AUTOMATIC DETECTION FOR CARDIAC ARRHYTHMIA BASED ON ECG AND DEEP LEARNING APPROACHES

A thesis submitted to the University of Manchester
for the degree of Doctor of Philosophy
in the Faculty of Science & Engineering

2023

ZICONG LI
Department of Physics and Astronomy

Contents

Word count 32161 words

List of Tables

List of Figures

The University of Manchester

ZICONG LI

Doctor of Philosophy

Automatic detection for cardiac arrhythmia based on ECG and deep learning approaches

February 15, 2023

Electrocardiograms (ECG) provide information about the electrical activity of the heart, which is useful for diagnosing abnormal cardiac functions such as arrhythmias. Over the works of this thesis, the author aims to provide novel auto-detection approaches for arrhythmia diagnosis based on advanced deep learning methods and preprocessing techniques. The first presented auto-detection method proposed a novel frame-blocking-based preprocessing method that divides a 12-lead ECG recording into frames of a uniform length. The novel preprocessing method addressed the uneven length of clinical signals and limited the loss of valid signals. The multi-labeled classification is then split into several binary classification tasks, with each binary classifier consisting of an attention-based BiLSTM and a ResNet-based network. The advanced preprocessing technique and structure of classifiers fulfilled multi-label classification and achieved a satisfied average F1 score of 0.908. The second presented project utilized an advanced metric-learning algorithm in the training process and proposed to extract comprehensive ECG features on both morphological and temporal domains. The metric-learning-based training model and fused features contributed to more discriminative ECG features than traditional training models merely based on morphological features. With a relatively small model size and GFLOPs, the proposed algorithm achieved an average score of 0.874 and exhibits promising efficiency. Since some rare arrhythmias caused extremely insufficient training samples, the classification performance of the previous classification algorithm can be negatively affected. The third project presented a parallel multi-scale convolution based prototypical network (PM-CNN ProtoNet) for processing the few-shot learning tasks of ECG beats classification. The presented method, which represents a novel attempt to use few-shot learning in ECG auto-detection, exhibits a competitive result when compared to other state-of-the-art models and also demonstrates its potential to address the issue of limited training samples as well as real-world medical applications.

Declaration

No portion of the work referred to in the thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

Copyright Statement

Acknowledgements

First and foremost I would like to thank my supervisor Prof. Henggui Zhang for his supervisor and guidance. I appreciate that he gives me this opportunity to conduct the research topic under his supervision and support. I am also grateful to my family for their patience and encouragement. Their unconditional love gives me motivation to move on. Finally, I would like to give a special thanks to my boyfriend. His companionship and understanding contribute to the huge support throughout my studies.

Dedication

To my parents, Bing Li and Yingjie Liu, who always support me to make choices independently during my life and inspire me to be the best that I can be. The work cannot be accomplished without their love and support.

The author

Zicong Li was born in 1996 in Shandong, China. She received the B.S. degree in university of Jinan of computer science and technology, China, in 2018. In September 2018, she joined the Biological Physics Group at the school of Physics and Astronomy in the university of Manchester under the supervision of Prof. Henggui Zhang for pursuing her PhD degree.

Support Publications

Li, Z. and Zhang, H., 2021. Automatic Detection for Multi-Labeled Cardiac Arrhythmia Based on Frame Blocking Preprocessing and Residual Networks. *Frontiers in cardiovascular medicine*, 8, p.616585.

Z. Li and H. Zhang, Parallel Multi-scale convolution based prototypical network for few-shot ECG beats classification. *2022 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)* Ioannina, Greece, 2022, pp. 1-4

Chapter 1

Introduction

1.1 The Heart

The heart, as one of the most important organs in the human body, continuously pumps blood throughout the cardiovascular system around the human body [1, 2]. The human heart pumps blood to deliver oxygen and nutrients to cells through a network of blood arteries (circulatory system), preserving the healthy operation of human organs. The electrical conduction system is responsible for regulating the pumping actions of the human heart. Normally, the sinoatrial (SA) node generates and delivers electrical impulses at a regular rate approximately 60 to 100 times per minute [3], but the heart rate can alter in terms of different human activities. The electrical activity of the heart can be recorded as electrocardiogram (ECG) which is a composite recording of all action potential produced by cells of the myocardium [4]. Therefore, ECG is essential for depicting the electrical activity of the cardiovascular system and supporting clinical diagnosis during the clinical examination.

Pathological alterations such as heart failure, cardiac arrhythmia, congenital heart diseases and cardiomyopathy that occur to heart functions can be fatal [5]. According to the study on the global burden of disease (GBD) [6], the total number of death brought on by cardiovascular diseases (CVDs) rosed by approximately 21.1% from 2007 to 2017. Cardiovascular diseases cause a significant financial and economic burden in addition to the high death rate. In terms of the statistics from Public Health England [7], the economic cost of CVD can rise from £7.4 billion each year to £15.8 billion when

the wider economic costs are involved. Additionally, conventional diagnosis approaches rely significantly on clinical specialists with sufficient experience, further increasing the labour costs. Therefore, it is vital to improve early detection and automatic diagnosis technologies in order to avert severe cardiac consequences.

In the last few decades, various machine-learning-based approaches [8–13] have been proposed for the automatic detection of cardiac arrhythmia. These studies aim to extract the physiological features from ECG signals and then adopt machine-learning approaches as classifiers for diagnosis. However, these studies require very complex pre-processing for obtaining discriminative features, which is also time-consuming [9]. A research challenge is still the automatic diagnosis of different forms of heart illnesses based on ECG. To further improve the efficiency and clinical applicability of automatic detection of cardiac arrhythmia, novel approaches should be proposed.

1.2 Automatic detection for cardiac arrhythmia

Over years, the automatic detection approaches for cardiac arrhythmia have developed rapidly for assisting the clinical diagnosis and reducing labour expenses. These approaches serve as a computer-aid tool for automatically extracting features from ECG and classifying different cardiac arrhythmias. Such approaches can be categorised into two general categories: adopting features as input or adopting ECG signals as input.

For the first category, features extracted from original ECG signals are utilized as input for machine-learning classifiers such as support vector machine (SVM) [14–20], K nearest neighbours (KNN) [21–23], and Decision Tree [8, 24–26] etc. Moreover, implementation of such approaches contains feature extraction methods such as principle component analysis (PCA) [27–30], discrete wavelet transform (DWT) [15, 25, 31–33], Empirical mode decomposition (EMD) [34–36] etc. Except for SVM, which necessitates a huge computational cost under large-scale datasets and high dimensional features, the majority of machine-learning classifiers conduct training and testing at less computational cost. The information required by classifiers is constrained by feature selection in the studies stated above, which require more computation in the stage

of extracting features from ECG in temporal and morphological attributes. Therefore, the proper feature extraction and selection algorithms remain challenges here.

Approaches in the second category allow the ECG signals as direct input, processing classification tasks via Neural Networks (NN). Due to the numerous hidden layers and nodes in neural networks, these approaches crunch a vast number of samples to extract discriminating information for model training, which requires a significant amount of computing. Nonetheless, these approaches can automatically extract comprehensive information, which is more applicable for complex classification tasks. In earlier research, Ozbay et al. [37] proposed a model based on artificial neural network (ANN) to classify 10 different abnormal arrhythmias with low average error rate of 4.3% for single classification. The advent of convolutional neural network [38] promotes the development of automatic classification for cardiac arrhythmia. In the study [39], authors adopted a 9-layers deep convolutional neural network (DCNN) to classify 5 types of heartbeats from the MIT-BIH dataset, and obtained the accuracy with 94.03%. The combination of DCNN and long-short term memory(LSTM) [40] obtained a higher accuracy with 99.3% for heartbeats classification on the same dataset. For more complicated classification tasks and ECGs, CNN combined with LSTM still performed well with 82.21% F1 score in arrhythmias classification [41]. The current generation of arrhythmia classification intends to use advanced deep-learning models to further improve accuracy and address the class imbalance. Recent studies [42, 43] utilize generative adversarial networks (GAN) to generate synthetic ECG signals for data augmentation and provided a new thought for deep-learning based arrhythmia detection.

Although these studies show encouraging accuracy, the limitations associated with deep learning models and clinical practicability still persist. Clinical ECG signals are usually of different lengths while CNN cannot accept varied-length signals as input. For automatic classification tasks, the proper pre-processing is required. Class imbalance is a common issue in the automatic classification of cardiac arrhythmia. Synthetic data can augment the training data whereas increasing the computational cost [44]. For rare cardiac arrhythmia, the extant amount of samples is insufficient

for synthesizing signals, which still remains challenges. Besides classification accuracy, the research of automatic classification for cardiac arrhythmia should consider the clinical applicability. This enables automatic classification algorithms to explore the advancing interdisciplinary research and demonstrate their potential in replacing manual diagnosis.

1.3 Thesis Overview

The two main aims of that thesis are:

1. To implement ECG automatic detection and classification approaches based on advanced deep learning techniques, contributing to the clinical auxiliary diagnosis and reducing manual cost.

2. To comprehensively consider the practicability of the approaches, trying to improve the approaches in aspects of classification performance, model size, computation usage and training algorithm.

The whole thesis is grouped into three main parts: Part I contains three chapters that provided the introduction, scientific background and review of automatic ECG analysis approaches. Part II involves the research of three chapters about the thesis title. Part III consists of the final chapter for comprehensive discussion and conclusion.

Consequently, the thesis is divided into seven chapters, each of which can be summarised as follows: Chapter 1 provides a basic introduction to the background and development of automatic detection for cardiac arrhythmia. Chapter 2 presents the biological, computational and technical background of the projects in that thesis. Chapter 3 reviewed the studies presented so far regarding the use of machine learning and deep learning for ECG automatic detection and classification.

In chapter 4, a novel preprocessing method and ResNet-based neural network is implemented for conducting multi-labelled 12-lead ECG classifications. Frame blocking, as a novel usage in ECG preprocessing and segmentation, divided an original 12-lead ECG recording into frames with a uniform length thus meeting the requirement of

neural network inputs. Multi-labelled classification is decomposed into binary classification tasks and the proposed neural network work as each binary classifier showed satisfying results. Through the experiments, the novel preprocessing method and the proposed network were shown to be an improved method for auto-classifying multiple types of arrhythmias when compared to recent studies.

Chapter 5 focuses on an advanced deep learning method-metric learning which is common-used in the computer vision domain. This work proposed a novel encoder-decoder model for ECG classification. In terms of the encoder, the combination of ResNet-based neural network and metric learning can extract discriminative features whilst reducing the complexity of network training. Furthermore, this work adopts feature fusion and offers thorough feature maps with morphological and temporal features. The experimental results show that the deep metric-based model with feature fusion is capable of accurately classifying a variety of arrhythmias

Chapter 6 concentrates on an advanced deep learning technique which is not involved in previous studies. This study aims to identify the unseen ECG data based only on a small number of training samples by using few-shot learning in ECG auto-classification. This chapter describes an ECG beats classification method based on few-shot learning tasks using a parallel multi-scale convolution-based prototype network. Through the evaluation, the proposed model displays results that are competitive with other state-of-art models in the few-shot learning domain.

As the final discussion, chapter 7 discusses the major findings, significance, novelty contribution, limitations and future works of each work presented in this thesis.

Chapter 2

Background

2.1 Electrocardiogram

The electrocardiogram is significant in the clinical cardiology domain since the features of ECG can provide diagnostic bases for cardiac arrhythmia [45]. ECG can reflect changes in the electrical activity of the heart with time during the propagation process of the action potentials throughout the heart at each cardiac cycle [46]. As for normal heartbeats, the electrical impulse produced by sinoatrial (SA) node propagates orderly throughout the heart [47]. Figure 2.1 illustrates the pathway of the electrical impulse in the cardiac conduction system. When the electrical impulse starts to spread throughout the atria, the electric potential difference between the interior and exterior surfaces of membrane of atrial myocardial cells rapidly approaches zero, which process is known as depolarization and causes atrial contraction [48]. Then, the electrical current passes through the atrioventricular (AV) node located in the right atrium and further spreads to the muscles of the ventricles via Purkinje fibres, causing ventricles to contract. The atria are gradually regaining the electric potential difference between the interior and exterior cell membrane surfaces when the ventricles are being depolarized, which is known as repolarization [49]. ECG can depict the complete process of depolarization (contraction) and repolarization(relaxation) of the heart, reflecting the health conditions of the heart.

