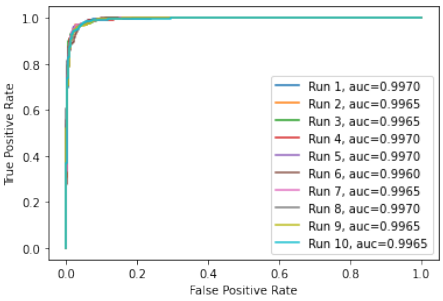


SI/SR

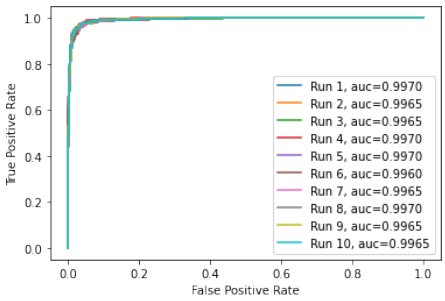
A.7 Experiment 7



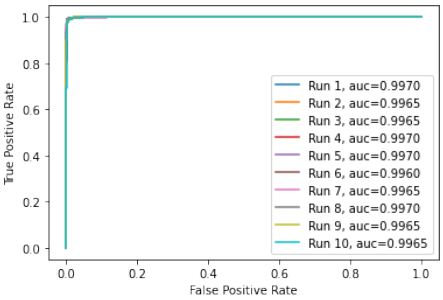

0.9240± 0.0130
0.0690± 0.0125
0.0025± 0.0030
0.0045± 0.0050
0.0450± 0.0075
0.9470± 0.0075
0.0000± 0.0000
0.0080± 0.0055
0.0040± 0.0015
0.0000± 0.0005
0.9905± 0.0025
0.0050± 0.0020
0.0070± 0.0035
0.0130± 0.0070
0.0180± 0.0050
0.9620± 0.0075
AFIB/AF
gSVT
SB
SR/SI
Grounds Truth
AFIB/AF
gSVT
SB
SR/SI
Prediction
Number of examples
0
100
200
300
400
500
600
700



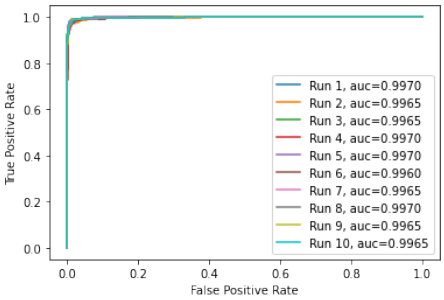
AF/AFIB



gSVT

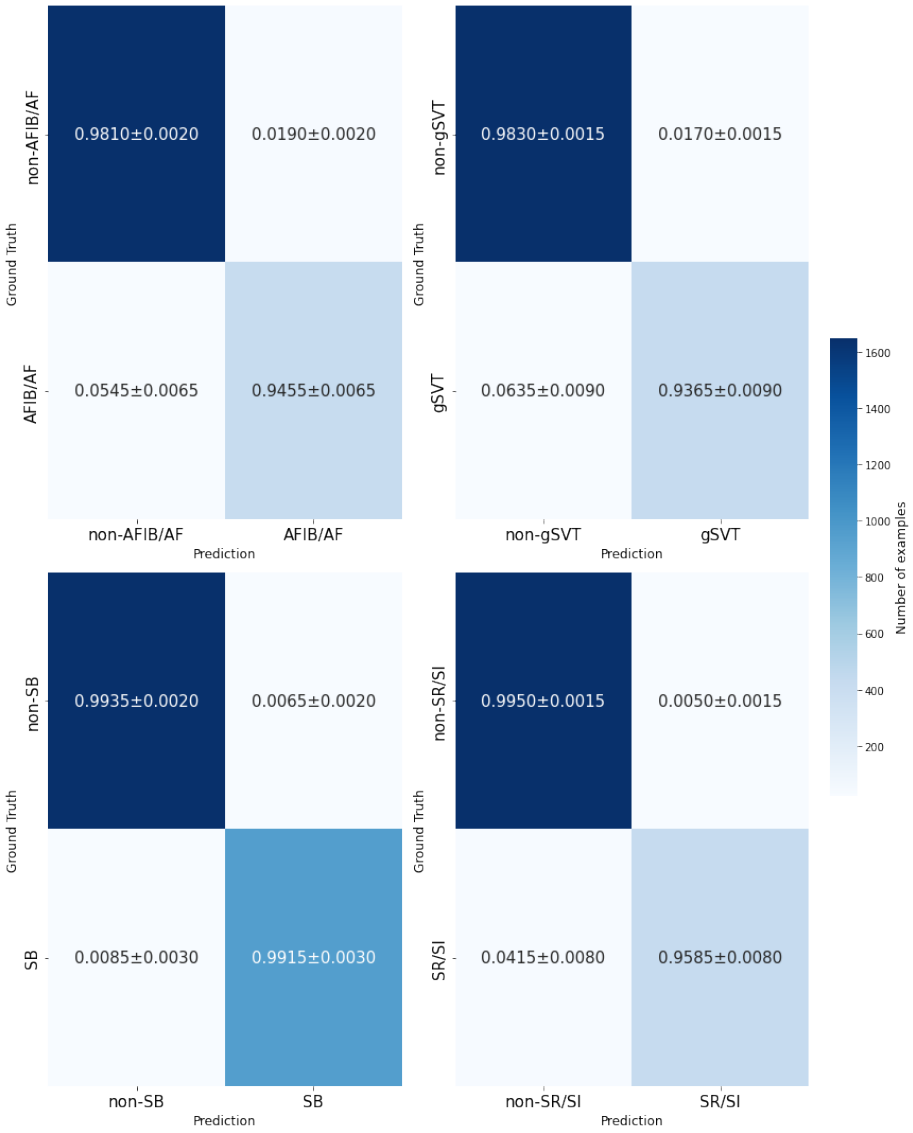


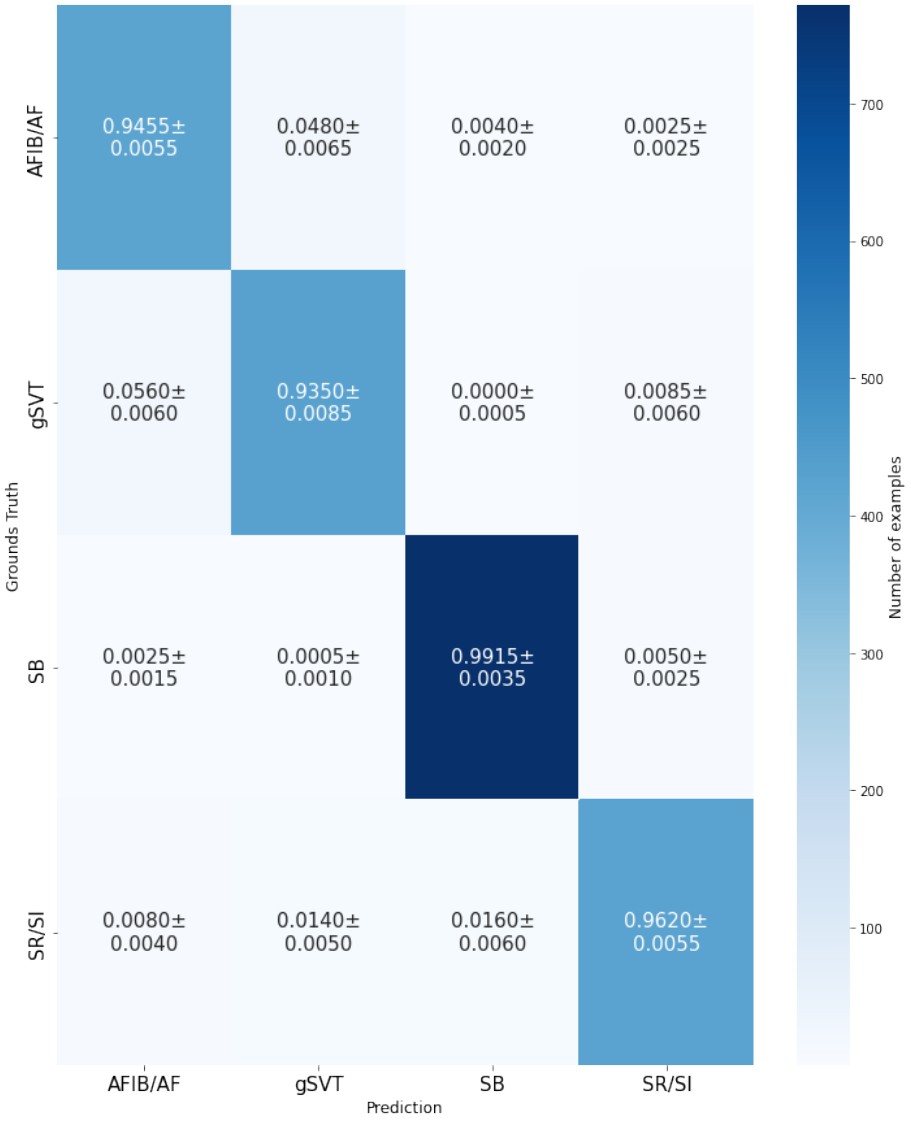
SB

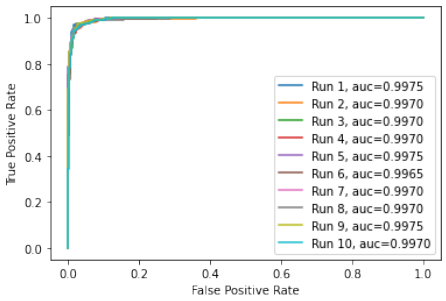


SI/SR

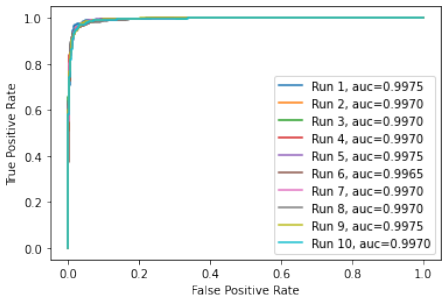
A.8 Experiment 8



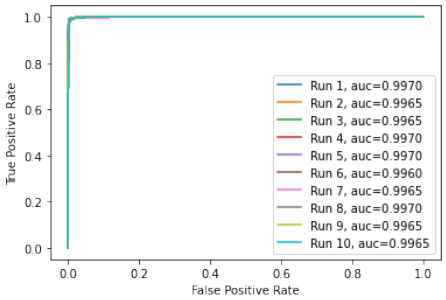

AFIB/AF
gSVT
SB
SR/SI
Grounds Truth
0.9455± 0.0055
0.0480± 0.0065
0.0040± 0.0020
0.0025± 0.0025
0.0560± 0.0060
0.9350± 0.0085
0.0000± 0.0005
0.0085± 0.0060
0.0025± 0.0015
0.0005± 0.0010
0.9915± 0.0035
0.0050± 0.0025
0.0080± 0.0040
0.0140± 0.0050
0.0160± 0.0060
0.9620± 0.0055
AFIB/AF
gSVT
SB
SR/SI
Prediction
Number of examples
700
600
500
400
300
200
100



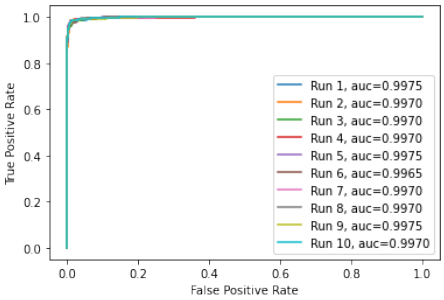
AF/AFIB



gSVT

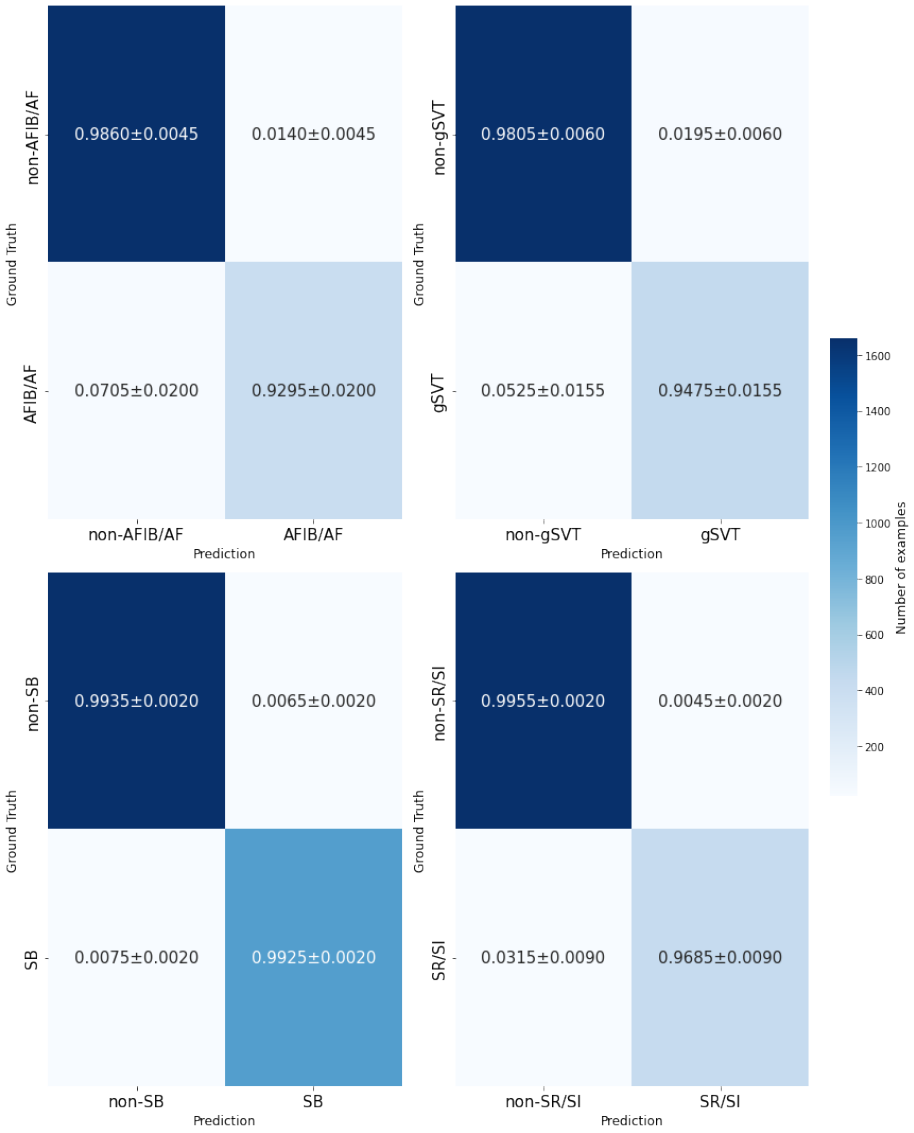


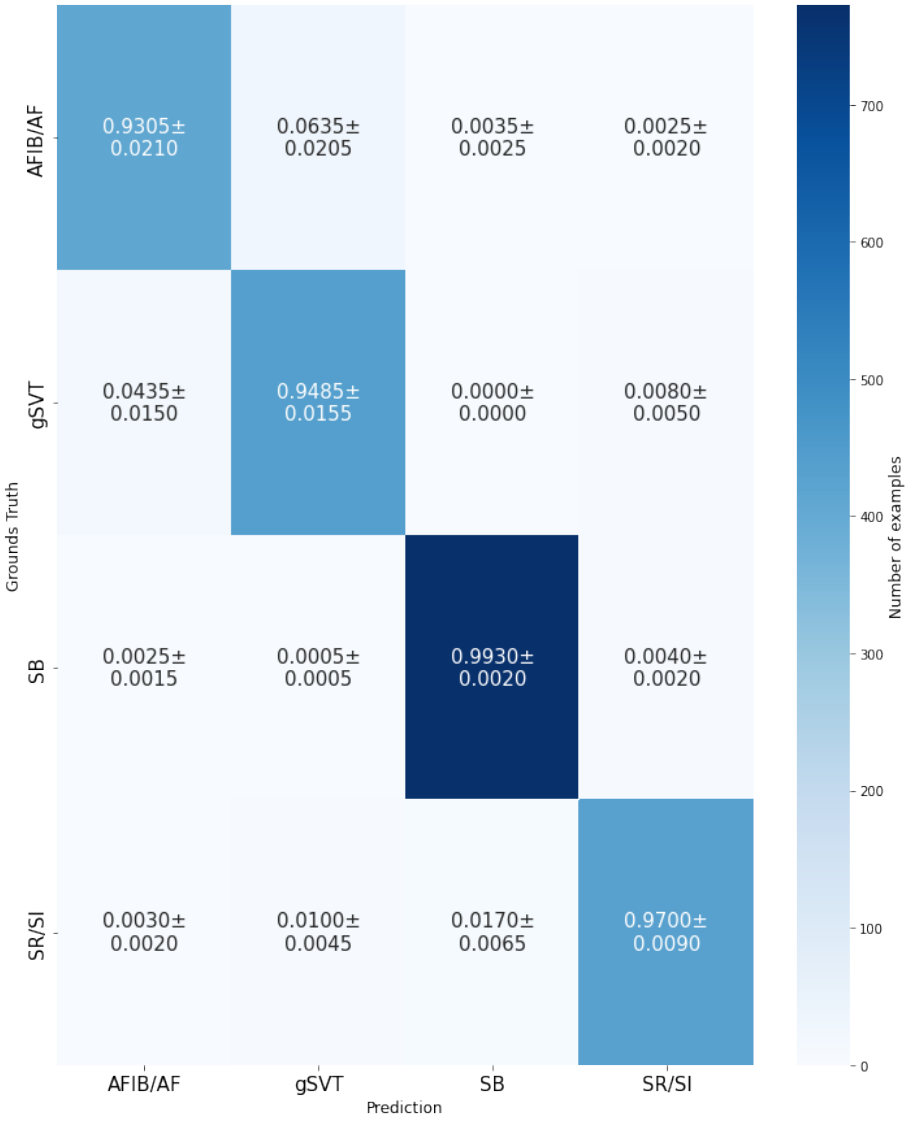
SB

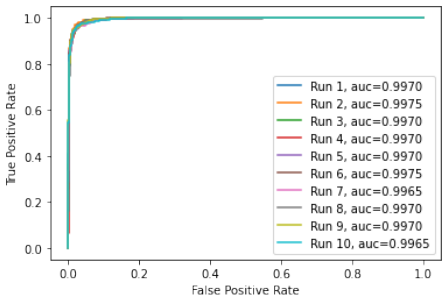


SI/SR

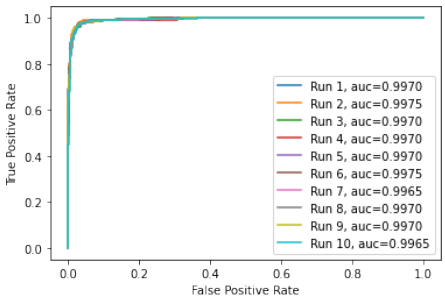
A.9 Experiment 9



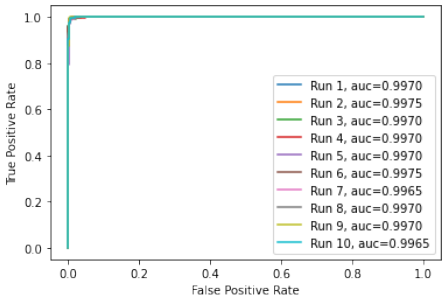

0.9305± 0.0210
0.0635± 0.0205
0.0035± 0.0025
0.0025± 0.0020
0.0435± 0.0150
0.9485± 0.0155
0.0000± 0.0000
0.0080± 0.0050
0.0025± 0.0015
0.0005± 0.0005
0.9930± 0.0020
0.0040± 0.0020
0.0030± 0.0020
0.0100± 0.0045
0.0170± 0.0065
0.9700± 0.0090
AFIB/AF
gSVT
SB
SR/SI
Grounds Truth
AFIB/AF
gSVT
SB
SR/SI
Prediction
Number of examples
0
100
200
300
400
500
600
700



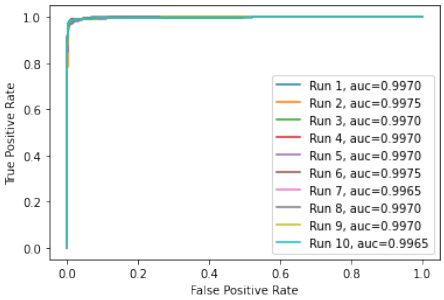
AF/AFIB



gSVT

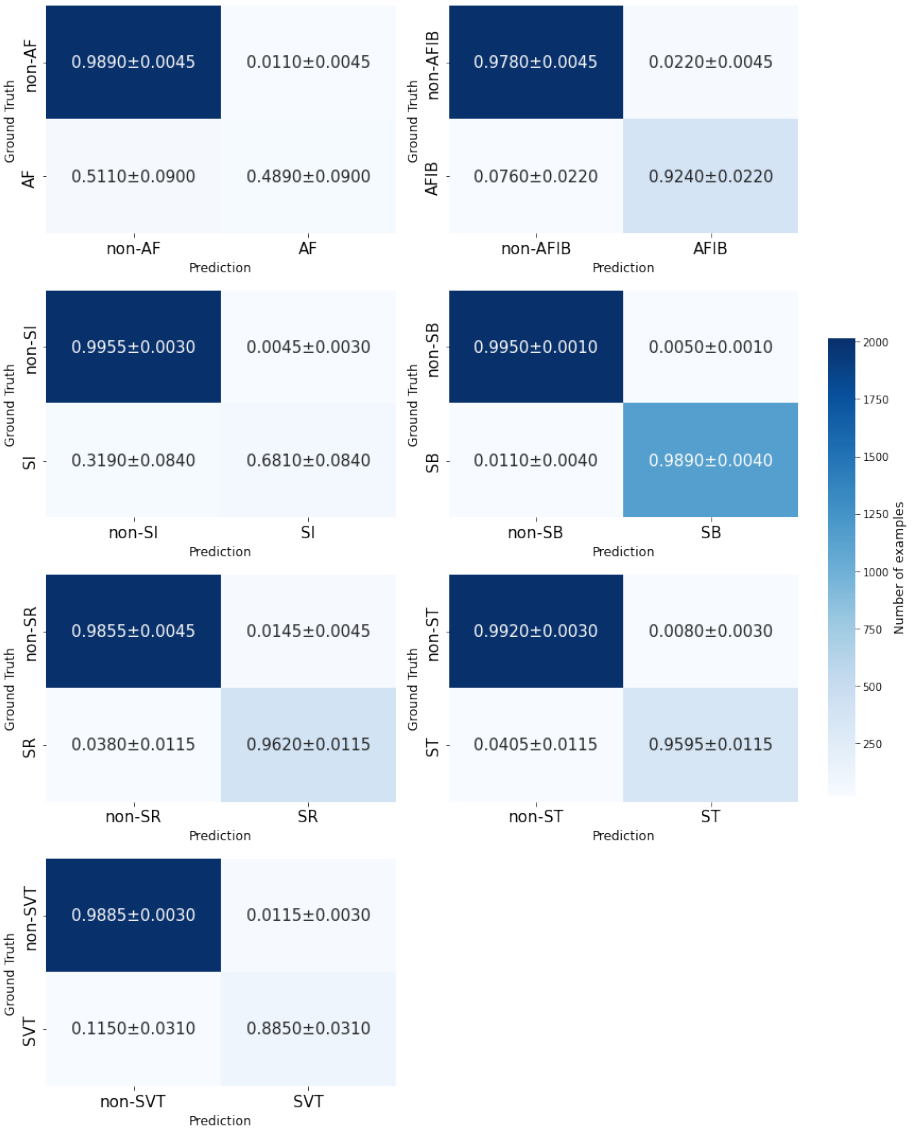


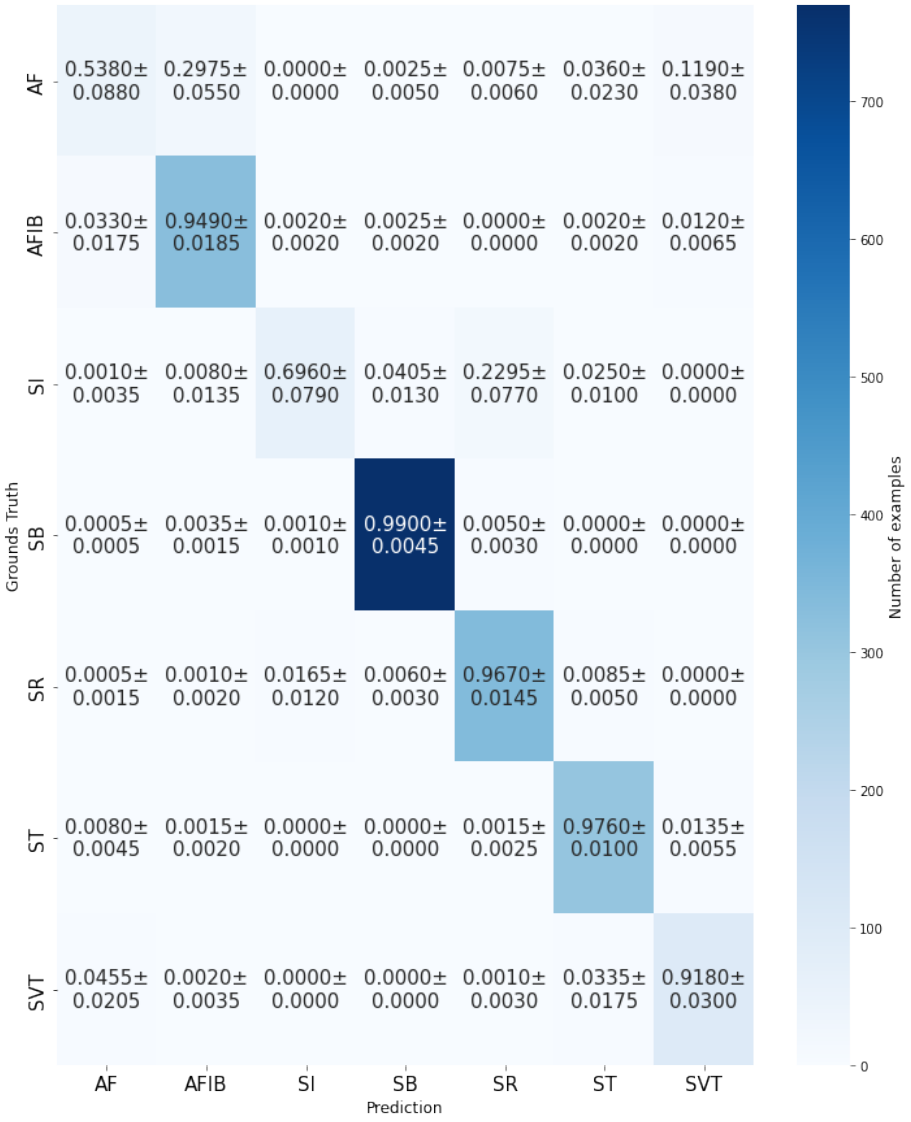
SB

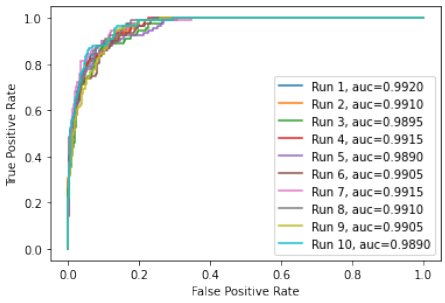


SI/SR

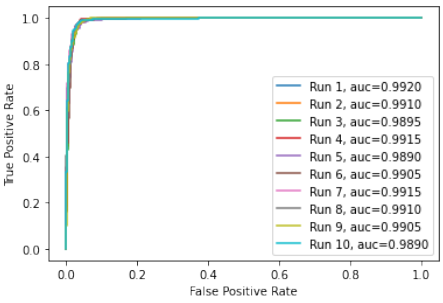
A.10 Experiment 10



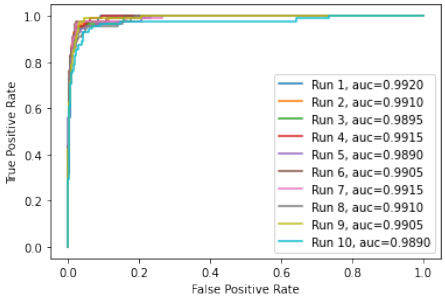

AF 0.5380± 0.0880 0.2975± 0.0550 0.0000± 0.0000 0.0025± 0.0050 0.0075± 0.0060 0.0360± 0.0230 0.1190± 0.0380
AFIB 0.0330± 0.0175 0.9490± 0.0185 0.0020± 0.0020 0.0025± 0.0020 0.0000± 0.0000 0.0020± 0.0020 0.0120± 0.0065
SI 0.0010± 0.0035 0.0080± 0.0135 0.6960± 0.0790 0.0405± 0.0130 0.2295± 0.0770 0.0250± 0.0100 0.0000± 0.0000
SB 0.0005± 0.0005 0.0035± 0.0015 0.0010± 0.0010 0.9900± 0.0045 0.0050± 0.0030 0.0000± 0.0000 0.0000± 0.0000
SR 0.0005± 0.0015 0.0010± 0.0020 0.0165± 0.0120 0.0060± 0.0030 0.9670± 0.0145 0.0085± 0.0050 0.0000± 0.0000
ST 0.0080± 0.0045 0.0015± 0.0020 0.0000± 0.0000 0.0000± 0.0000 0.0015± 0.0025 0.9760± 0.0100 0.0135± 0.0055
SVT 0.0455± 0.0205 0.0020± 0.0035 0.0000± 0.0000 0.0000± 0.0000 0.0010± 0.0030 0.0335± 0.0175 0.9180± 0.0300
Grounds Truth
AF AFIB SI SB SR ST SVT
Prediction
Number of examples
0 100 200 300 400 500 600 700