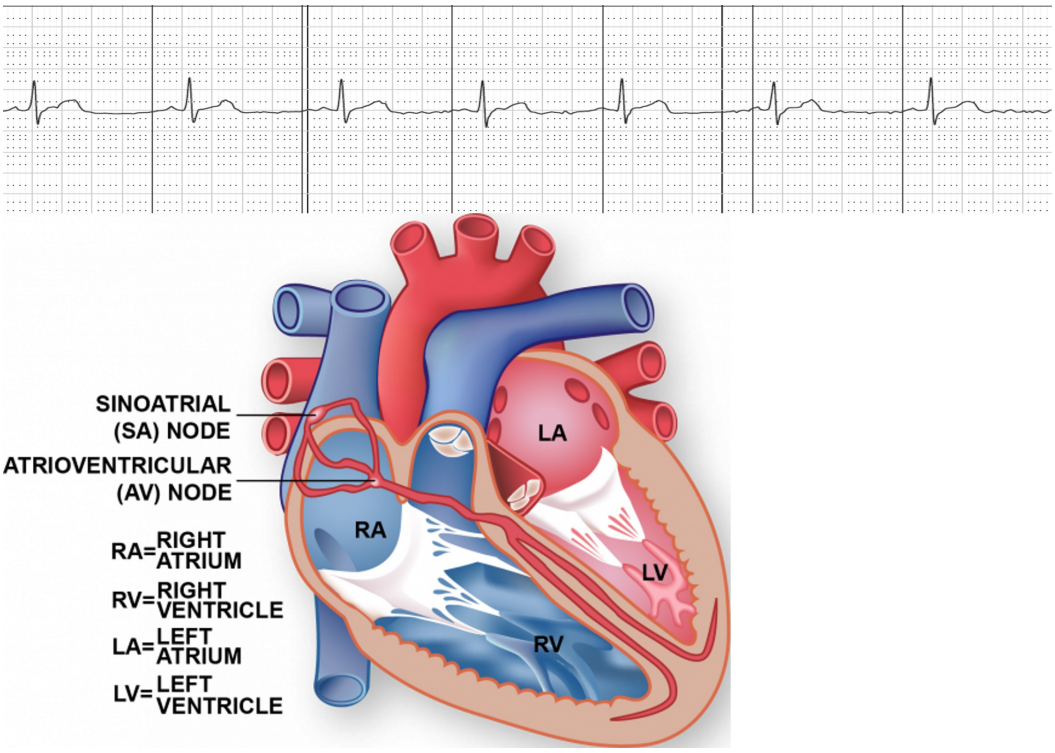

Figure 2.1: The simplified version of the pathway of conduction system. Image taken from [50]

2.1.1 ECG Cycle Details

As shown in Figure 2.2, a typical ECG complex consists of different waves, that represent the depolarization and repolarization of specific parts of the heart, where the horizontal axis represents time and the vertical axis represents voltage. P wave describes the movement of the depolarization of the atria by the electrical impulse released from SA node [51]. The P-R segment represents a pause in electrical activity and the P-R interval represents the time from the beginning of atrial depolarization to the beginning of ventricle depolarization, indicating the AV node function [52]. As an important component in ECG, the QRS complex contains Q, R and S waves, indicates ventricular depolarization (conduction). Moreover, the interval of the QRS complex can illustrate the conduction time of ventricles and determine the possible cause of conduction arrhythmias [53]. A normal interval of the QRS complex is in a range of 0.08 and 0.1 seconds while the interval of greater than 0.12 seconds is regarded as abnormal [53, 54].

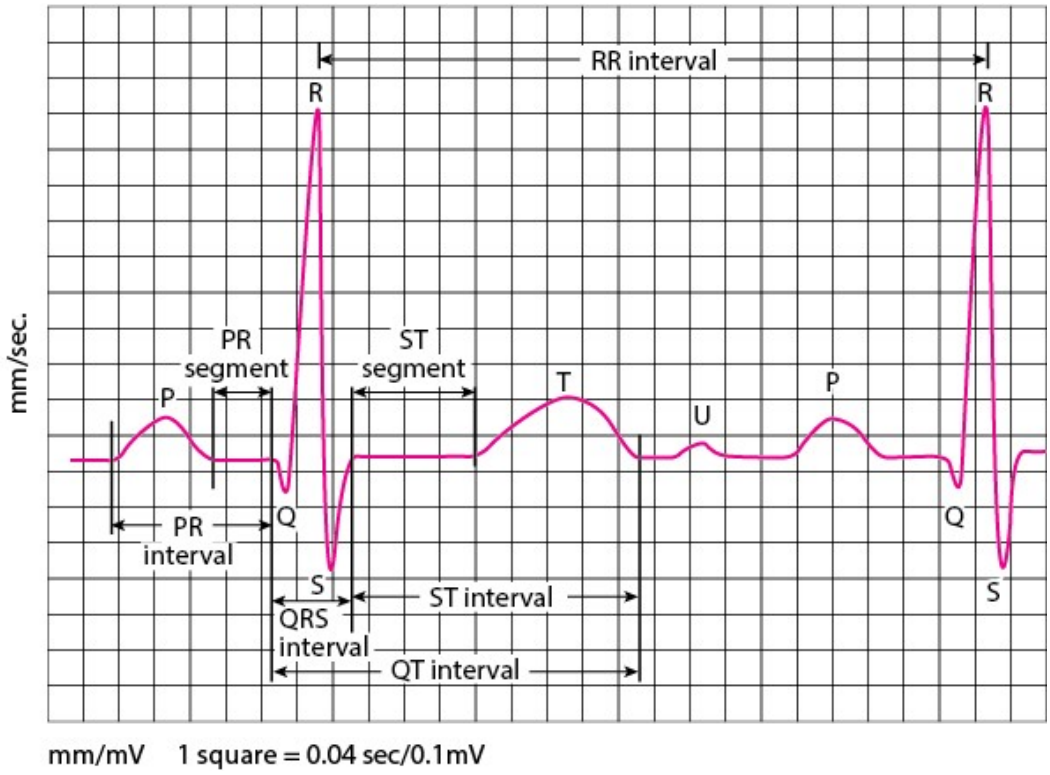

Figure 2.2: Typical ECG features of waveform: P-wave, P-R interval, P-R segment, QRS complex, S-T segment, S-T interval, Q-T interval, T-wave and RR interval. Image taken from[58]

T wave represents the ventricular repolarization and the Q-T interval represents the time from the starting of Q wave to the T wave end. Q-T interval is tightly linked to the heart rate, where the prolonged and abbreviated QT intervals are relative to the rising risk of ventricular arrhythmias [55]. U wave probably represents the after-depolarization of ventricles and the abnormal U wave may indicate ventricular arrhythmias. Besides the Q-T interval, the RR interval is another important clinical indicator, representing the time between two adjacent R waves. The Normal RR interval is between 0.6 and 1.2 seconds for resting human hearts [56]. To some extent, analysing the irregularity of RR intervals can reflect heart rate variability [57], further detecting underlying cardiac arrhythmias.

2.1.2 ECG acquisition

An Electrocardiogram is measured via electrodes placed on the human skin, detecting the electrical activities of the heart [59]. As shown in Figure 2.3, a standard ECG normally requires ten electrodes for providing the twelve-lead view where each lead represents a specific angle of orientation about the heart. The 12 leads can divide into limb leads (I,II,III,aVR,aVL,aVF) and precordial leads (V1,V2,V3,V4,V5,V6), comprehensively recording the electrical activity of the human heart.

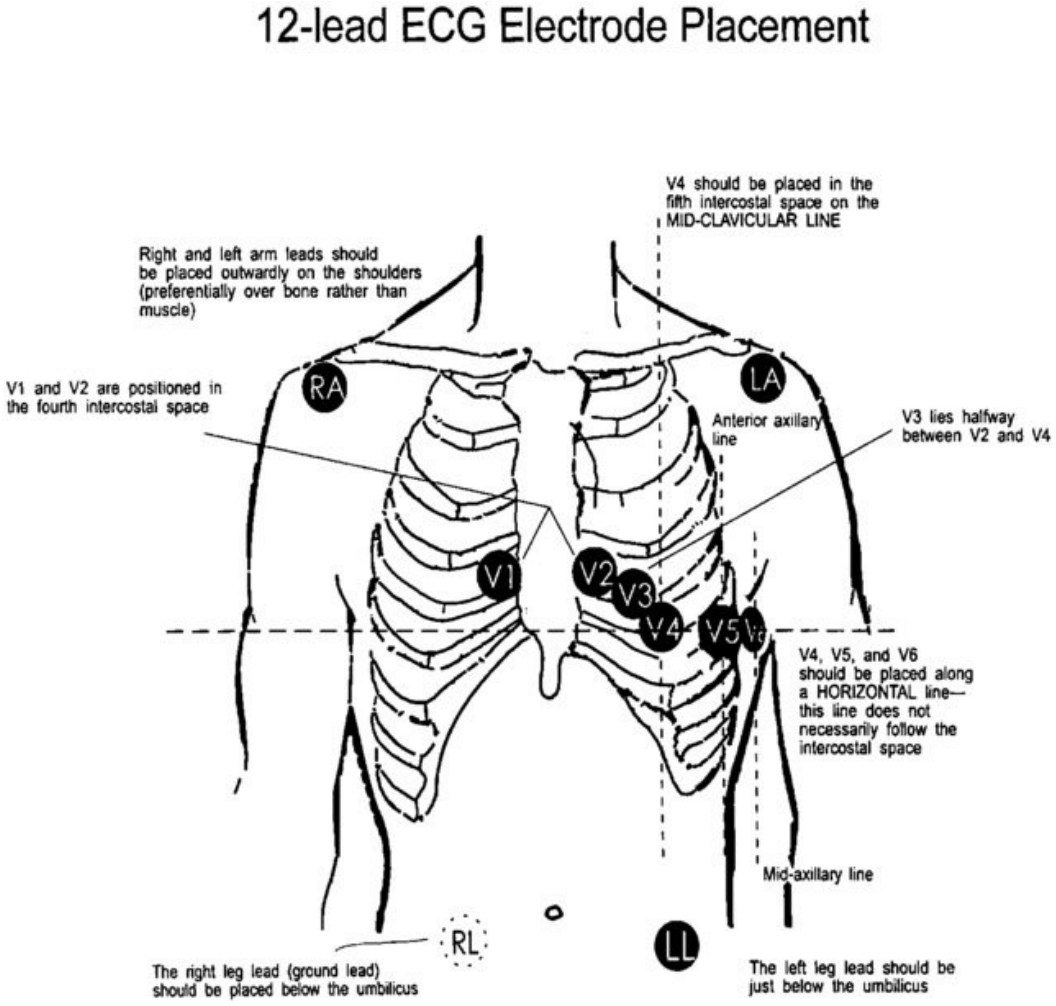

Figure 2.3: The illustration of 12-lead ECG electrode placement.Image taken from[60]

The limb leads can be further divided into Bipolar limb leads (I,II,III) and Augmented unipolar limb leads (aVR,aVL,aVF) [61]. All limb leads can be calculated by the potential differences between electrodes placed on the limb. The calculation can be

written as follows:

$$I = U_{LA} - U_{RA} \tag{2.1}$$

$$II = U_{LL} - U_{RA} \tag{2.2}$$

$$III = U_{LL} - U_{LA} \tag{2.3}$$

$$aVL = U_{LA} - \frac{U_{RA} + U_{LL}}{2} \tag{2.4}$$

$$aVR = U_{RA} - \frac{U_{LA} + U_{LL}}{2} \tag{2.5}$$

$$aVF = U_{LL} - \frac{U_{RA} + U_{LA}}{2} \tag{2.6}$$

where the U represents the function of electrodes potential. As a reference to the body surface potential, Wilson's Central Terminal [62] is the average electrodes potential of three limb leads:

$$U_W = \frac{U_{RA} + U_{LA} + U_{LL}}{3} \tag{2.7}$$

Then, the chest leads are the potential difference between the precordial electrodes and Wilson's Central Terminal:

$$V1 = U_{V1} - U_w \tag{2.8}$$

$$V2 = U_{V2} - U_w \tag{2.9}$$

$$V3 = U_{v3} - U_w \tag{2.10}$$

$$V4 = U_{v4} - U_w \tag{2.11}$$

$$V5 = U_{V5} - U_w \tag{2.12}$$

$$V6 = U_{V6} - U_w \tag{2.13}$$

2.2 Cardiac Arrhythmia

Cardiac arrhythmias refer to the abnormality and irregularity of the rhythm of the heart [63]. Severe arrhythmias may cause irregular heart contraction which can rising the risks of stroke or death. Several arrhythmias mentioned in thesis projects are listed in the following, and the visualization of ECG lead II waveforms for these arrhythmias can be found in Figure 2.4.