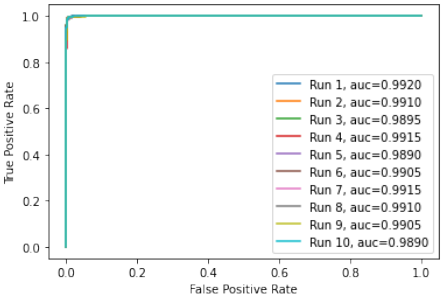
AF



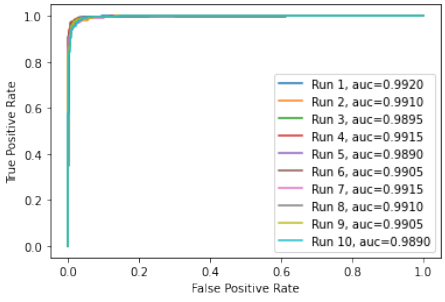
AFIB



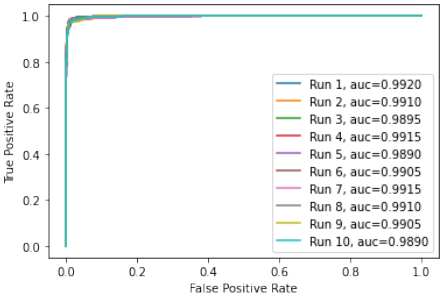
SI



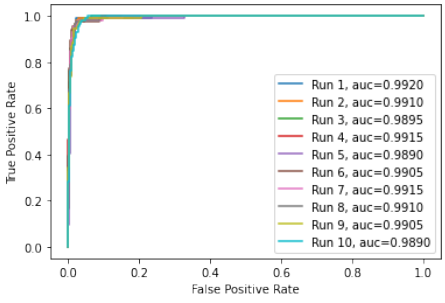
SB



SR

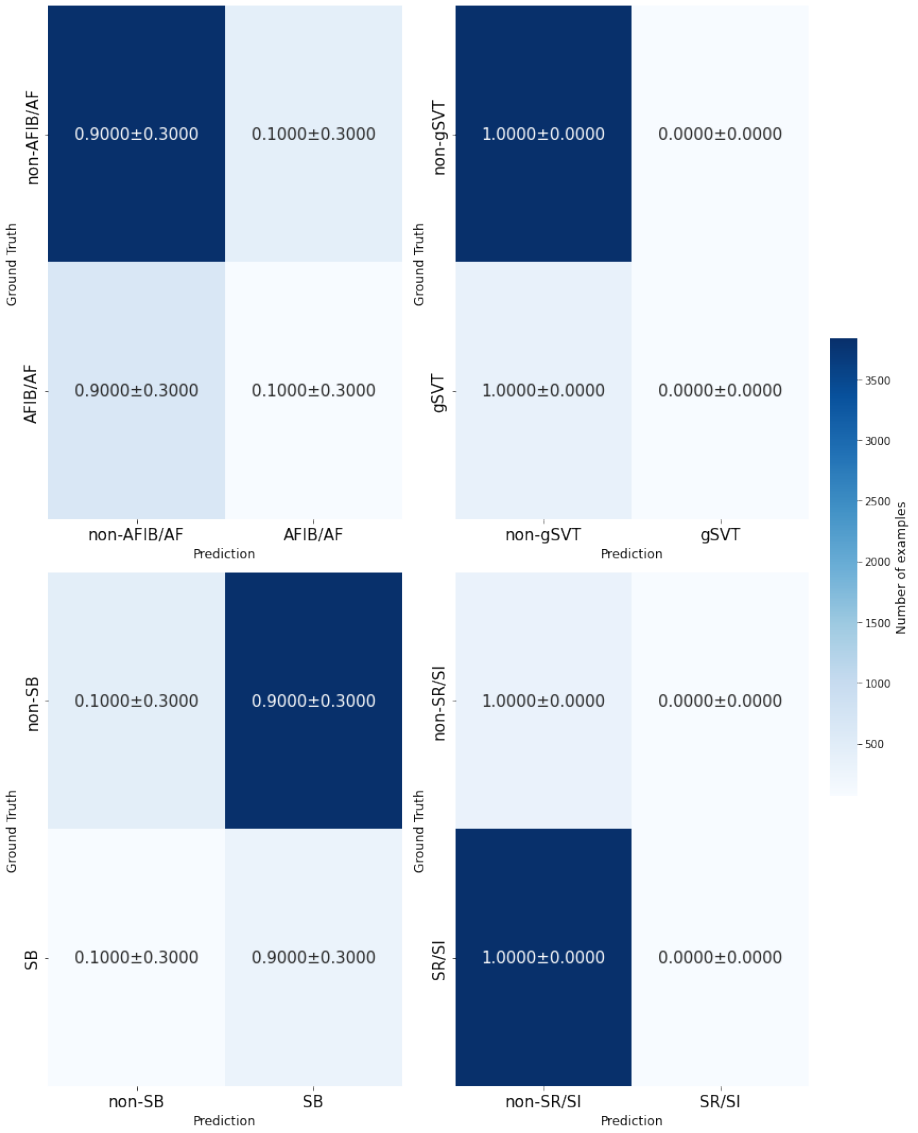


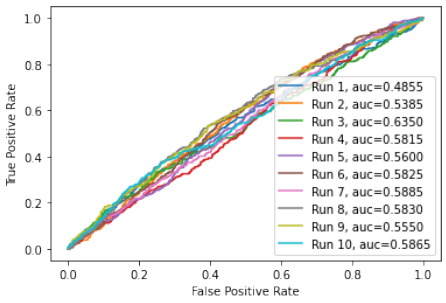
ST



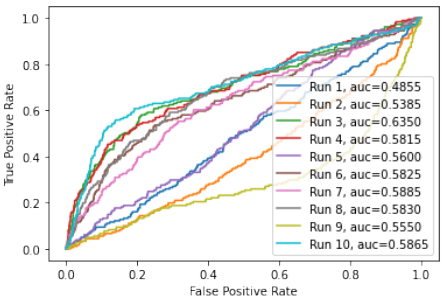
SVT

A.11 Experiment 11

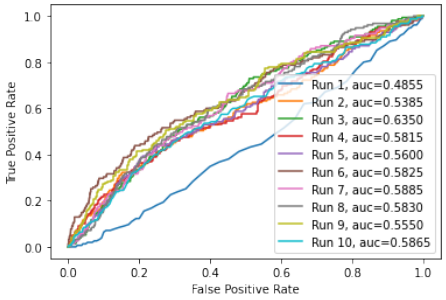




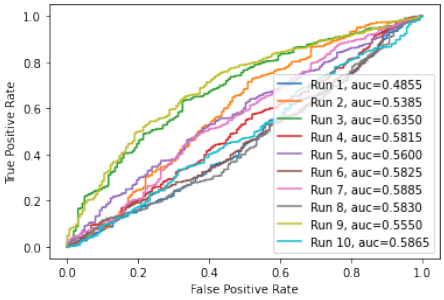
AF/AFIB



gSVT

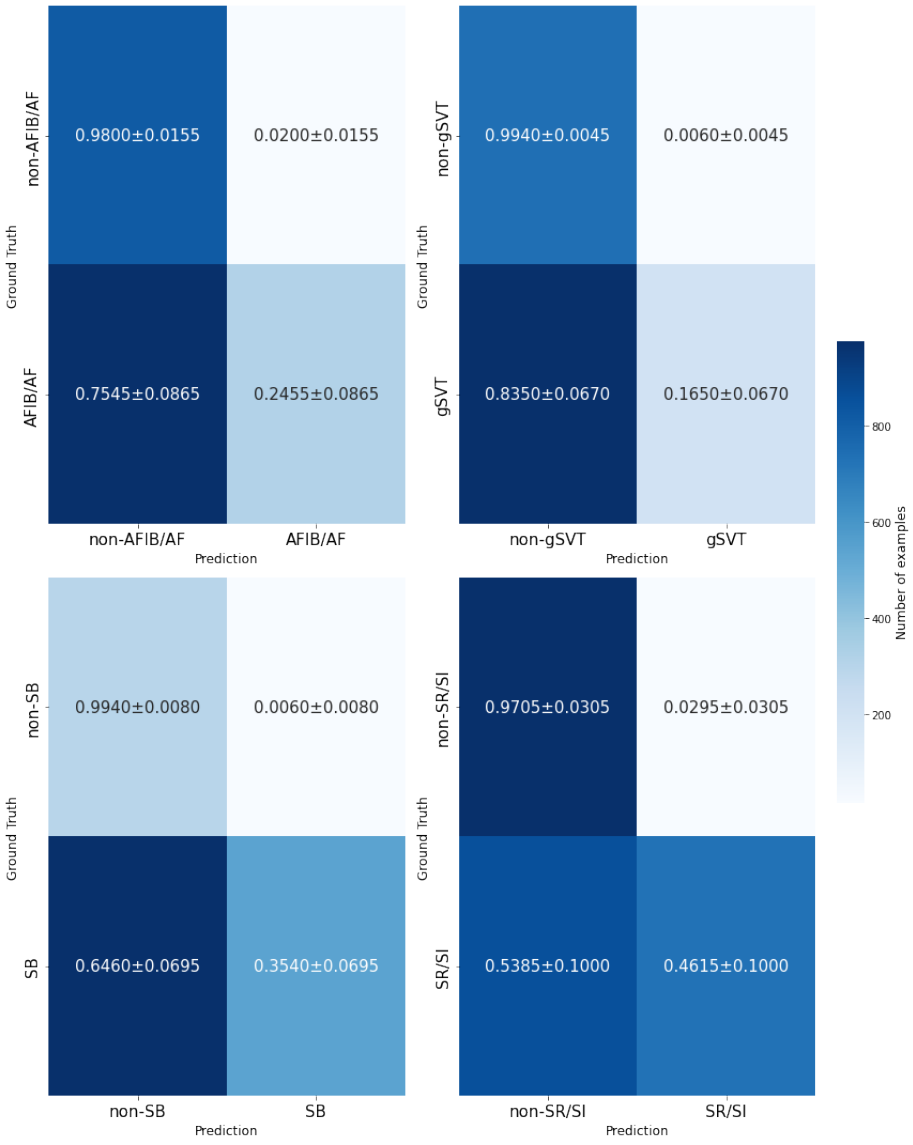


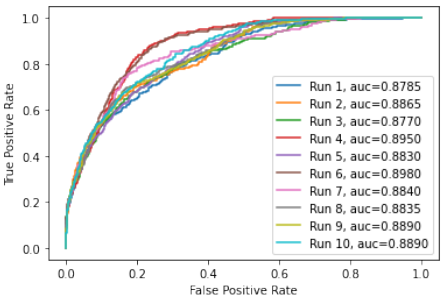
SB



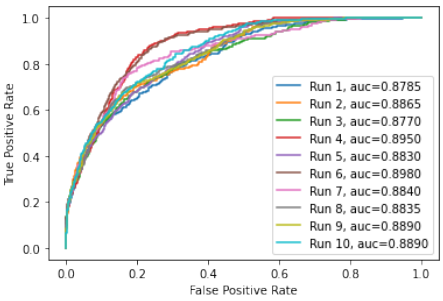
SI/SR

A.12 Experiment 12

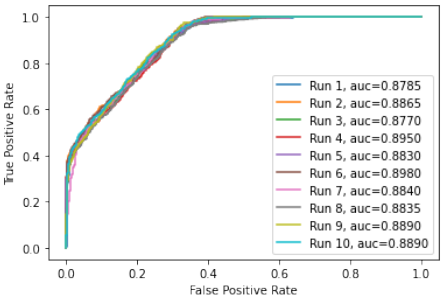




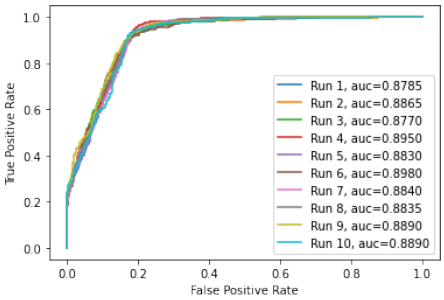
AF/AFIB



gSVT

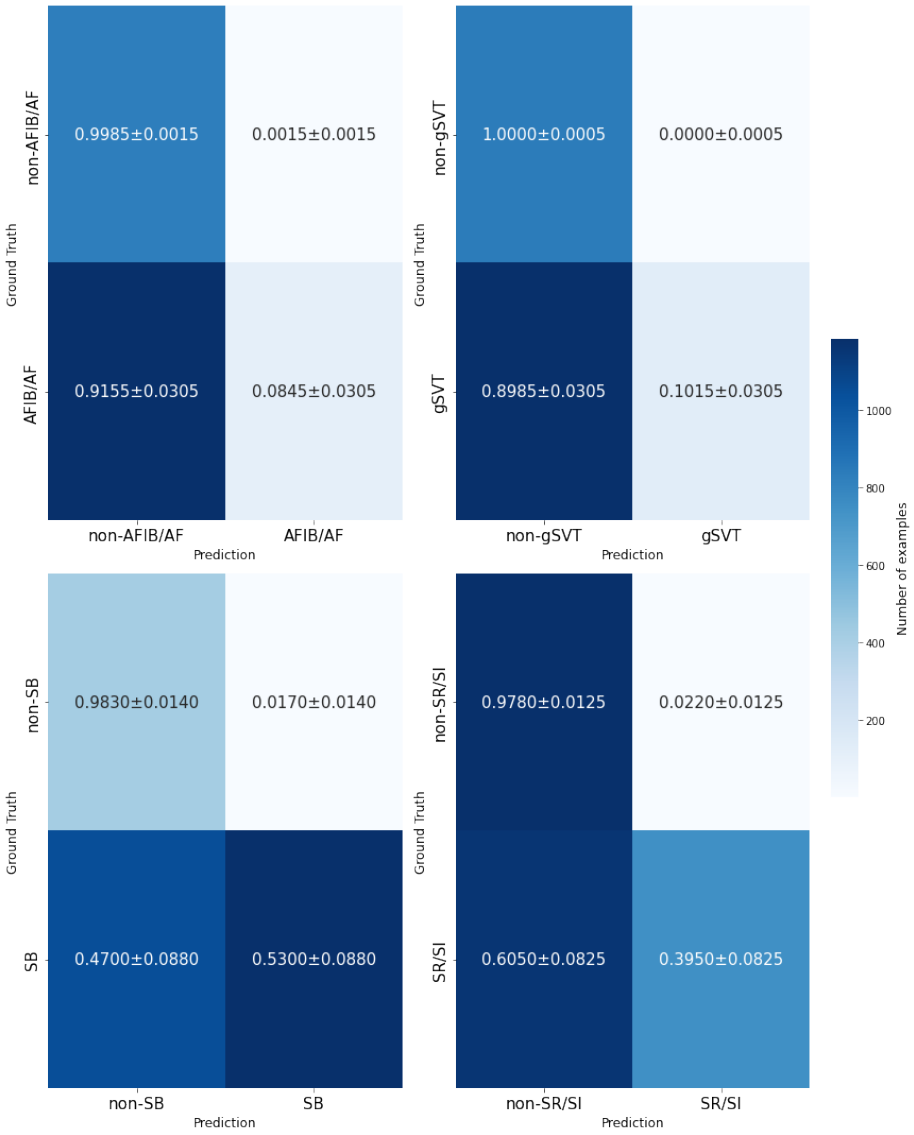


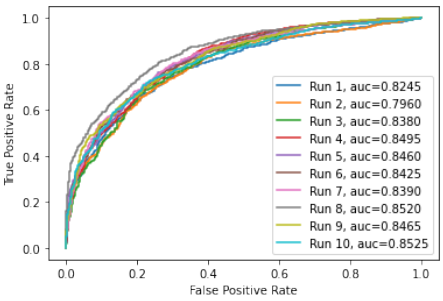
SB



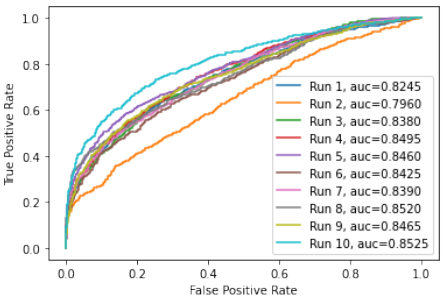
SI/SR

A.13 Experiment 13

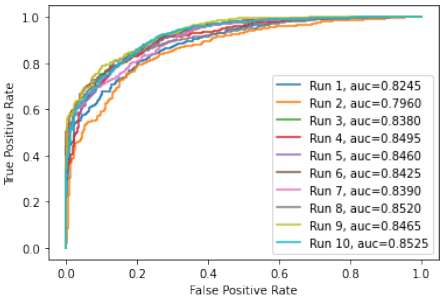




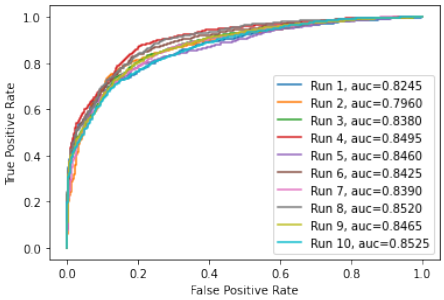
AF/AFIB



gSVT

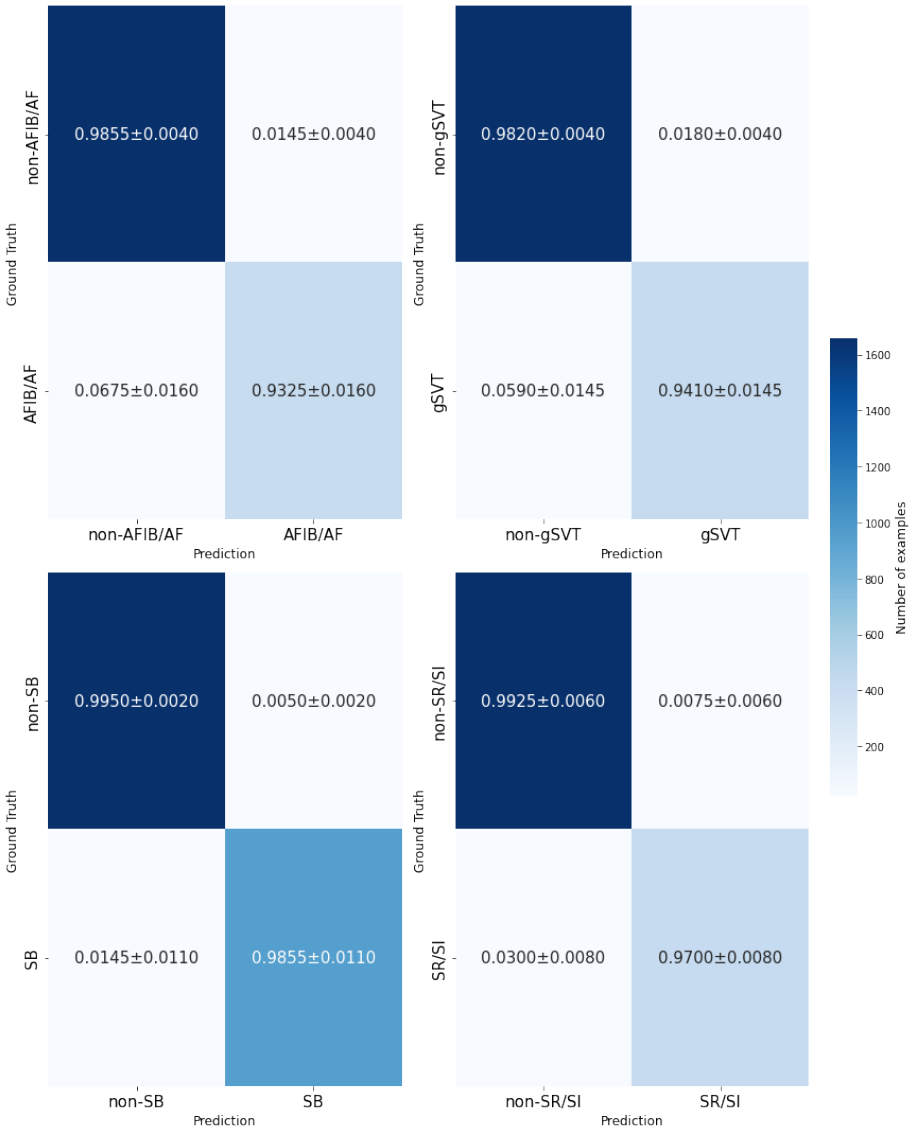


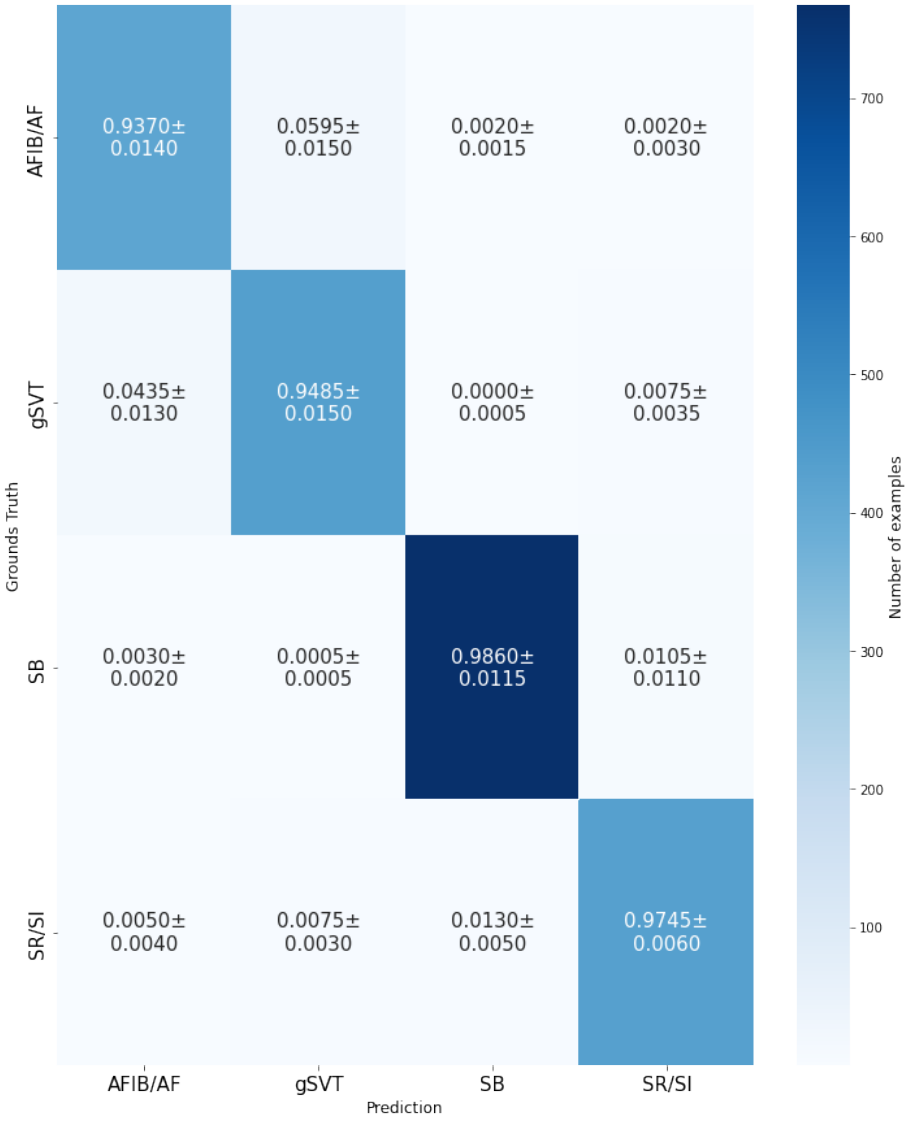
SB

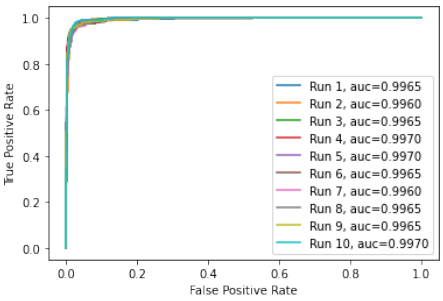


SI/SR

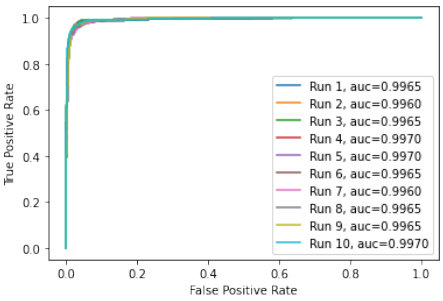
A.14 Experiment 14



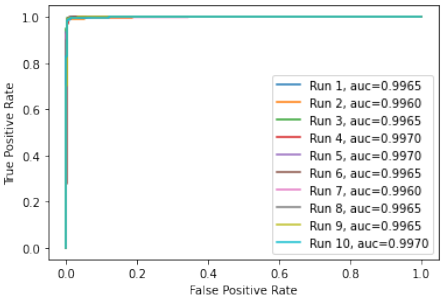

AFIB/AF
gSVT
SB
SR/SI
Grounds Truth
0.9370± 0.0140
0.0595± 0.0150
0.0020± 0.0015
0.0020± 0.0030
0.0435± 0.0130
0.9485± 0.0150
0.0000± 0.0005
0.0075± 0.0035
0.0030± 0.0020
0.0005± 0.0005
0.9860± 0.0115
0.0105± 0.0110
0.0050± 0.0040
0.0075± 0.0030
0.0130± 0.0050
0.9745± 0.0060
AFIB/AF
gSVT
SB
SR/SI
Prediction
Number of examples
700
600
500
400
300
200
100


True Positive Rate
False Positive Rate
Run 1, auc=0.9965
Run 2, auc=0.9960
Run 3, auc=0.9965
Run 4, auc=0.9970
Run 5, auc=0.9970
Run 6, auc=0.9965
Run 7, auc=0.9960
Run 8, auc=0.9965
Run 9, auc=0.9965
Run 10, auc=0.9970

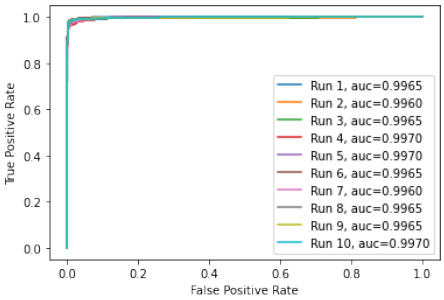
AFIB


True Positive Rate
False Positive Rate
Run 1, auc=0.9965
Run 2, auc=0.9960
Run 3, auc=0.9965
Run 4, auc=0.9970
Run 5, auc=0.9970
Run 6, auc=0.9965
Run 7, auc=0.9960
Run 8, auc=0.9965
Run 9, auc=0.9965
Run 10, auc=0.9970