2.2.1 Fibrillation

Fibrillation refers to the rapid irregularity contractions of the heart, leading to the asynchronism between heartbeat and pulse. As the most common type of arrhythmia [64], atrial fibrillation (AFib) is caused by the chaotic transmission of the electrical impulse in the atria, then leading to an abnormal heart rate up 100 to 175 beats per minute [65]. As for diagnosis, ECG can reflect the abnormal electrical activity of atria where the P wave may be absent or be replaced by fibrillatory waves due to atrial fibrillation [66]. The increasing and abnormal heart rhythm lead to irregular RR intervals on ECG, which can be another ECG feature of atrial fibrillation.

2.2.2 Atrioventricular Block

Atrioventricular block (AVB) refers to the block of the electrical impulse conduction between atria and ventricles. The atrioventricular block may cause insufficient pumping of the heart and slower heartbeats [67]. According to the extent of electrical signal impairment, AVB can be divided into First-degree heart block (I-AVB), Second-degree heart block (II-AVB) and Third-degree heart block(III-AVB). I-AVB is common and usually characterized by prolonged PR interval (>0.2s) [68]. The loss of QRS complex and widened QRS complex may be the ECG characteristics for II-AVB and III-AVB respectively.

2.2.3 ST segment elevation/depression

As the name implies, ST segment elevation/depression (STD/STE) refer to the abnormalities of ST segment in ECG. ST segment is normally flat with a duration of approximately 0.08 seconds [69] and it represents an isoelectric period between ventricular depolarization and repolarization. The abnormal waveform morphologies of the ST segment correspond to various cardiovascular diseases. An elevated ST segment may indicate Left ventricular hypertrophy, Left bundle branch block, Left bundle branch block,etc [70]. ST depression may be caused by hypokalemia, subendocardial myocardial infarction, etc. [71, 72].

2.2.4 Premature atrial/Ventricular contractions

Premature atrial contractions (PACs) indicate extra heartbeats that start at the atria. When the electrical current prematurely triggers the contraction of the atria, the heart lacks sufficient blood to pump out, leading to a narrow QRS complex or skipped beat [73]. PAC can be characterized by an abnormal waveform of P wave in ECG. According to figure 2.4, the invert P wave with a short P-R interval may indicate the atria retrograde activation caused by PACs [74]. Moreover, the presence of the post-extrasystolic pauses can be another ECG characteristic for PAC diagnosis [75]. PAC may trigger the SA node reset, thus causing a longer RR interval between adjacent beats.

As a similar condition, premature ventricular contractions (PVCs) are extra beats that start in the ventricles and also illustrate irregular heartbeats in ECG. For distinguishing PACs and PVCs in ECG, the differences in morphology of QRS complex, preceding P wave and length of the pause following extra beats can be the proper characteristic [76, 77].

2.2.5 Left/Right bundle branch block

As the name suggests, bundle branch block indicates the blockage that occurs in the pathway which conducts the electrical impulse to the left or right ventricles. A simple approach to distinguish the bundle branch block in ECG is the prolonged QRS complex (>0.12s) [78]. For further distinguishing the Left bundle branch block (LBBB) and right bundle branch block (RBBB), the leads I, V1-V6 and aVL in ECG can provide more morphological characteristics. In these lateral leads (I, aVL, V5-V6), LBBB may show the broad monophasic R wave while in RBBB there are widened and deep S waves [78, 79].

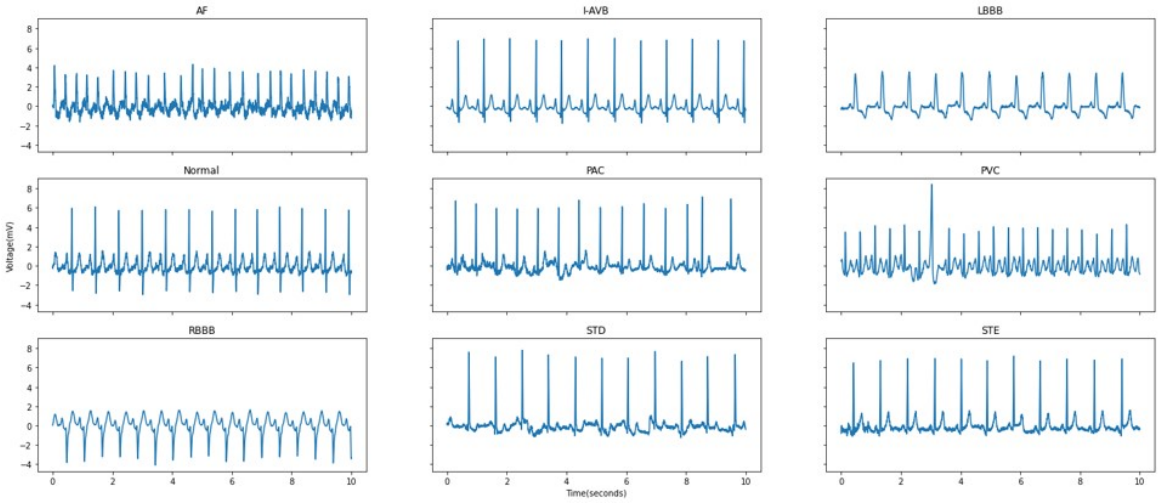

Figure 2.4: The visualization of ECG lead II waveforms for different types of arrhythmias, including Atrial Fibrillation(AF), Atrioventricular Block(I-AVB), Left bundle branch block (LBBB), Normal, Premature atrial contractions(PAC), Premature ventricular (PVC), Right bundle branch block(RBBB), ST segment depression(STD), ST segment elevation(STE).

2.3 Deep learning technology

As a subset of Machine learning (ML), Deep learning aims to train the computational models to perform human-like tasks [80]. Different from mathematical algorithms of machine learning, deep learning devotes to training the multi-layer artificial neural networks to learn autonomously for automatic feature extraction and pattern recognition [81, 82]. Artificial neural networks are modelled based on the structure of the human brain, which enables deep learning widely used in speech recognition, image classification, predictions etc [80, 83]. Deep learning has ability of pattern identification and classify various types of data is also crucial for dealing with a large amount of data. Moreover, deep learning approaches can be supervised, semi−supervised or unsupervised in terms of different learning tasks [82].

In the projects of this thesis, deep learning approaches adopt supervised learning in signal or image classification since the neural network extracts features from labelled data. The basic flow diagram of supervised deep learning for classification tasks is shown in Figure 2.5. Initially, raw data is divided into the training set, validation set and test set and the three sets are disjoint. The training set is utilized for model training, aiming to fit the weights and bias of connections between neurons in the neural networks [80]. The validation set can evaluate the model fit on the training set while tuning hyper-parameters of the model [84]. As novel data for the model, the test set can obtain the unbiased evaluation of completely trained models [85]. In classification tasks, evaluation metrics are crucial for evaluating the capability to distinguish different classes. Furthermore, the comprehensive evaluations estimate the generalization ability of the fit model on the unseen data and provide the basis for model optimization [82].

This section aim to explain the key algorithms and advanced models in the deep leaning domain. The algorithms and models involved in this section are closely associated with thesis projects.

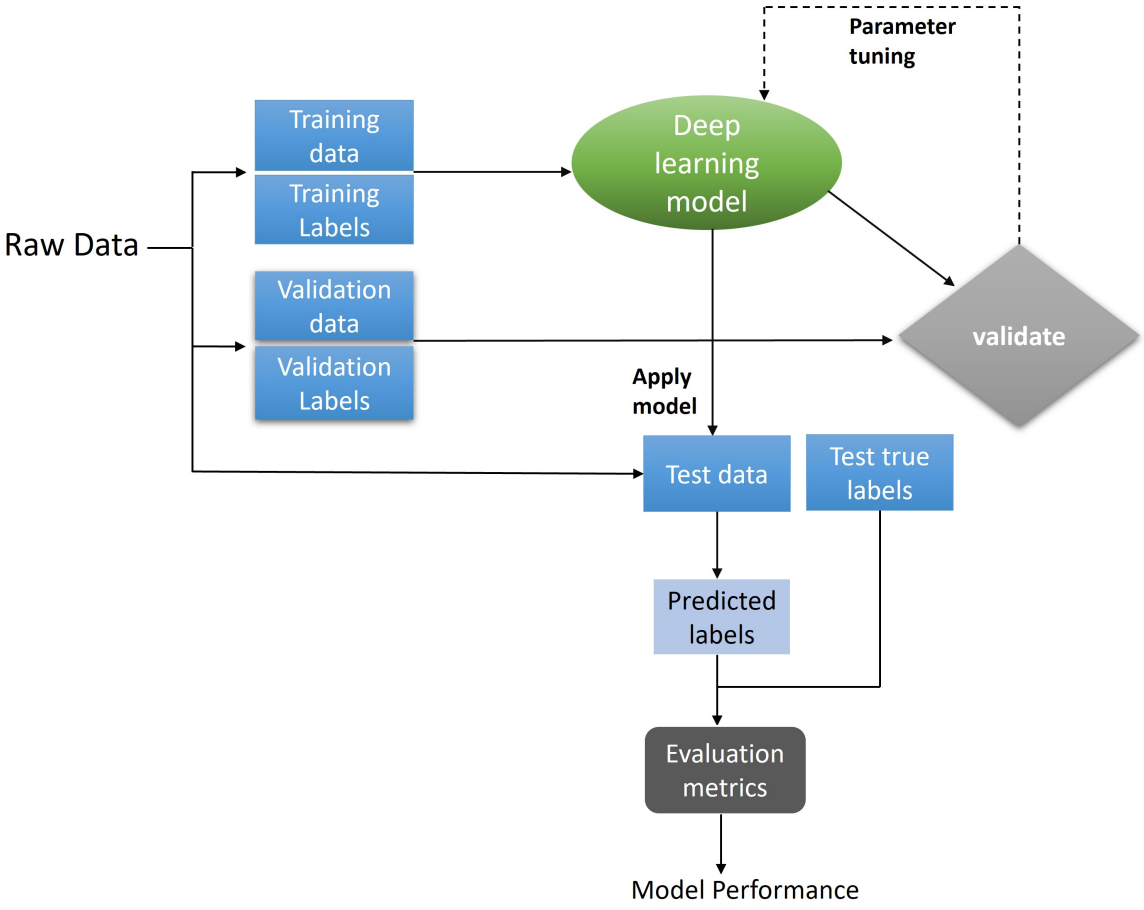

Figure 2.5: The flow chart of supervised algorithm in deep learning

2.3.1 Deep Neural Networks

The neural network (NN) is built to mimic the working pattern of cerebral neurons and it is the key to deep learning approaches [86]. A neural network contains interconnected neurons that are arranged in layers where each neuron involves an activation function for conducting feature extraction and mapping [86, 87]. The components of the neuron in the neural network are shown in Figure 2.6. The X represents input vectors and W represents the weights of the neuronal synapses. The neuron aims to calculate the weighted sum of input vectors and weights, then add the bias to the weighted sum and obtain the output via the activation function. For an artificial neuron, its output can be formulated as follows:

$$y = \varphi(\sum_{i=0}^{n} W_i X_i + b) \tag{2.14}$$

where the φ represents the activation function.