SI

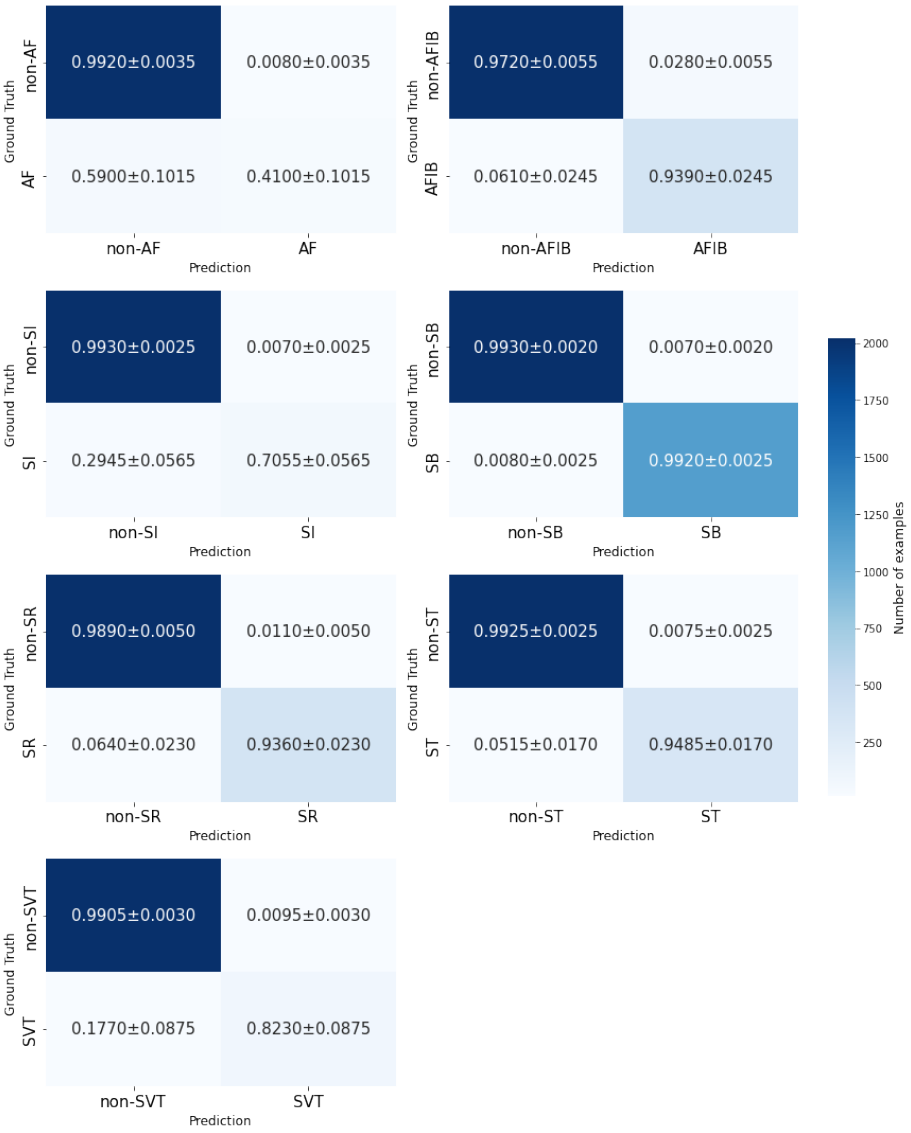

True Positive Rate
False Positive Rate
Run 1, auc=0.9965
Run 2, auc=0.9960
Run 3, auc=0.9965
Run 4, auc=0.9970
Run 5, auc=0.9970
Run 6, auc=0.9965
Run 7, auc=0.9960
Run 8, auc=0.9965
Run 9, auc=0.9965
Run 10, auc=0.9970

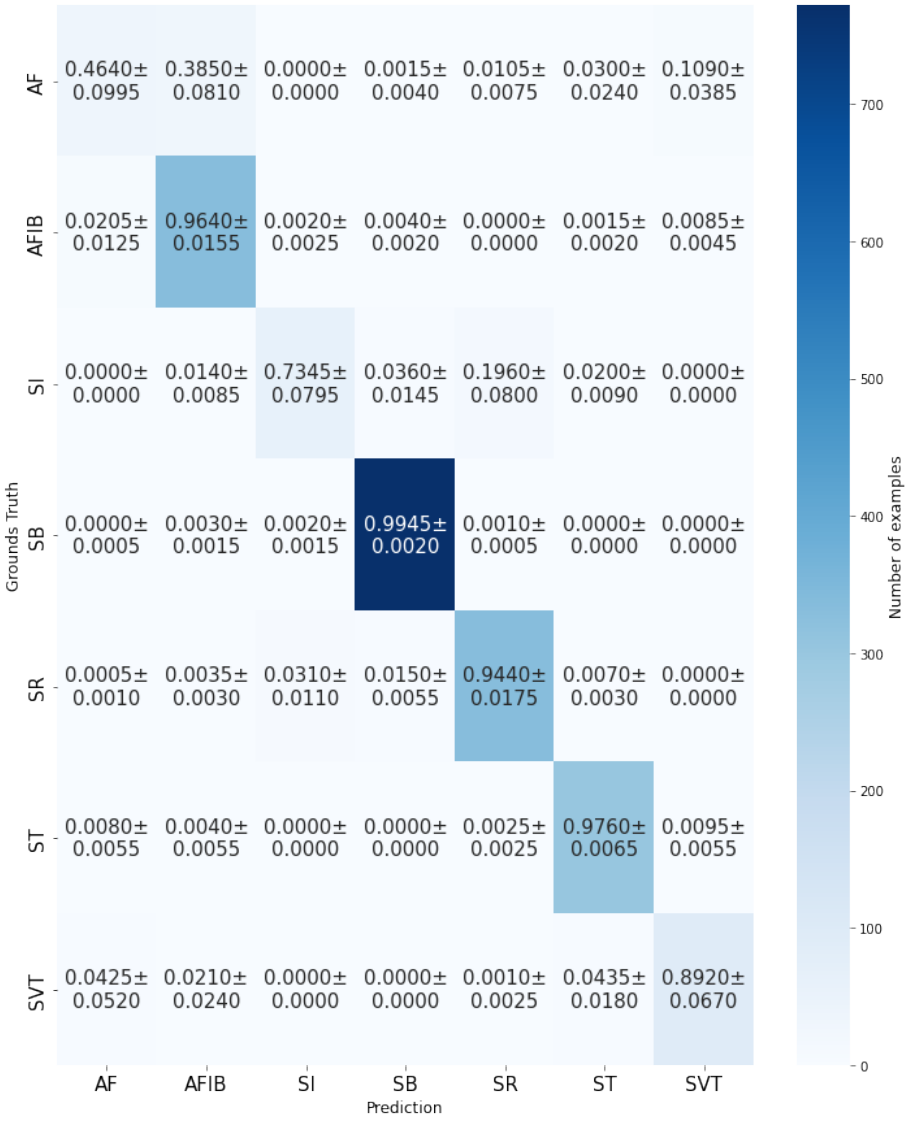
SB

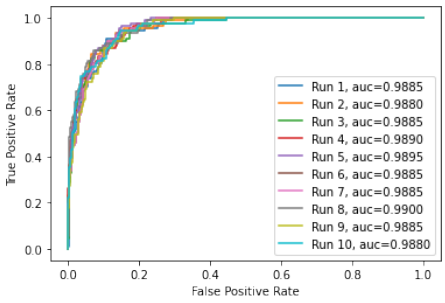

True Positive Rate
False Positive Rate
Run 1, auc=0.9965
Run 2, auc=0.9960
Run 3, auc=0.9965
Run 4, auc=0.9970
Run 5, auc=0.9970
Run 6, auc=0.9965
Run 7, auc=0.9960
Run 8, auc=0.9965
Run 9, auc=0.9965
Run 10, auc=0.9970

SR

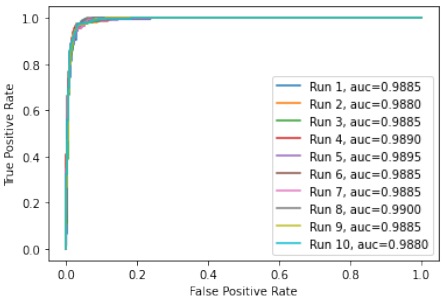
A.15 Experiment 15



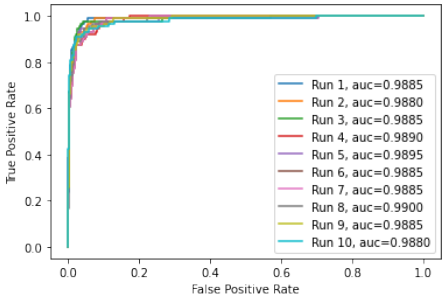

AF 0.4640± 0.0995 0.3850± 0.0810 0.0000± 0.0000 0.0015± 0.0040 0.0105± 0.0075 0.0300± 0.0240 0.1090± 0.0385
AFIB 0.0205± 0.0125 0.9640± 0.0155 0.0020± 0.0025 0.0040± 0.0020 0.0000± 0.0000 0.0015± 0.0020 0.0085± 0.0045
SI 0.0000± 0.0000 0.0140± 0.0085 0.7345± 0.0795 0.0360± 0.0145 0.1960± 0.0800 0.0200± 0.0090 0.0000± 0.0000
SB 0.0000± 0.0005 0.0030± 0.0015 0.0020± 0.0015 0.9945± 0.0020 0.0010± 0.0005 0.0000± 0.0000 0.0000± 0.0000
SR 0.0005± 0.0010 0.0035± 0.0030 0.0310± 0.0110 0.0150± 0.0055 0.9440± 0.0175 0.0070± 0.0030 0.0000± 0.0000
ST 0.0080± 0.0055 0.0040± 0.0055 0.0000± 0.0000 0.0000± 0.0000 0.0025± 0.0025 0.9760± 0.0065 0.0095± 0.0055
SVT 0.0425± 0.0520 0.0210± 0.0240 0.0000± 0.0000 0.0000± 0.0000 0.0010± 0.0025 0.0435± 0.0180 0.8920± 0.0670
Grounds Truth
AF AFIB SI SB SR ST SVT
Prediction
Number of examples
0 100 200 300 400 500 600 700



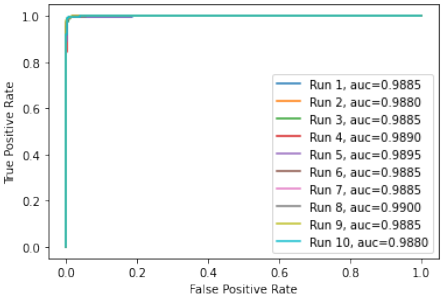
AF



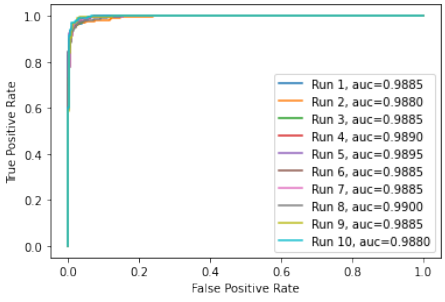
AFIB



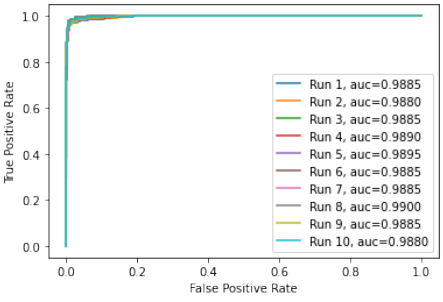
SI



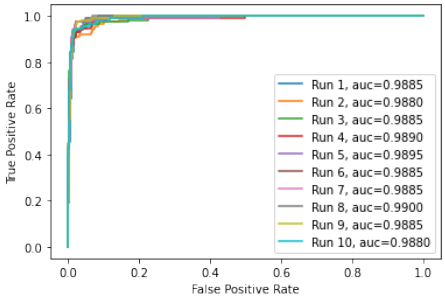
SB



SR

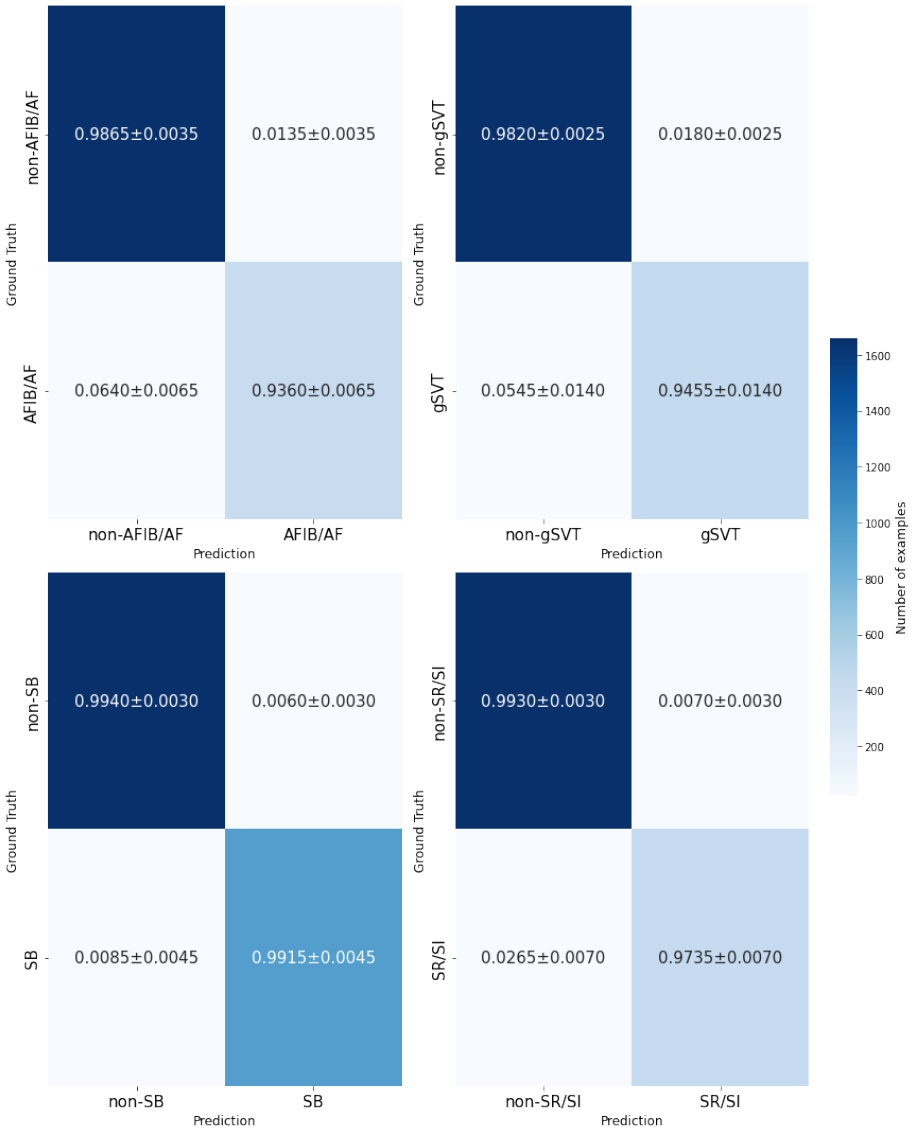


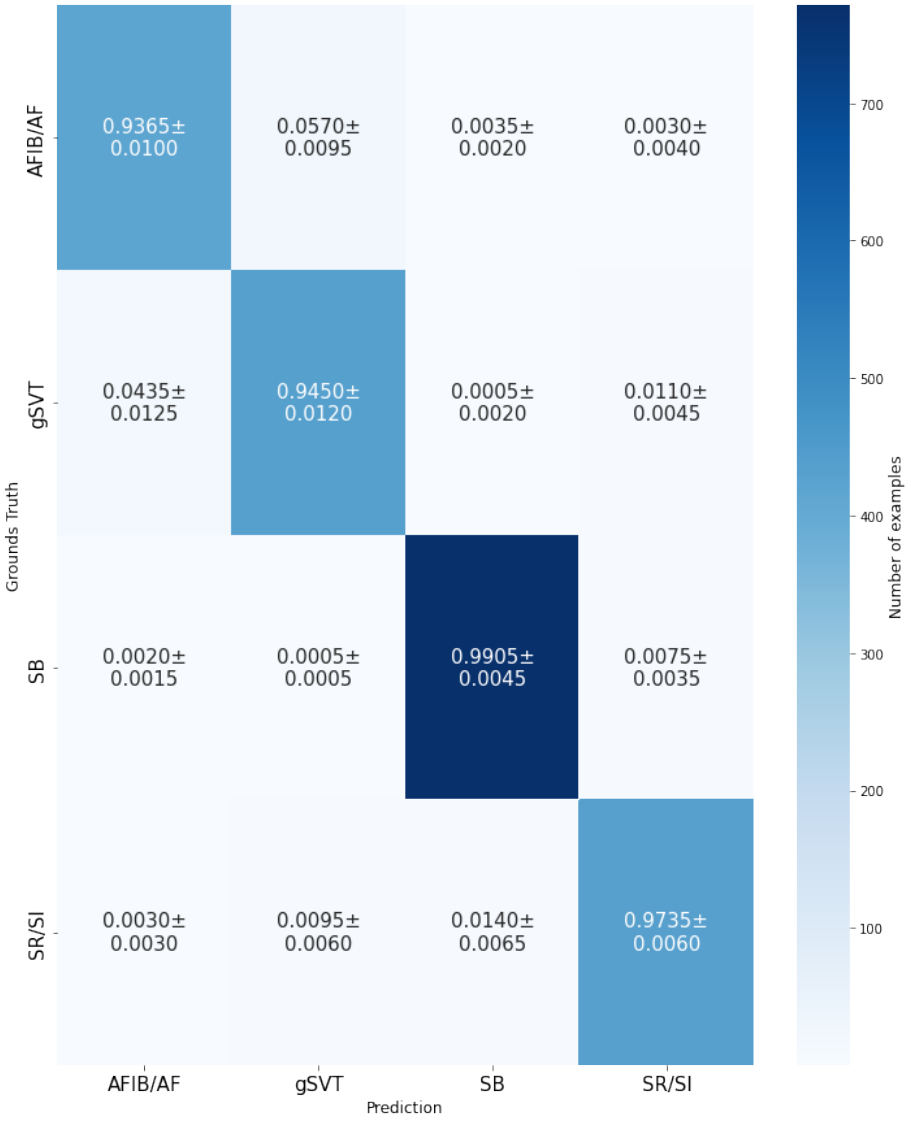
ST

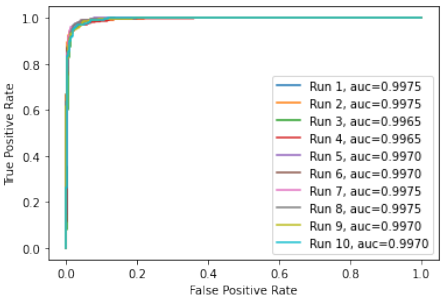


SVT

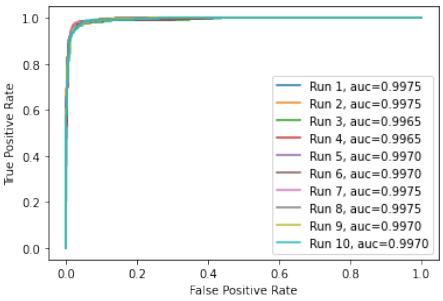
A.16 Experiment 16



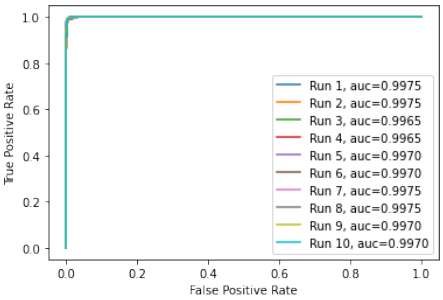

0.9365± 0.0100
0.0570± 0.0095
0.0035± 0.0020
0.0030± 0.0040
0.0435± 0.0125
0.9450± 0.0120
0.0005± 0.0020
0.0110± 0.0045
0.0020± 0.0015
0.0005± 0.0005
0.9905± 0.0045
0.0075± 0.0035
0.0030± 0.0030
0.0095± 0.0060
0.0140± 0.0065
0.9735± 0.0060
AFIB/AF
gSVT
SB
SR/SI
Grounds Truth
AFIB/AF
gSVT
SB
SR/SI
Prediction
700
600
500
400
300
200
100
Number of examples


True Positive Rate
False Positive Rate
Run 1, auc=0.9975
Run 2, auc=0.9975
Run 3, auc=0.9965
Run 4, auc=0.9965
Run 5, auc=0.9970
Run 6, auc=0.9970
Run 7, auc=0.9975
Run 8, auc=0.9975
Run 9, auc=0.9970
Run 10, auc=0.9970

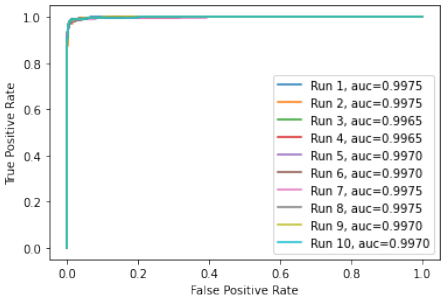
AFIB


True Positive Rate
False Positive Rate
Run 1, auc=0.9975
Run 2, auc=0.9975
Run 3, auc=0.9965
Run 4, auc=0.9965
Run 5, auc=0.9970
Run 6, auc=0.9970
Run 7, auc=0.9975
Run 8, auc=0.9975
Run 9, auc=0.9970
Run 10, auc=0.9970

SI

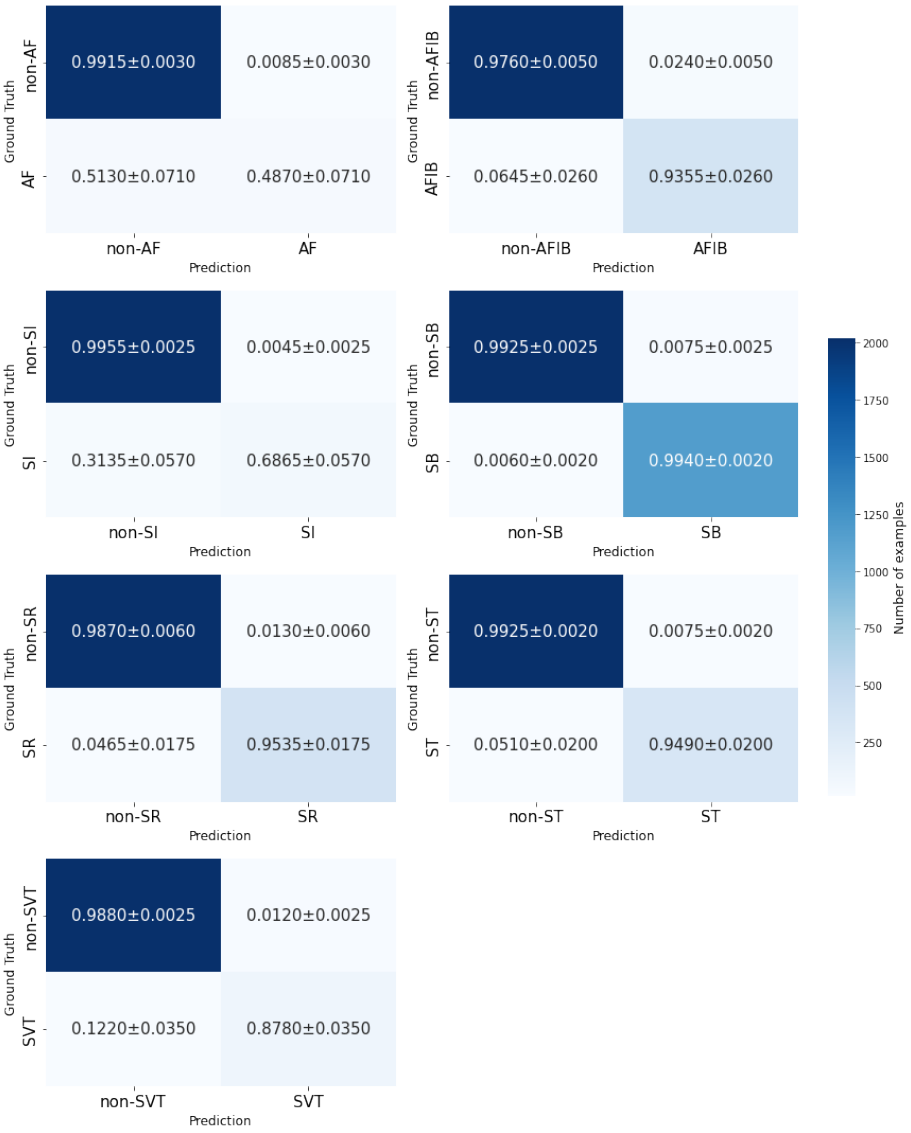

True Positive Rate
False Positive Rate
Run 1, auc=0.9975
Run 2, auc=0.9975
Run 3, auc=0.9965
Run 4, auc=0.9965
Run 5, auc=0.9970
Run 6, auc=0.9970
Run 7, auc=0.9975
Run 8, auc=0.9975
Run 9, auc=0.9970
Run 10, auc=0.9970

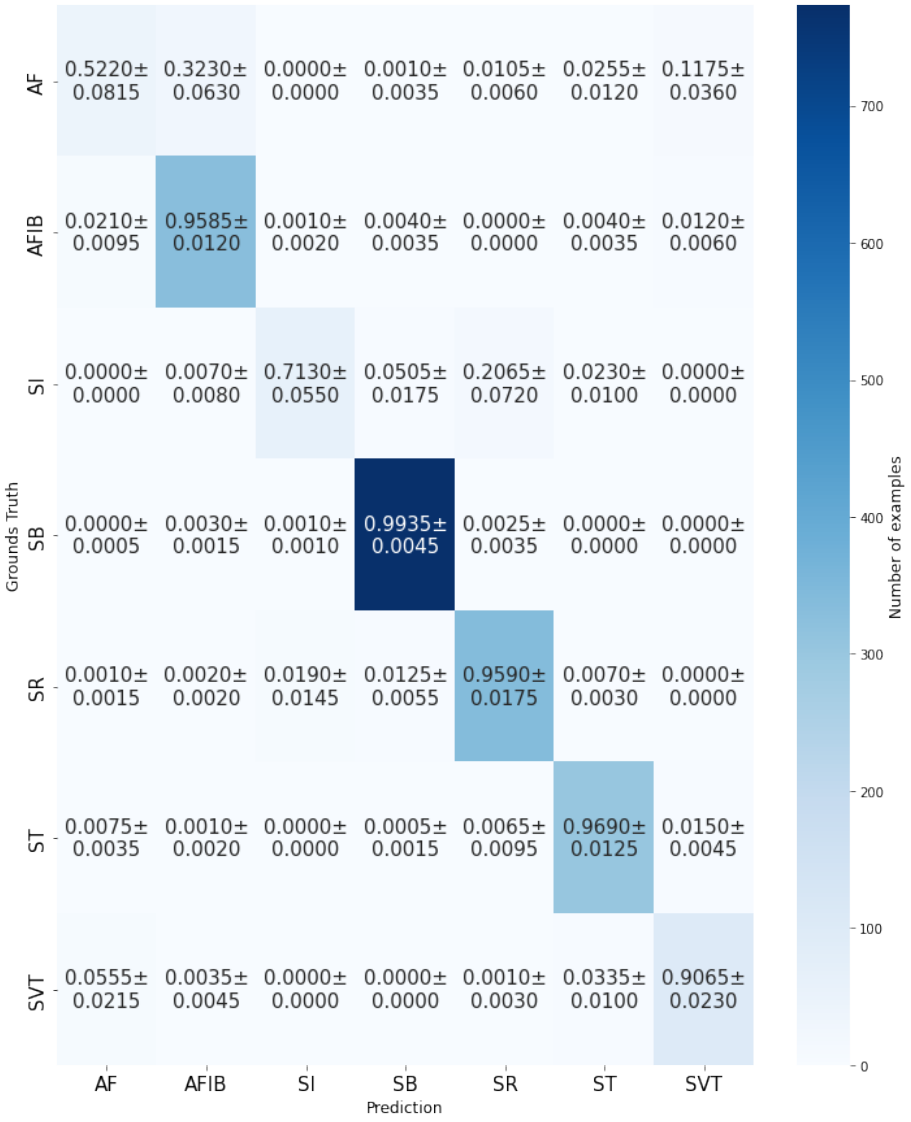
SB

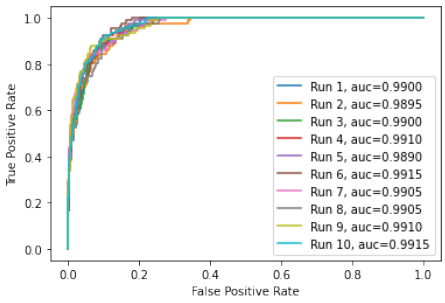

True Positive Rate
False Positive Rate
Run 1, auc=0.9975
Run 2, auc=0.9975
Run 3, auc=0.9965
Run 4, auc=0.9965
Run 5, auc=0.9970
Run 6, auc=0.9970
Run 7, auc=0.9975
Run 8, auc=0.9975
Run 9, auc=0.9970
Run 10, auc=0.9970