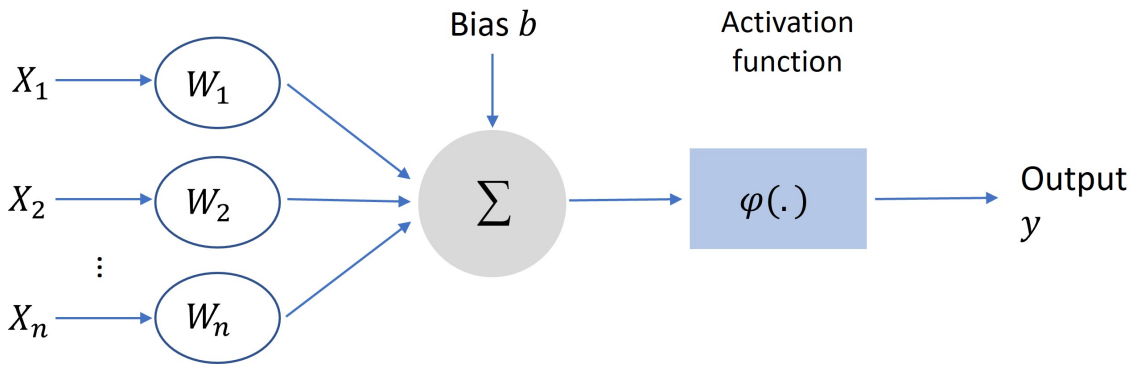

Figure 2.6: The structure of a neuron in neural network

Deep neural networks consist of multiple layers where each layer contains interconnected neurons [88]. Figure 2.7 illustrates a basic structure of a deep neural network which is also known as a Multilayer Feedforward Network (MFNN) or a Multilayer perceptron (MLP) [82, 89, 90]. A MFNN basically consists of three types of layers: input layer, hidden layer and output layer. Input data is initially fed into the input layer which ingests the data and transmits it to the next layer as input vectors [91]. Hidden layers are responsible for extracting features from input vectors via the transmission and weighted sum layer-by-layer [92]. Thus, the depth of the neural network depends on the number of hidden layers. As for the last layer of neurons, the output layer coalesces, analysis the previous information and produces the output vectors for different tasks [86].

In the training process, MFNN unidirectionally transmits the information from the input layer to the output layer and updates the parameters(i.e. weight,bias) via backpropagation [89]. The algorithms of activation function, gradient descent, feedforward and backpropagation are introduced below for further information on the learning process of MFNN.

Activation function

Aiming at processing the non-linear classification tasks, the activation functions should be nonlinear and continuously derivable. Several common activation functions are listed below:

As shown in figure 2.8, the sigmoid function is known as the logistic function that has a range of function values between 0 and 1 [93]. The mathematical formula of the

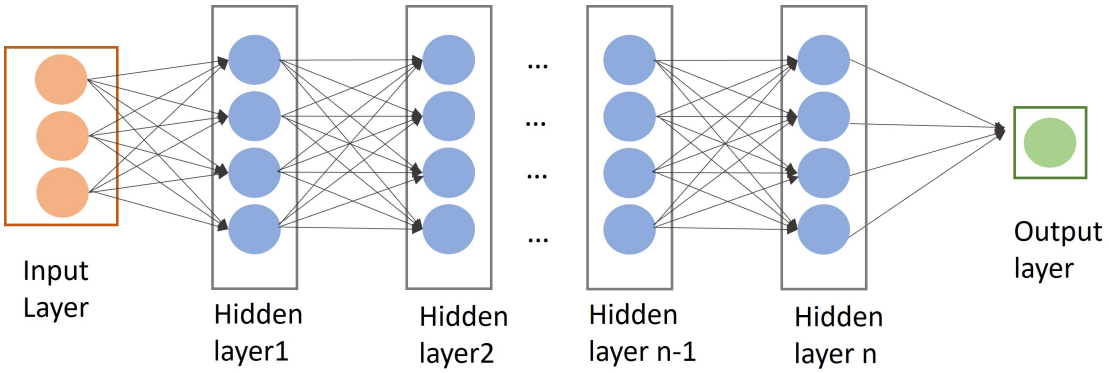

Figure 2.7: Deep neural network architecture

sigmoid function and its derived function can be written below:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2.15}$$

$$f(x)' = f(x)(1 - f(x)) \tag{2.16}$$

Considering the specific output range, the sigmoid function is commonly used for neural networks that aim to output the probability in binary-classification or multi-label classification tasks [94]. While in the deep neural network, the sigmoid function should be used with caution. According to the derived function, the derivative value is closed to 0 when the input value goes positive infinity or negative infinity. When multiple hidden layers adopt the sigmoid function, the multiply of small derivative values causes the gradient descend exponentially and gradient vanished.

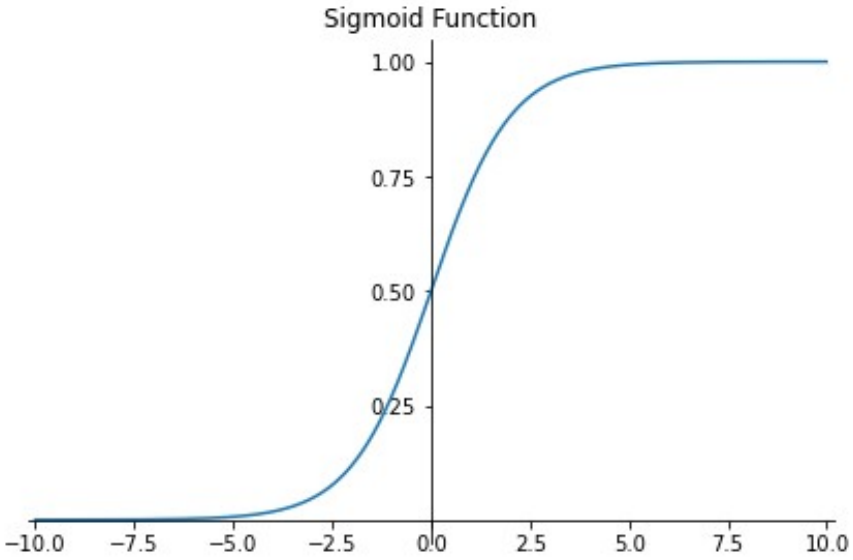

Figure 2.8: Sigmoid/Logistic activation function

In figure 2.9, the tanh function (Hyperbolic Tangent) has similar a S-shape as the Sigmoid function while the range of tanh function is between -1 to 1. The mathematical definition of the tanh function and its derived function are listed below:

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{2.17}$$

$$f(x)' = 1 - \left(\frac{e^x - e^{-x}}{e^x + e^{-x}}\right)^2 \tag{2.18}$$

Both the sigmoid function and the tanh function face the vanishing gradient problem while the mean of the tanh function is closer to 0, contributing to faster convergence [95].

ReLu function is known as rectified linear unit function. Figure 2.10 plots the ReLu function and mathematical formulations are listed below:

$$f(x) = \begin{cases} x, & \text{if } x > 0 \\ 0, & \text{if } x \leq 0 \end{cases} \tag{2.19}$$

$$f(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{if } x \leq 0 \end{cases} \tag{2.20}$$

Adoption of ReLu in the deep feedforward network can improve the learning performance [94]. ReLu activation function provides a simple calculation that is linear for

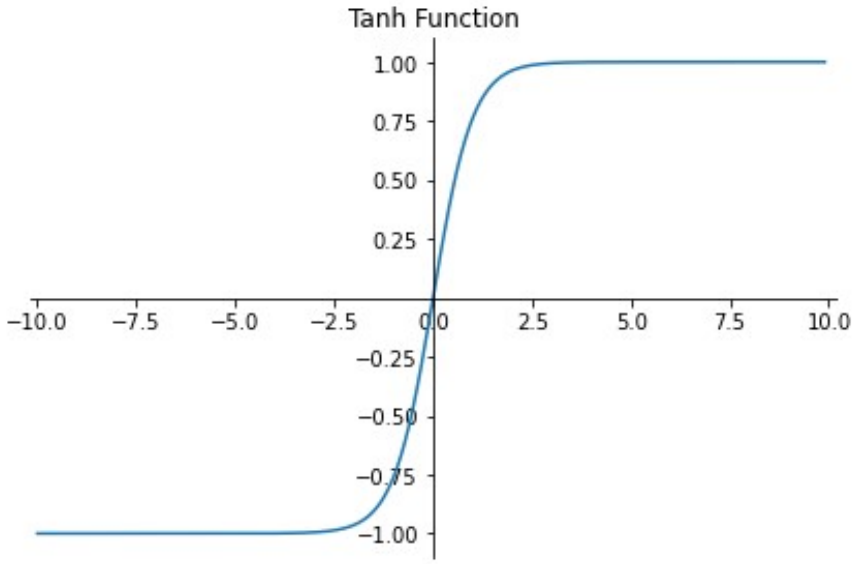

Figure 2.9: Tanh activation function

input values greater than 0 and returns 0 when values are not positive. It can selectively active neurons in hidden layers,which is known as sparse representation [96]. As a non-linear activation function, ReLu not only has advantages of linear activation function and its representational sparsity simplifies the computation during training process. However, the dying ReLu problem may occur and causes a slow training process of the network [97]. To prevent the dying ReLu problem, several extensions of ReLu (i.e. LReLu,PReLu,ELU) [97–99] allow the small negative values and reduce the number of neurons never been activated.

The softmax activation function can map the input vector to probabilities in the range 0 to 1 and the sum of probabilities equals 1 [100]. Different from the sigmoid function, the softmax function is normalized functions that are normally used in the output layer of deep neural networks and applicable for multi-class classification [94]. The mathematical definition of softmax function is formulated below:

$$f(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{N} e^{x_j}} \tag{2.21}$$

where the x_i represents the input vector with i dimensions and N is the number of classes in the multi-class classification.

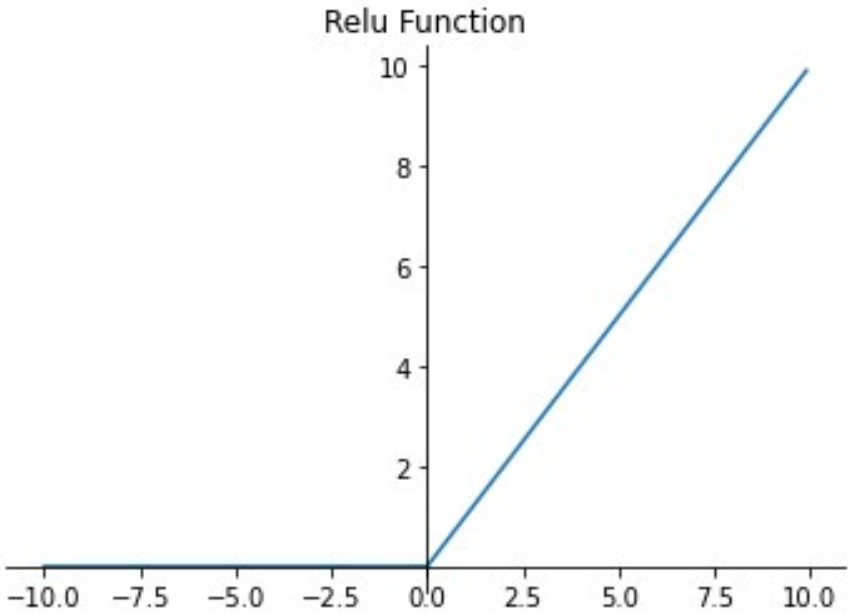

Figure 2.10: ReLu activation function

Loss function

Loss function is known as the cost function, aims to evaluate the error between the predicted value and true value for optimizing parameters of neural networks [101]. Thus, the training of the deep neural network is a procedure that minimizes the loss function based on gradient methods [94]. It is crucial for choosing the proper loss function for classification tasks as the function should faithfully represent the classification goals [94]. Four common loss functions for classification are explained below:

For binary classification tasks, binary cross-entropy is a proper choice [102, 103]. Binary classification problems aim to divide the input samples into two groups upon the features learned by the networks. Binary cross entropy loss (sigmoid cross-entropy loss) compares each of the predicted probabilities with true class output (0 or 1) and calculates a score to penalize the probabilities upon the distance from the true class value [104]. In mathematics, binary cross entropy loss can be formulated below:

$$p_i = sigmoid(z_i) = \frac{1}{1 + e^{-z_i}} \tag{2.22}$$

$$Loss = -y_i * log(p_i) + (1 - y_i) * log(1 - p_i) \tag{2.23}$$

where the y_i is the true class, p_i and $(1 - p_i)$ are the probability of class 0 and 1 respectively. Besides, binary cross-entropy loss can be used in multi-label classification either [105, 106]. In multi-label classification tasks, the true class label can be one-hot (i.e.[1,0,0,1]) which may have more than one positive class of M classes. Thus, a multi-label task can be split into M independent binary classification tasks, where each sigmoid activation function in the output layer independently predicts the probability for a sample belonging to each class of M classes.