SR

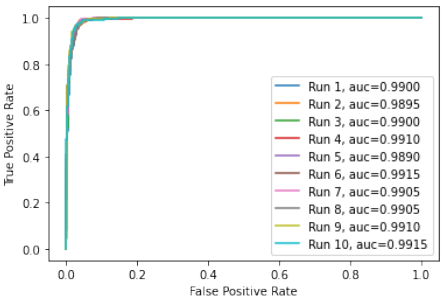
A.17 Experiment 17



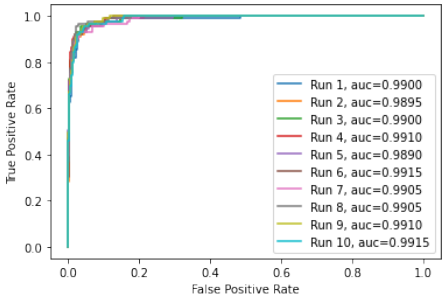

AF 0.5220± 0.0815 0.3230± 0.0630 0.0000± 0.0000 0.0010± 0.0035 0.0105± 0.0060 0.0255± 0.0120 0.1175± 0.0360
AFIB 0.0210± 0.0095 0.9585± 0.0120 0.0010± 0.0020 0.0040± 0.0035 0.0000± 0.0000 0.0040± 0.0035 0.0120± 0.0060
SI 0.0000± 0.0000 0.0070± 0.0080 0.7130± 0.0550 0.0505± 0.0175 0.2065± 0.0720 0.0230± 0.0100 0.0000± 0.0000
SB 0.0000± 0.0005 0.0030± 0.0015 0.0010± 0.0010 0.9935± 0.0045 0.0025± 0.0035 0.0000± 0.0000 0.0000± 0.0000
SR 0.0010± 0.0015 0.0020± 0.0020 0.0190± 0.0145 0.0125± 0.0055 0.9590± 0.0175 0.0070± 0.0030 0.0000± 0.0000
ST 0.0075± 0.0035 0.0010± 0.0020 0.0000± 0.0000 0.0005± 0.0015 0.0065± 0.0095 0.9690± 0.0125 0.0150± 0.0045
SVT 0.0555± 0.0215 0.0035± 0.0045 0.0000± 0.0000 0.0000± 0.0000 0.0010± 0.0030 0.0335± 0.0100 0.9065± 0.0230
Grounds Truth
AF AFIB SI SB SR ST SVT
Prediction
Number of examples
0 100 200 300 400 500 600 700



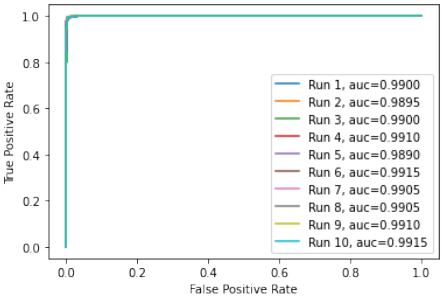
AF



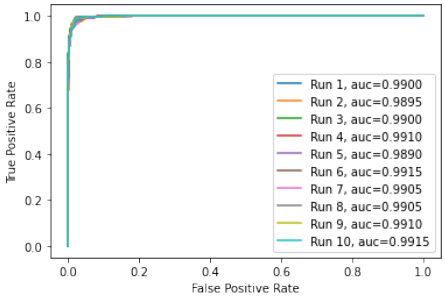
AFIB



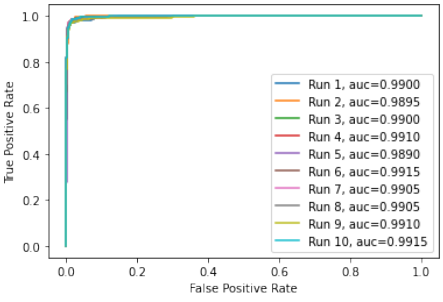
SI



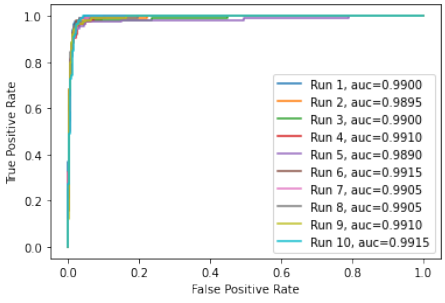
SB



SR

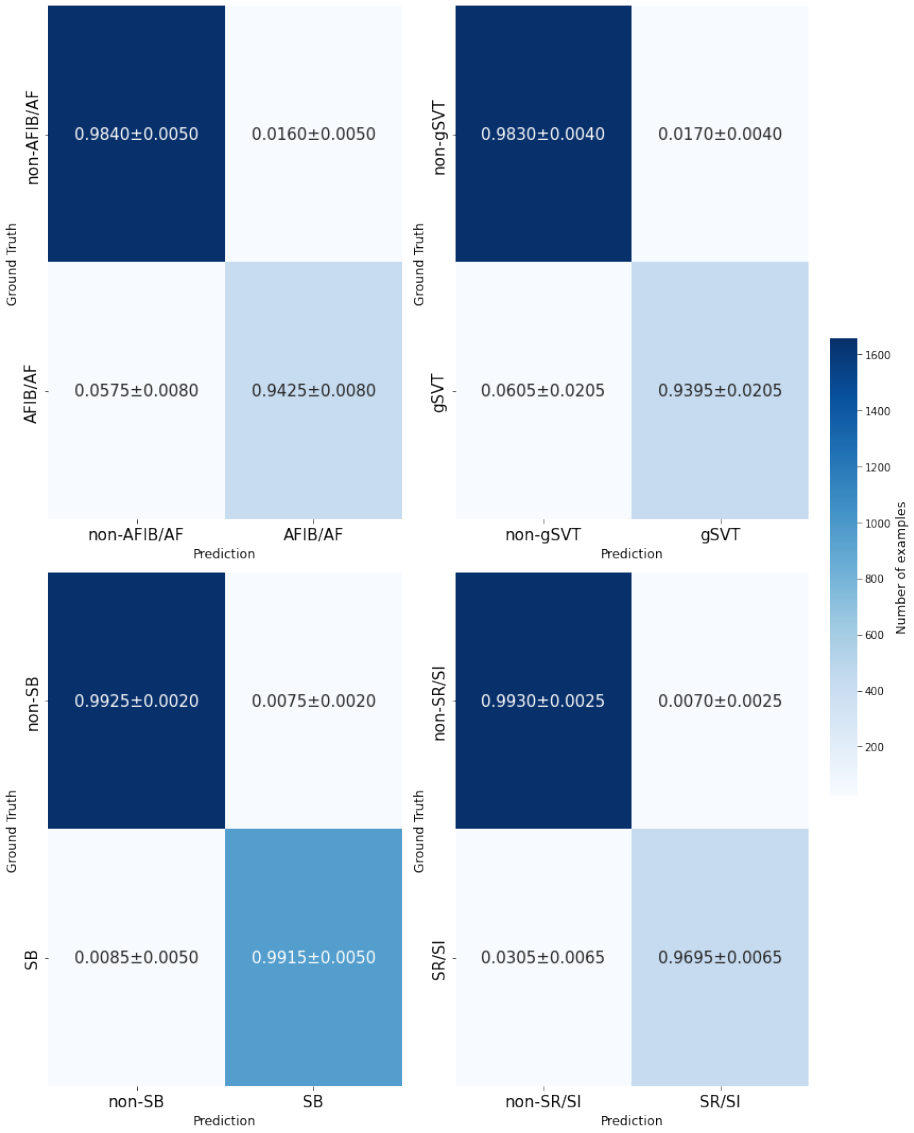


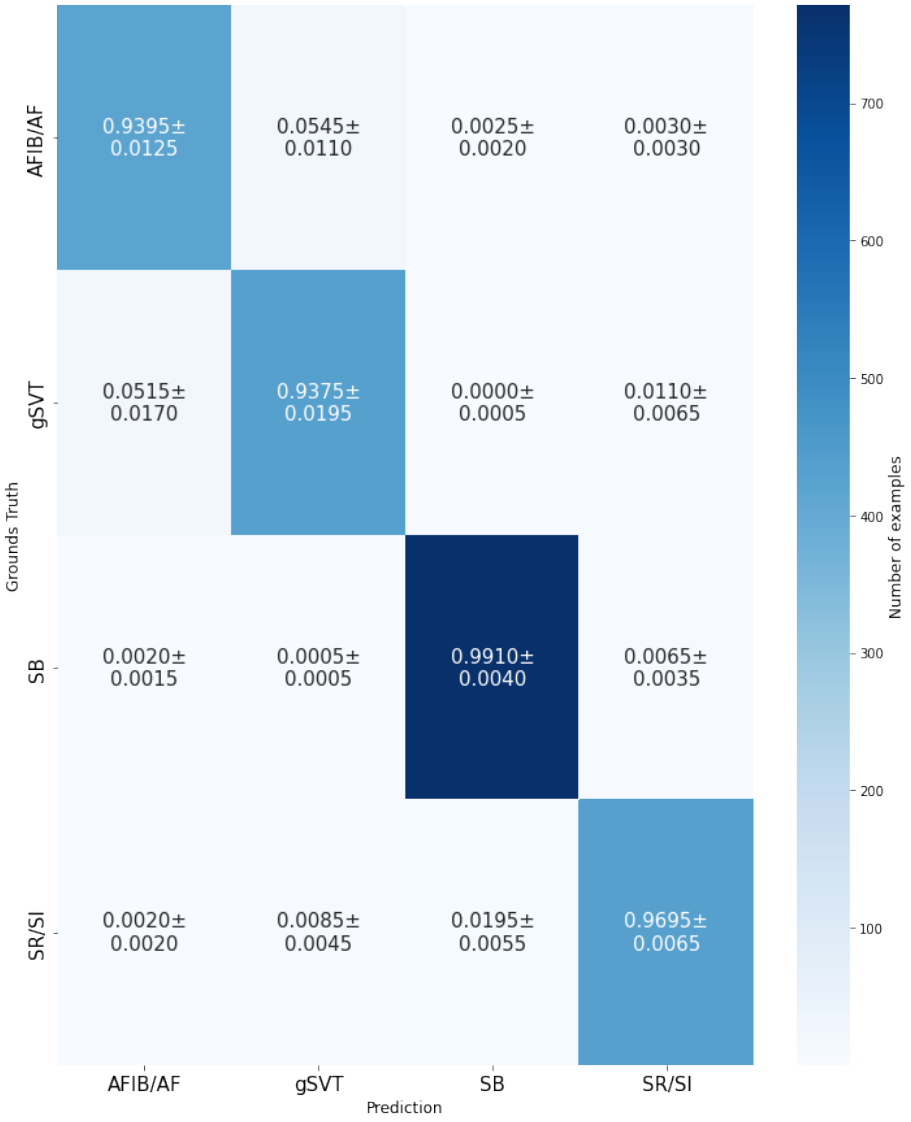
ST

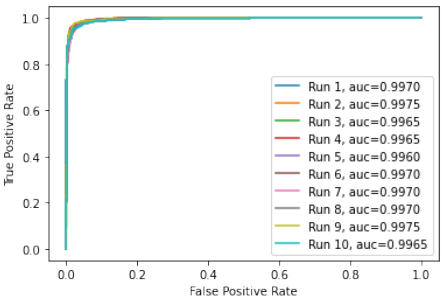


SVT

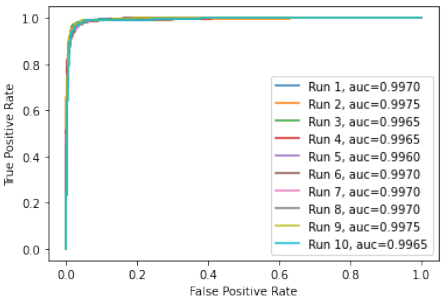
A.18 Experiment 18



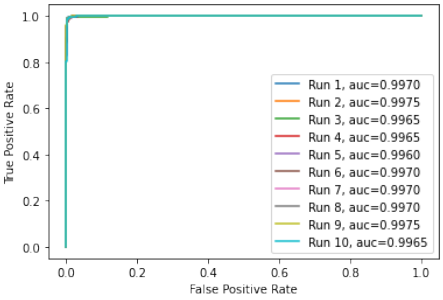

AFIB/AF
gSVT
SB
SR/SI
Grounds Truth
0.9395± 0.0125
0.0545± 0.0110
0.0025± 0.0020
0.0030± 0.0030
0.0515± 0.0170
0.9375± 0.0195
0.0000± 0.0005
0.0110± 0.0065
0.0020± 0.0015
0.0005± 0.0005
0.9910± 0.0040
0.0065± 0.0035
0.0020± 0.0020
0.0085± 0.0045
0.0195± 0.0055
0.9695± 0.0065
AFIB/AF
gSVT
SB
SR/SI
Prediction
Number of examples
700
600
500
400
300
200
100


True Positive Rate
False Positive Rate
Run 1, auc=0.9970
Run 2, auc=0.9975
Run 3, auc=0.9965
Run 4, auc=0.9965
Run 5, auc=0.9960
Run 6, auc=0.9970
Run 7, auc=0.9970
Run 8, auc=0.9970
Run 9, auc=0.9975
Run 10, auc=0.9965

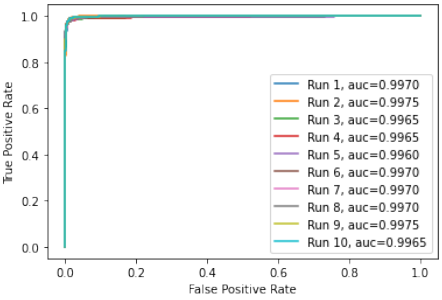
AFIB


True Positive Rate
False Positive Rate
Run 1, auc=0.9970
Run 2, auc=0.9975
Run 3, auc=0.9965
Run 4, auc=0.9965
Run 5, auc=0.9960
Run 6, auc=0.9970
Run 7, auc=0.9970
Run 8, auc=0.9970
Run 9, auc=0.9975
Run 10, auc=0.9965

SI

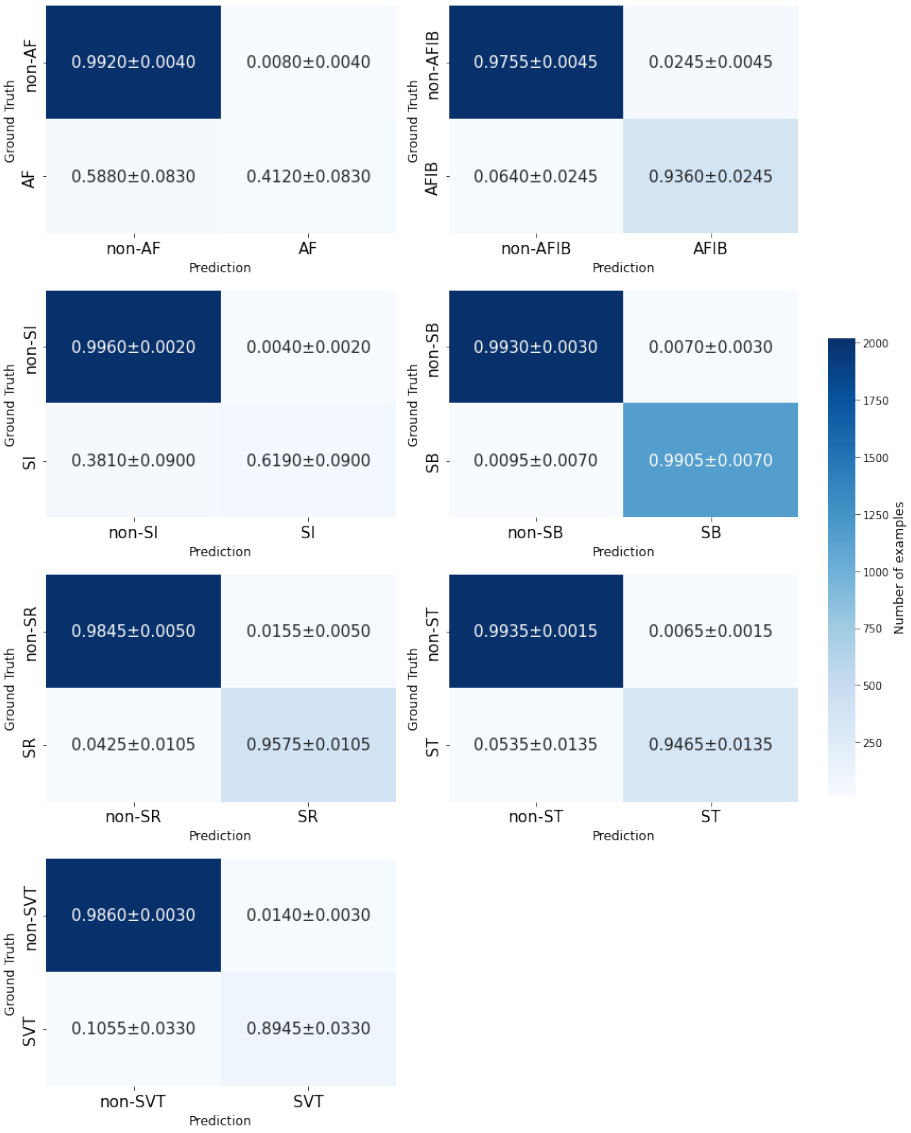

True Positive Rate
False Positive Rate
Run 1, auc=0.9970
Run 2, auc=0.9975
Run 3, auc=0.9965
Run 4, auc=0.9965
Run 5, auc=0.9960
Run 6, auc=0.9970
Run 7, auc=0.9970
Run 8, auc=0.9970
Run 9, auc=0.9975
Run 10, auc=0.9965

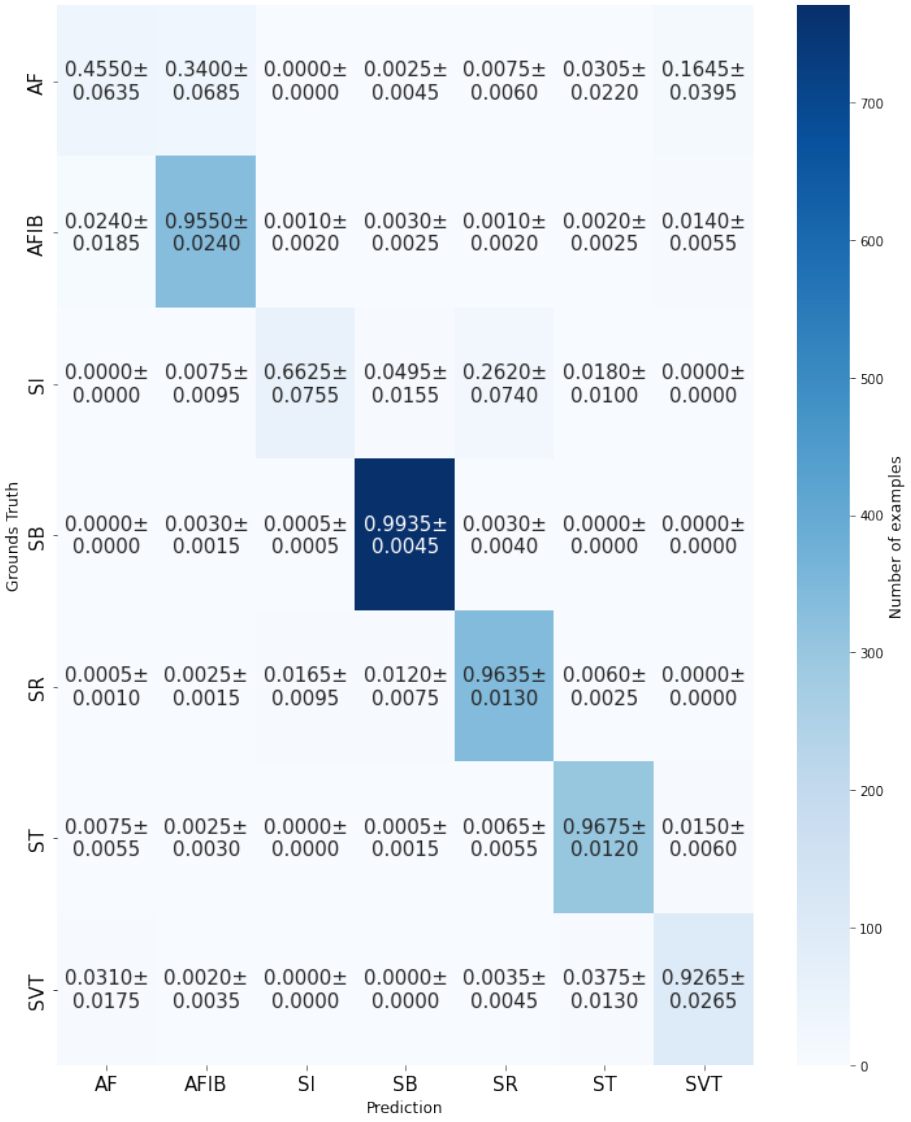
SB

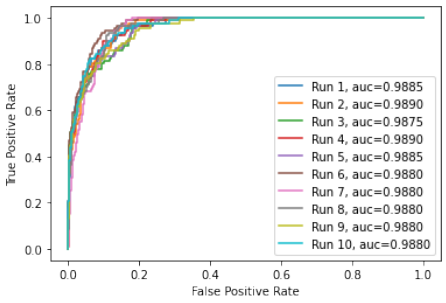

True Positive Rate
False Positive Rate
Run 1, auc=0.9970
Run 2, auc=0.9975
Run 3, auc=0.9965
Run 4, auc=0.9965
Run 5, auc=0.9960
Run 6, auc=0.9970
Run 7, auc=0.9970
Run 8, auc=0.9970
Run 9, auc=0.9975
Run 10, auc=0.9965

SR

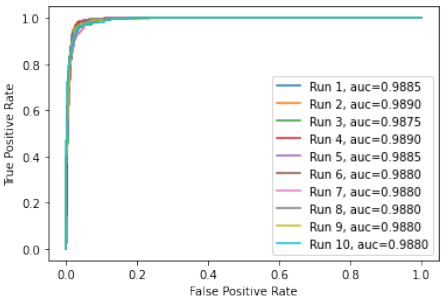
A.19 Experiment 19



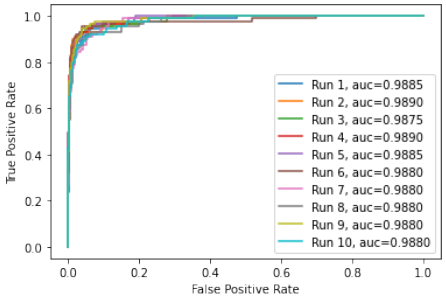

AF 0.4550± 0.0635 0.3400± 0.0685 0.0000± 0.0000 0.0025± 0.0045 0.0075± 0.0060 0.0305± 0.0220 0.1645± 0.0395
AFIB 0.0240± 0.0185 0.9550± 0.0240 0.0010± 0.0020 0.0030± 0.0025 0.0010± 0.0020 0.0020± 0.0025 0.0140± 0.0055
SI 0.0000± 0.0000 0.0075± 0.0095 0.6625± 0.0755 0.0495± 0.0155 0.2620± 0.0740 0.0180± 0.0100 0.0000± 0.0000
SB 0.0000± 0.0000 0.0030± 0.0015 0.0005± 0.0005 0.9935± 0.0045 0.0030± 0.0040 0.0000± 0.0000 0.0000± 0.0000
SR 0.0005± 0.0010 0.0025± 0.0015 0.0165± 0.0095 0.0120± 0.0075 0.9635± 0.0130 0.0060± 0.0025 0.0000± 0.0000
ST 0.0075± 0.0055 0.0025± 0.0030 0.0000± 0.0000 0.0005± 0.0015 0.0065± 0.0055 0.9675± 0.0120 0.0150± 0.0060
SVT 0.0310± 0.0175 0.0020± 0.0035 0.0000± 0.0000 0.0000± 0.0000 0.0035± 0.0045 0.0375± 0.0130 0.9265± 0.0265
Grounds Truth
AF AFIB SI SB SR ST SVT
Prediction
Number of examples
0 100 200 300 400 500 600 700



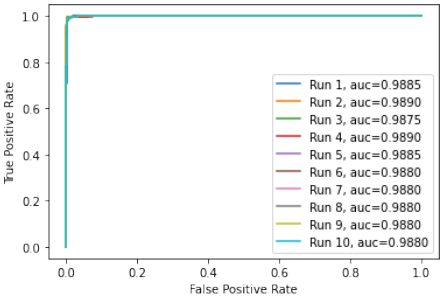
AF



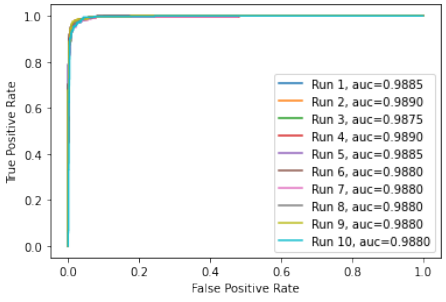
AFIB



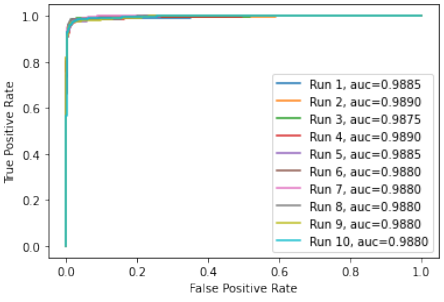
SI



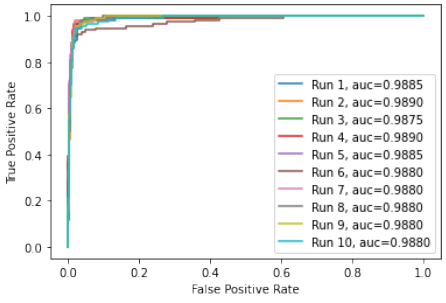
SB



SR



ST



SVT

Appendix B

Alternative Sets of Hyper Parameters for the Proposed Architecture

Hyper Parameter	Set 1	Set 2	Set 3	Set 4	Set 5
N.o. filters in CNN-layer 1	128	128	64	256	32
Filter-size in CNN-layer 1	8	24	32	16	4
Number of filters in CNN-layer 2	16	256	256	64	16
Filter-size in CNN-layer 2	10	32	4	14	10
Number of filters in CNN-layer 3	64	64	64	64	64
Filter-size in CNN-layer 3	12	10	12	16	10
Size of CNN-pooling-layer	8	10	6	6	8
N.o. *encoder*-layers	2	1	1	3	4
N.o. attention-heads in enc-layer	4	16	3	1	4
N.o. neurons in t.d. FC-layer	5098	1024	1024	1024	2048
N.o. neurons in pre-final FC-layer	64	32	128	128	256