Categorical cross-entropy is also known as softmax loss which combines the softmax activation function with cross-entropy loss [103]. Its mathematical equations are shown below:

$$p_i = softmax(z_i) = \frac{e^{z_i}}{\sum_{m=1}^{M} e^{z_m}} \tag{2.24}$$

$$Loss = \sum_{i=1}^{M} y_i log(p_i) \tag{2.25}$$

where the M represent the number of classes, y_i is true class and p_i is the predicted outputs. Categorical cross-entropy loss is usually used in multi-class classification tasks [103, 107]. In the multi-class classification, each sample can only belong to one of the M classes. The true class label y_i can be one-hot (i.e. [0,0,0,1]) with a positive class and the $M - 1$ negative classes.

Gradient descent

Gradient descent is an iterative optimization approach that aims to improve the deep learning model (neural network) by minimizing the loss function [108, 109]. As shown in figure 2.11, gradient descent starts with a random point (w_1) on the continuously differentiable function ($L(w)$) and moves in the negative direction of the gradient (derivative) of the function to reach the global or local minimum (w_n).
In math, gradient descent for minimizing the function can be formulated in the following. According to figure 2.11, the point needs to consistently move to the next position during multiple iterations until reaches the minimum:

$$w_{n+1} = w_n - \eta \nabla L(w_n) \tag{2.26}$$

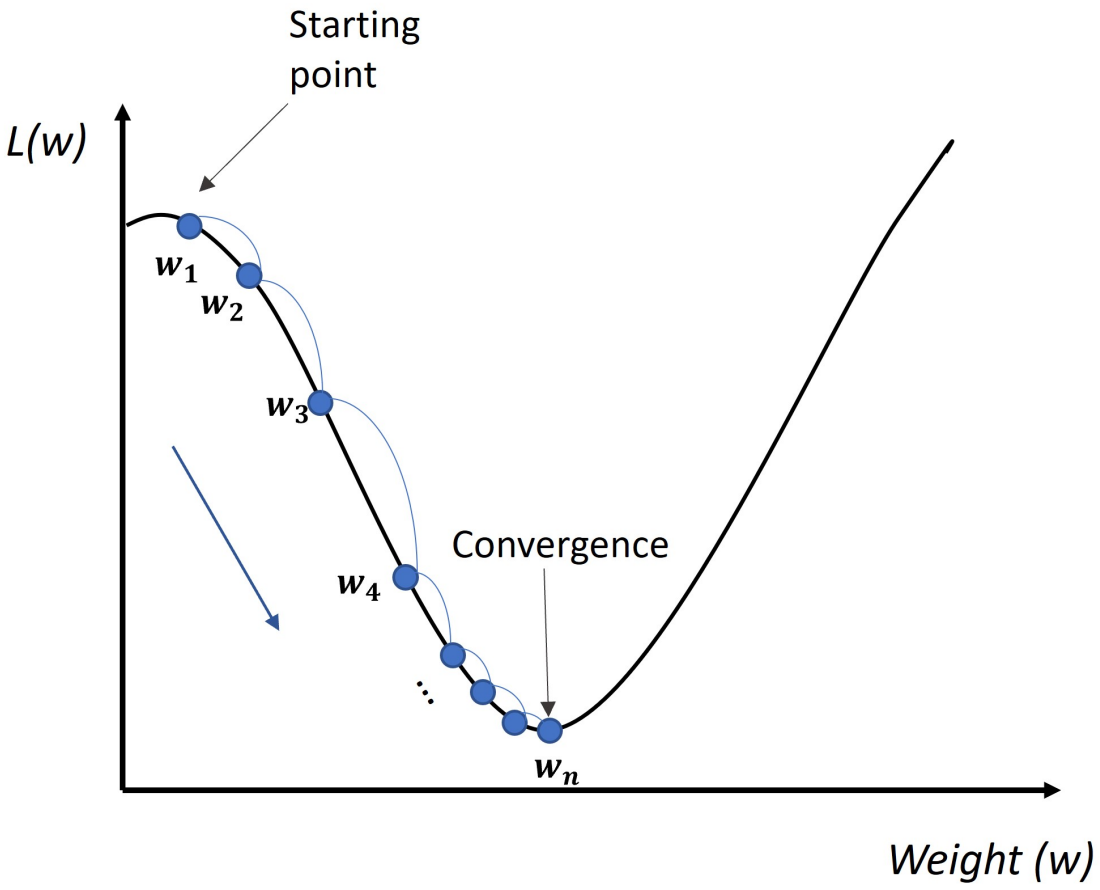

Figure 2.11: The illustration of gradient descent algorithm

where the η is a small positive number which is known as the learning rate. Thus, the process of weight update can be written as:

$$\begin{aligned} &\textit{Repeat until convergence}\,\{ \\ &\qquad w \leftarrow w - \eta \nabla L(w) \\ &\} \end{aligned} \tag{2.27}$$

and a monotonic sequence can be obtained:

$$L(w_0) \geq L(w_1) \geq L(w_2) \geq \cdots \tag{2.28}$$

Moreover, the setting of the learning rate is crucial in gradient descent since the learning rate determines the number of weights that are updated in each iteration [94, 110]. As shown in figure 2.12, the large learning rate can cause a larger step while increasing the risk of overshooting the minimum. Compared to the large learning rate, the small learning rate seems to be more conducive to convergence. Nevertheless, the very small learning rate may compromise the overall efficiency and permanently be stuck on a suboptimal result [94]. Normally, the range of values of the learning rate is between 10^{-6} and 1.0 and the default value of initial learning rate is 0.1 or 0.01 [111]. As an important hyper-parameter, the tuning of the learning rate is challenging and time-consuming [112, 113]. More information about configuring the learning rate can be found in the section on hyper-parameter tuning.

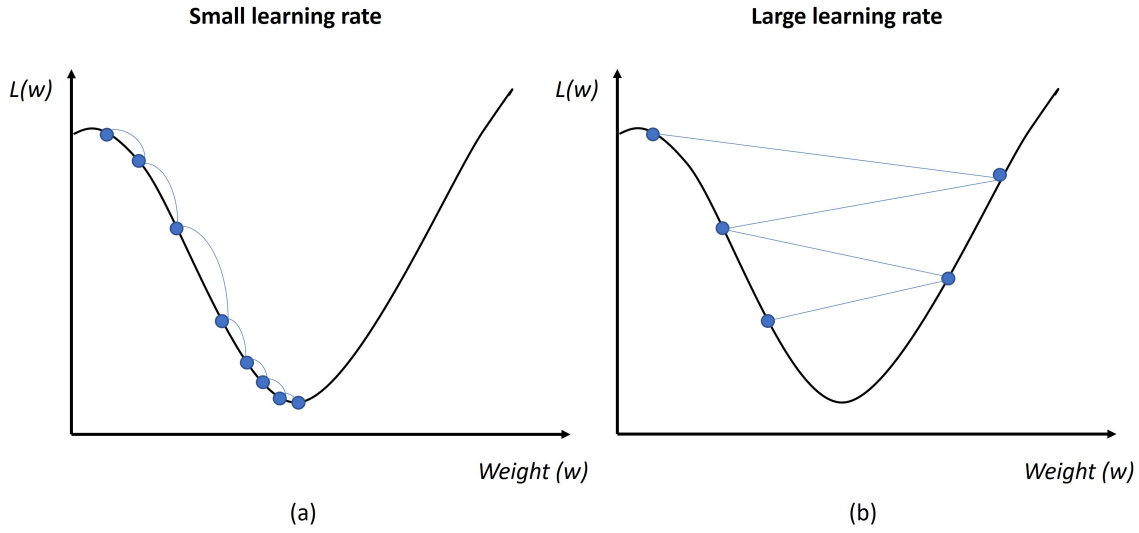

Figure 2.12: (a) Gradient descent with small learning rate (b) Gradient descent with large learning rate.

Backpropagation

When calculating the gradient of the loss function, backpropagation can help to calculate derivatives and update the weights layer by layer in a negative direction of feedforward pass [89]. Thus, backpropagation can be regarded as a concrete implementation algorithm of gradient descent during the training of the deep neural network. Figure 2.13 shows a brief structure of backpropagation performed in the deep neural network. In the process of the forward propagation, input samples are fed into the neural network for computing the intermediate variables (i.e. the output of neurons)

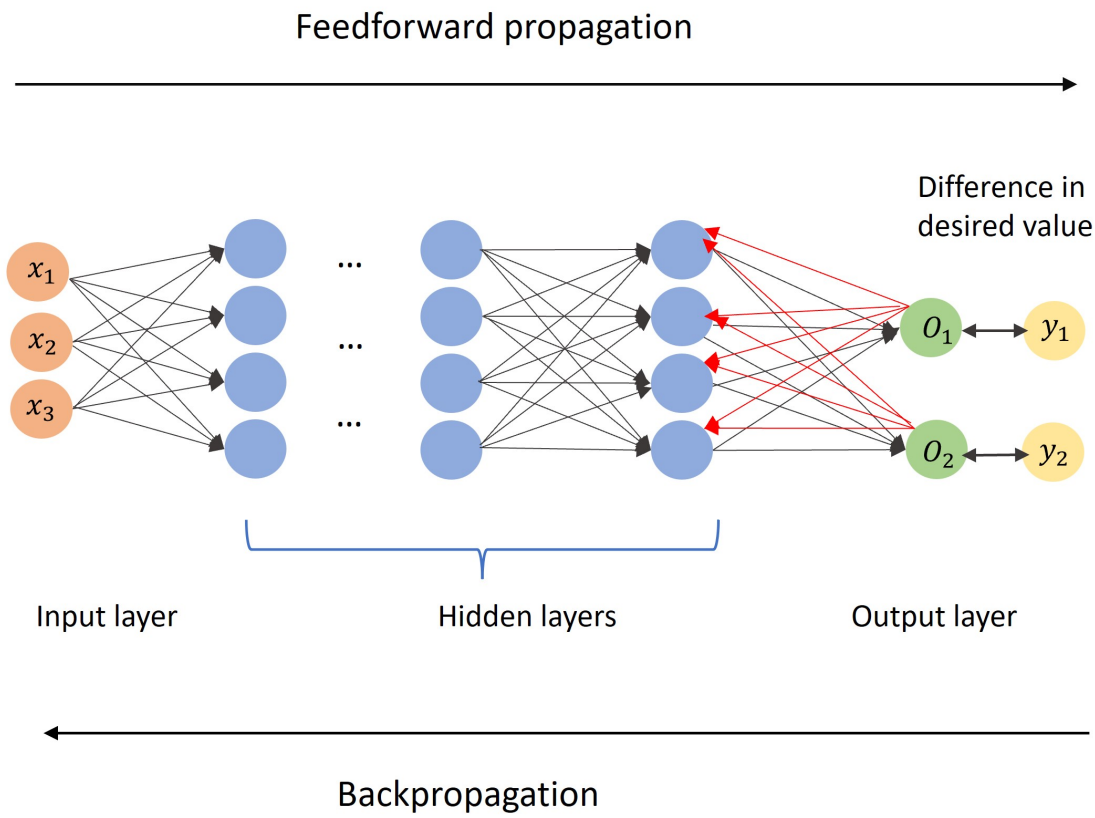

Figure 2.13: The working structure of backpropagation in the deep neural network

in sequence from the input layer to the output layer [114]. Then, the value of the loss function (error) can be computed via the differences between the desired value and the output of the network. Backpropagation aims to propagate the error in a reverse order and computes the gradients of the loss function relative to weight parameters in each layer [115]. Finally, the gradient descent uses the gradients to update the weights and further minimize the loss function.

Mathematically, gradients are equivalent to the set of the partial derivative of the loss function (error) concerning weights in the network [116]. The main deducing steps of backpropagation are listed below.

To explain the algorithm, figure 2.14(a) illustrates the computation of the input and output of a neuron with a sigmoid activation function, and then figure 2.14(b) shows a simple feedforward network with one hidden layer. As shown in figure 2.14(a), a

neuron can perform two operations:

$$\begin{aligned} In &= \alpha * a + \beta * b \\ Out &= Sigmoid(In) \end{aligned} \tag{2.29}$$

where the In operation refers to the weighted accumulative addition for the output of adjacent neurons in the previous layer and Out operation process a non-linear transform based on the sigmoid activation function. Thus, the input and output of neurons h_1 and h_2 can be computed as follows:

$$\begin{aligned} In_{h_1} &= w_1 * x_1 + w_3 * x_2 \\ h_1 = Out_{h_1} &= Sigmoid(In_{h_1}) \end{aligned} \tag{2.30}$$

$$\begin{aligned} In_{h_2} &= w_2 * x_1 + w_4 * x_2 \\ h_2 = Out_{h_2} &= Sigmoid(In_{h_2}) \end{aligned} \tag{2.31}$$

then, the neuron O_1 in the output layer can be represented as:

$$\begin{aligned} In_{O_1} &= w_5 * h_1 + w_7 * h_2 \\ O_1 = Out_{O_1} &= Sigmoid(In_{O_1}) \end{aligned} \tag{2.32}$$

In the same way, the In_{O_2} and O_2 can be obtained as:

$$\begin{aligned} In_{O_2} &= w_6 * h_1 + w_8 * h_2 \\ O_2 = Out_{O_2} &= Sigmoid(In_{O_2}) \end{aligned} \tag{2.33}$$

At the end of the forward propagation, the error can be calculated via the Mean square error (MSE) loss function and ground truth y_i:

$$Error = \frac{1}{2}\sum_{i=1}^{2}(O_i - y_i)^2 \tag{2.34}$$

The large error means that the parameters of the neural network (i.e. weight) should be further optimized until the error approaches zero. For passing backwards the error and weight optimization, backpropagation computes the gradients of the weights in the network. For example, the computation of the gradient of w_5 based on the chain rule is shown below:

$$\delta_5 = \frac{\partial Error}{\partial w_5} = \frac{\partial Error}{\partial O_1} * \frac{\partial O_1}{\partial In_{O_1}} * \frac{\partial In_{O_1}}{\partial w_5} \tag{2.35}$$

where,

$$\begin{aligned}
\frac{\partial Error}{\partial O_1} &= O_1 - y_1 \leftarrow Error = \frac{1}{2}\sum_{i-1}^{2}(O_i - y_i)^2 \\
\frac{\partial O_1}{\partial In_{O_1}} &= O_1 * (1 - O_1) \leftarrow O_1 = Sigmoid(In_{O_1}) \\
\frac{\partial In_{O_1}}{\partial w_5} &= h_1 \leftarrow In_{O_1} = w_5 * h_1 + w_7 * h_2
\end{aligned} \tag{2.36}$$

In the same way, the error can be continually passed backwards for computing the gradient of all weights in the neural network. For example, the backward path of gradient calculation for w_1 is plotted as red lines in figure 2.14(b). The gradient computation of w_1 can be formulated as:

$$\begin{aligned}
\delta_1 = \frac{\partial Error}{\partial w_1} &= \frac{\partial Error}{\partial O_1} * \frac{\partial O_1}{\partial w_1} + \frac{\partial Error}{\partial O_2} * \frac{\partial O_2}{\partial w_1} \\
&= (\delta_5 + \delta_6) * \frac{\partial h_1}{\partial In_{h_1}} * \frac{\partial In_{h_1}}{\partial w_1} \\
&= (\delta_5 + \delta_6) * \frac{\partial h_1}{\partial w_1}
\end{aligned} \tag{2.37}$$

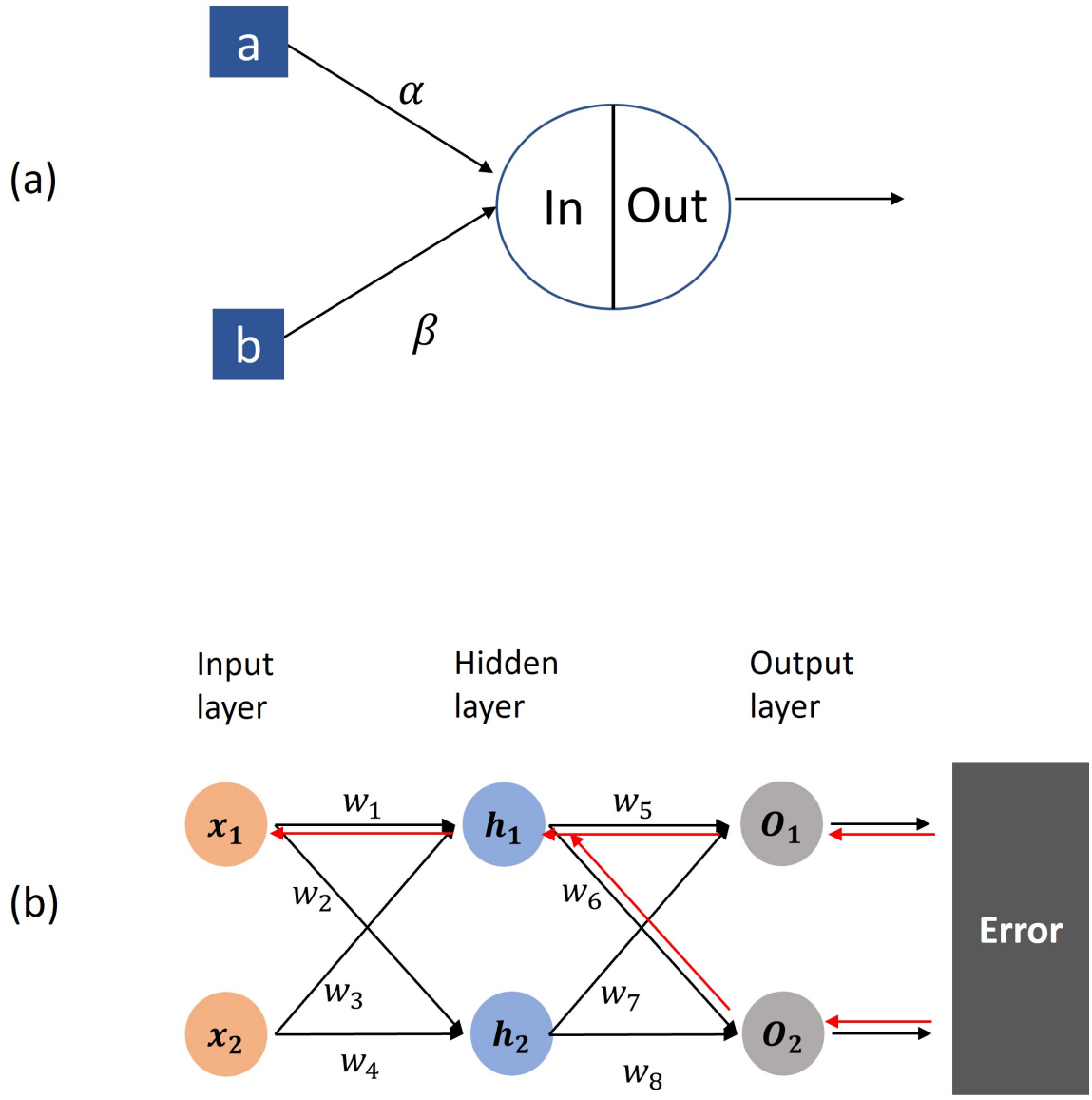

Figure 2.14: (a):'In' and 'Out' operation in a neuron of feedforward neural network (b): Computational graph of a simple feedforward neural network. Assuming the activation function used in hidden layer and output layer is sigmoid function.

2.3.2 Hyperparameter Tuning

Hyperparameters, as a group of pre-set parameters, have a crucial impact on the performance of deep neural networks. Different from the model parameters (i.e. weights), hyperparameters are usually constant and cannot be optimized during training process. To properly tune the hyperparameters, the observations for loss function and validation or test accuracy are crucial. In this section, the typical hyperparameters that are closely associated with the model performance are listed in the following.

Batch size

According to the previous section, gradient descent helps to train the neural network where the estimate of the loss function for updating weights is computed via subsets of the training dataset. The number of samples divided from the training dataset utilized in the estimate of the loss function is regarded as the batch size which is a crucial hyperparameter for the training efficiency of the neural network [117].

There are three types of gradient descent algorithms depending on different batch sizes: batch gradient descent (batch size = size of training dataset), stochastic gradient descent (batch size = 1) and mini-batch gradient descent (1<batch size<size of training dataset) [118]. A proper batch size can improve the convergence rate of the network and adjust the memory utilization [119]. Small values of the batch size contribute to quick convergence and less memory usage, whereas it introduces noise during the training process. The large batch size causes a slow convergence, while the estimation of the gradient of the loss function is more accurate. For a large amount of training dataset and deep neural network, the mini-batch gradient descent may be a proper choice where the batch size of 32, 64, 128 are commonly used [111, 118].

Epoch

Epoch, as an important hyperparameter, defines the number of times that the neural network will be trained on the entire training dataset [94]. One epoch means that all samples in the training dataset will participate in the training process for updating weights [120]. In terms of the different gradient descent algorithms, one epoch consists of one or multiple batches. The number of epochs usually depends on the condition of loss function and allows the neural network to keep training until the loss function has been minimized and levelled off [121].

Learning rate

According to the previous section of the gradient descent algorithm, the learning rate is a key hyperparameter which controls the number of weights that can be updated in each iteration via the estimated loss function. When the learning rate is overlarge, the

network may skip the minimum of the loss function thus interfering with parameters updating [94]. Whereas the very small learning rate may result in a slow convergence and low efficiency of the training process [94].

Configuration of the learning rate is relevant to the different amounts of training data, batch sizes and optimizations [122]. Normally, the value of the learning rate has a positive relation with the batch size due to the noisy estimate of the error gradients [94]. Nevertheless, the proper learning rate cannot be configured only depending on experience and default settings. Besides, a constant learning rate may be inapplicable to all parameters updating when the training dataset is sparse. For automatically adjusting the learning rate, several optimization algorithms (i.e. Adagrad, RMS Prop, adam) provide build-in schemes for adaptively adjusting the learning rate without manual operation [111, 123]. The learning rate schemes and relevant optimizers are detailed in the section on optimizers.

Optimizers

Optimizers are algorithms or functions that can be utilized to modify the attributes (i.e. weights, learning rate) of the neural networks for minimizing the loss function (error) [124]. Most current optimizers are derivatives of the algorithms of gradient descent, aiming to accelerate the process of gradient descent for reaching the global minimum of the loss function [124, 125]. Explanation and comparison between common-used optimizers are listed in the following.

As a common variant of gradient descent, stochastic gradient descent (SGD) optimizer [126] updates model parameters (weights) one by one via random selections. The process of weights updating can be formulated as:

$$w \leftarrow w - \eta \nabla L(w; x(i); y(i)) \tag{2.38}$$

where the $x(i), y(i)$ represent the training samples. Compared with the original gradient descent optimizer, the SGD optimizer contributes to more frequent updates of parameters and accelerates the convergence of the loss function. Nevertheless, the frequent updates may cause noisy gradients and high variances between parameters,

leading to the instability of convergence [127].

As the name suggests, the mini-batch gradient descent optimizer updates parameters batch by batch and each batch is a subset randomly selected from the training dataset [109]. The process of weights updating is written as:

$$w \leftarrow w - \eta \nabla L(w; B(i)) \tag{2.39}$$

where $B(i)$ represents the batches split from the training dataset. Mini-batch gradient descent optimizer has less frequent updates than that of SGD optimizer, obtaining more stable convergence [127]. To achieve a good convergence, an appropriate batch size should be tuned prudently.

To smoothen the convergence and lessen the high variance in SGD, momentum introduce a hyperparameter γ to add a momentum term in SGD:

$$\begin{aligned} V(t) &= \gamma V(t-1) + \eta \nabla L(w; x(i); y(i)) \\ w &\leftarrow w - V(t) \end{aligned} \tag{2.40}$$

Momentum algorithm accumulate the past gradients via the update equation and continuous move towards their directions. It softens the convergence towards the relevant direction of gradient descent and weakens the fluctuation to an irrelevant direction [128]. The extra hyperparameter γ determines the amount of past gradients to add in the update equation and requires to fine-tune for better convergence.

Previous optimizers are variants of gradient descent and require a constant value of the learning rate, which may be unfit for all weights updating. Besides, the appropriate configure of the learning rate is still a challenge since it strongly associated with the convergence.

Adaptive Gradient Descent (AdaGrad) optimizer [129] allows using different learning rates in every iteration and the learning rate changes with the differences between parameters during the training process. Mathematically, the AdaGrad optimizer sets the corresponding learning rates η for each parameter at every time step t based on

the computation of the previous gradients:

$$\begin{aligned} g_t &= \nabla L(w_t) \\ S_t &= S_{t-1} + g_t^2 \\ w_t &\leftarrow w_{t-1} - \frac{\eta}{\sqrt{S_t + \epsilon}} \cdot g_t \end{aligned} \tag{2.41}$$

where the S_t is the sum of the squares of gradients g_t from beginning to time step t and ϵ is a small positive number to avoid divisibility by zero. Thus, the AdaGrad optimizer tunes the learning rate adaptively for each network parameter without hand-tuning, which is more flexible and efficient than the optimizers based on gradient descent. Nevertheless, using the AdaGrad optimizer can cause a continued decline in the learning rate and further result in a slow speed of convergence [125].

As a special version of AdaGrad, RMS Prop (Root Mean Square) optimizer [130] is invented for addressing the rapid decline of the learning rate. Mathematically, RMS Prop introduces a hyperparameter γ for utilizing the moving average of squared gradients instead of accumulating all squared gradients in AdaGrad:

$$\begin{aligned} g_t &= \nabla L(w_t) \\ S_t &= \gamma S_{t-1} + (1-\gamma) g_t^2 \\ w_t &\leftarrow w_{t-1} - \frac{\eta}{\sqrt{S_t + \epsilon}} \cdot g_t \end{aligned} \tag{2.42}$$

where the γ is a hyperparameter whose default value is 0.9. RMS Prop can also automatically adjust the learning rate for each parameter and the learning rate will not decay as the training time grows [125].

Adam (Adaptive Moment Estimation) optimizer [131], as one of the most popular optimizers for training deep neural networks, combining PMS Prop with momentum. As shown in the following formulations, Adam uses the momentum term and stores the decaying average of the past gradients V_t. It also retains the moving average of

squared gradients used S_t in RMS Prop.

$$
\begin{aligned}
V_t &= \gamma_1 V_{t-1} + (1-\gamma_1) g_t \\
S_t &= \gamma_2 S_{t-1} + (1-\gamma_2) g_t^2 \\
\hat{V}_t &= \frac{V_t}{1-\gamma_1^t} \quad and \quad \hat{S}_t = \frac{S_t}{1-\gamma_2^t} \\
w_t &\leftarrow w_{t-1} - \frac{\eta}{\sqrt{\hat{S}_t + \epsilon}} \cdot \hat{V}_t
\end{aligned}
\tag{2.43}
$$

where the default value of γ_1 and γ_2 are 0.9 and 0.999 respectively. Thus, Adam optimizer combines both of the advantages of momentum and RMS Prop for adjusting adaptively the learning rate and rectifying the decayed learning rate. Adam optimizer demonstrates its superiority and promising in the speed of convergence on papers [125, 127, 132].

Layer weight regularizers

In the training process, the weights of the network can be learned via the training data and gradient descent. As the training time increases, the weights become more specialized on the training data, which leads to over-fitting [133]. Then, the weights will grow to handle the specifics, whereas the large weights cause unstable neural networks [94].

Weight regularization is a general method to keep the weights small for reducing the risk of over-fitting and improving the generalization of the neural network [134]. As a concrete regularization technique, layer weight regularizers can apply per-layer penalties on layer parameters (i.e. weights, biases) during the training process [135]. Two main regularizers are L1 regularizer and L2 regularizer.

Both L1 and L2 regularizers aim to add a regularization term about the layer parameters in the loss function, thus applying penalties on large parameters [135]. The regularization term of the L1 regularizer is the sum of the absolute value of weights and the loss function with the L1 regularization term can be expressed as:

$$
Loss_{L_1} = Loss_{ini} + \lambda \sum |w| \tag{2.44}
$$

where the default value of λ is 0.01. Thus, the process of gradient descent and weight update can be formulated as:

$$\begin{aligned} \delta_{L_1} &= \frac{\partial Loss_{L_1}}{\partial w} = \frac{\partial Loss_{ini}}{\partial w} \pm \lambda \\ w &= w - \eta * \delta_{L_1} \\ &= w - \eta * (\frac{\partial Loss_{ini}}{\partial w} \pm \lambda) \\ &= w - \eta * (\delta_{ini} \pm \lambda) \end{aligned} \tag{2.45}$$

According to formulations, the weight becomes larger when the initial weight is negative and becomes smaller when the initial weight is positive. Thus, L1 regularizer encourages weights to 0 for obtaining a more sparse weights distribution. Simultaneously, L1 regularization assigns partial features with weights of 0, reducing the complexity of the neural network [136].

As for L2 regularizer, the regulation term added in the loss function is the sum of the squared value of weights:

$$Loss_{L_2} = Loss_{ini} + \lambda \sum w^2 \tag{2.46}$$

where the default value of λ is 0.01. Similarly, the process of gradient descent and weights update is written as:

$$\begin{aligned} \delta_{L_2} &= \frac{\partial Loss_{L_2}}{\partial w} = \frac{\partial Loss_{ini}}{\partial w} + 2\lambda w \\ w &= w - \eta * \delta_{L_2} \\ &= w - \eta * \frac{\partial Loss_{ini}}{\partial w} - 2\lambda\eta * w \\ &= (1 - 2\lambda\eta)w - \eta * \delta_{ini} \end{aligned} \tag{2.47}$$

Compared with the L1 regularizer, the L2 regularizer aims to smoothen the weights instead of making weights sparse. Smooth weights distribution is more conducive to the stability of neural networks and prevents over-fitting [137].

Dropout rate

As a common regularization approach, dropout refers to dropping the proportion of neurons in given layers during training for encouraging the sparse representation in

the deep neural network [138]. According to these studies [139, 140], the deep neural networks with dropout can reduce the risk of over-fitting and improve the generalization performance of the neural network. Dropout rate is a hyperparameter that can be used per-layer in the neural network ad determines the probability of the ignored neurons in the given layer. A common range value of dropout rate in hidden layers is from 0.5 to 0.8, and the value close to 1.0 in the input layer for retaining input information. [138, 141].

2.3.3 Convolutional Neural Network

As deep learning develops, the traditional feedforward neural network (i.e.MLP) is not suitable for processing high-dimensional perceptual samples such as medical images [142, 143]. When high-resolution images are passed to the network, the billions of parameters and high GPU-demanding will be required for MLP to fit the high-dimension data. Convolutional neural networks (CNNs) [144] are equipped with locality and translational invariance and locality, creating the feature representations for small region of the input and improving its robustness to variations in feature positions. Instead of the fully-connected pattern in MLP, CNNs conduct a sparse local connectivity pattern that allows each neuron to connect with a small regions of neurons (receptive field) of the previous layer [144].

As a variant of MLP, CNN follows the basic structure of MLP that consists of the input layer, hidden layers and the output layer. As shown in Figure 2.15, hidden layers in a convolutional neural network mainly compose of three types of layers: convolutional layers, pooling layers and fully-connected layers.

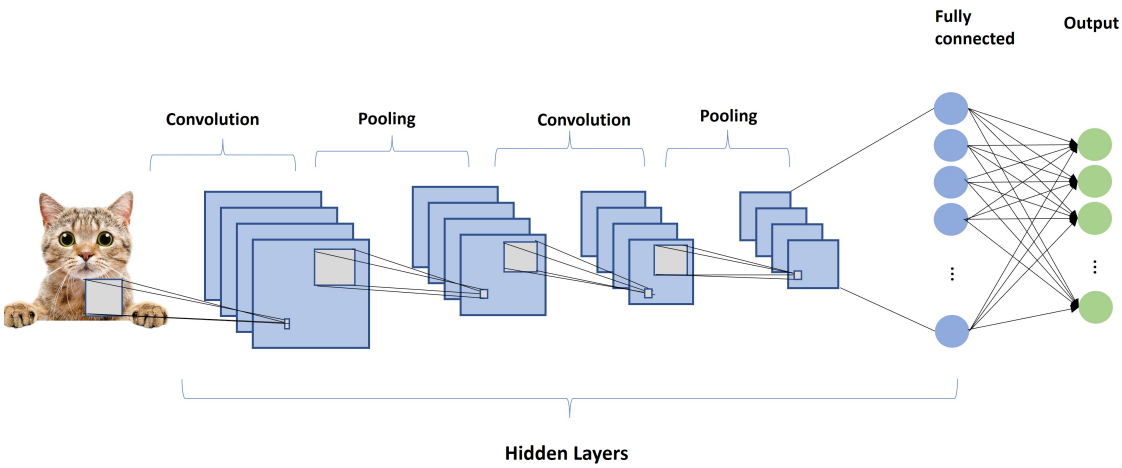

Figure 2.15: Basic structure of convolutional neural network

Convolutional layer

The convolutional layer (Conv) is the core component of the CNN that conducts the feature extraction via convolution operation [38]. Convolution operation is a class of linear operation that conducts an element-wise product between a set of layer parameters (kernels or filters) and receptive fields [145]. The kernel size ([width,height]) is smaller than that of the input while extending throughout the full depth of the input. Since the depth of kernels can extend up to all channels of the input, convolutional layers are skilled in processing the high-dimensional inputs that have multiple channels such as RGB images. During the forward propagation, the kernel slides across the width and height of the input, and then the element-wise product between the kernel and corresponding receptive field is computed, summing to obtain a two-dimensional feature map of that kernel [146] (Figure 2.16). This operation is repeated by applying a certain number of kernels to form their corresponding feature maps. Stacking the corresponding features maps for all kernels along the depth dimension produces the complete output volume of the convolution layer [145].

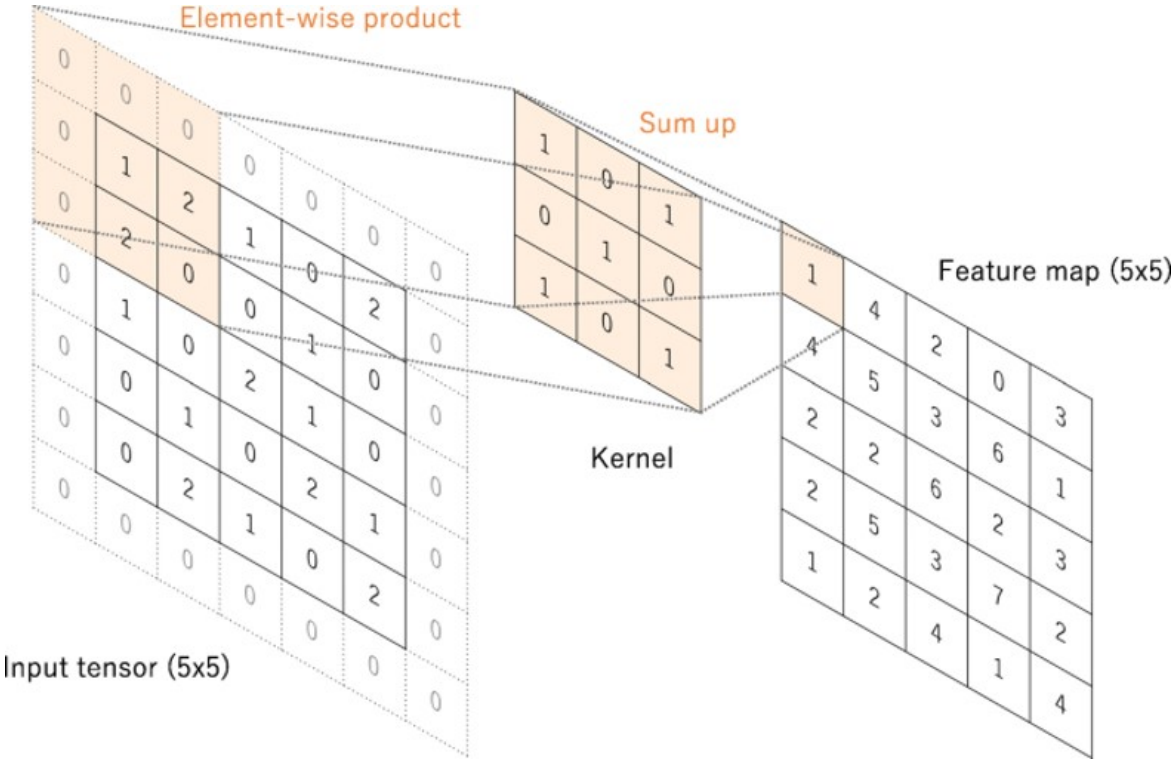

Figure 2.16: Basic structure of convolutional neural network [146]

Besides the size of the kernel, stride, padding and kernel numbers are three hyperparameters that can determine the size of the output volume obtained by a convolutional layer [38]. The computation of the spatial size of the output volume can be formulated below:

$$\begin{aligned} Output_{width} &= \frac{(W_{input} - K_{width} + 2 * P)}{S} + 1 \\ Output_{height} &= \frac{(H_{input} - K_{height} + 2 * P)}{S} + 1 \end{aligned} \tag{2.48}$$

where the W_{input} and H_{inputs} represent the width and height of the input volume respectively. The stride S denotes the sliding size of the kernel and padding P is known as zero padding, adding zeros around the border of the input to avoid the loss of the edge information during convolution operation. The number of kernels determines the depth of the output volume where each kernel refers to a feature extractor for representing a specific characteristic of the input.

In CNNs, kernels refer to the vector of weights and bias that are learnable parameters of the neural networks. Each kernel in the convolutional layer slides across the entire input field for forming one feature map, which means that all neurons in a feature map share the same parameters (weight, bias), called parameter sharing [147].

Parameter sharing scheme is crucial for convolution operation. It not only helps to control the number of parameters of the CNNs but is also combined with the max pooling operation, providing the CNNs with translation invariance [148].

Pooling layer

Pooling layers [149] are usually added behind convolutional layers, reducing the spatial size of the feature maps obtained by previous convolutional layers. Pooling operations work as downsampling, reducing the computational complexity of CNNs and extracting dominant features equipped with translation invariance. Figure 2.17 shows two types of pooling operations: max pooling operation and average pooling operation. Max pooling [150, 151] is the most popular pooling operation, extracting the maximum value from the region of the feature map corresponding to the kernel. Average pooling [149] extracts the mean of all the values from the region of the feature map covered by the kernel. The depth of features maps remains constant during both two pooling operations.

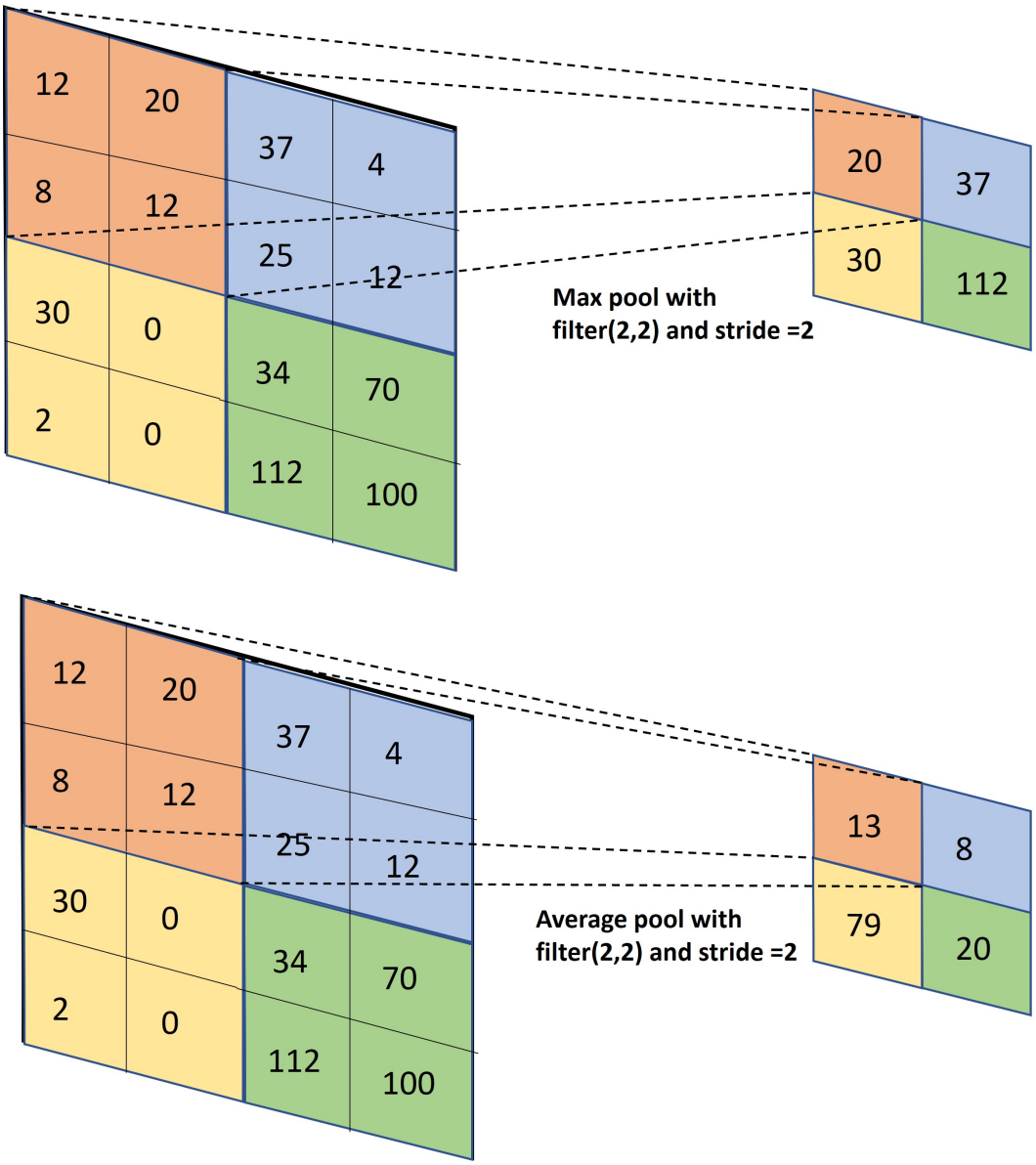

Figure 2.17: Two types of pooling operations.

Similar to convolutional layers, kernel size, stride and padding are three hyper-parameters that determine the spatial size of the output volume. The computation process is written as below:

$$Output_{shape} = \frac{W_{shape} - K_{size} + 2 * P}{S} + 1 \tag{2.49}$$

As the most common pooling layer, the max-pooling layer extracts the most prominent region of the feature map and discards redundancy information, contributing to dimensionality simultaneously reduction and noise reduction. Nevertheless, max pooling layers may drop useful information when the background of the input images

is complex. Average pooling layers aim to smooth out the feature map, which may weaken the characteristics of the input data.

Fully-connected layer

As the name suggests, all neurons in the fully-connected layer entirely connect with all neurons in adjacent layers [38]. In CNNs, the feature maps produced by the last convolutional layer or pooling layer are flattened as one-dimensional vectors, then passed into one or more fully-connected layers to form the final output [152]. Fully-connected layers contribute to mapping the feature maps to the final output of CNNs. To accomplish the classification tasks, the last fully connected layer in CNNs is followed by an activation function (i.e. softmax, sigmoid), predicting the probabilities for each class.

2.3.4 Recurrent Neural Network

Recurrent neural networks (RNNs) [153] are a special class of neural networks that are commonly adapted to process the time series data or sequential data such as speech signals [154] and language text [155]. Traditional feedforward networks only allow the samples to pass forward from input to output where samples are independent of one another. To process the sequential data, RNNs contain an internal memory to store the information of previous inputs that combine with the current inputs for generating the next output of the sequence [156].

Figure 2.18 shows a basic structure of an RNN. Different from the original feedforward network(figure 2.7), RNN utilizes one or more feedback loops in hidden layers, feeding the information stored by the internal memory back into the network. The feedback loop can be unrolled in t time steps to obtain an RNN shown in Figure 2.19.

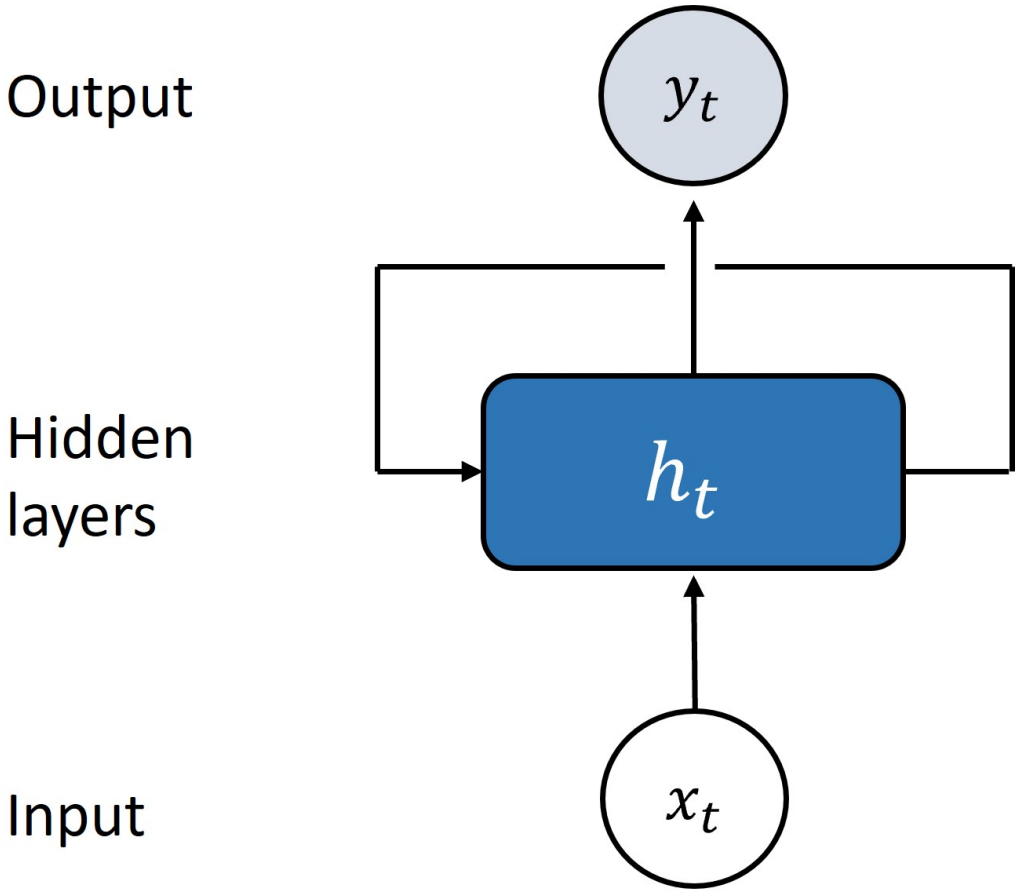

Figure 2.18: Basic structure of the recurrent neural network

Suppose there is an input time $\{x_0, x_1, x_2, ..., x_t\}$, where $x_t \in R^n$, k represents the time steps and n refers to the number of neurons in the input layer. The corresponding hidden states are $\{h_0, h_1, h_2, ..., h_t\}$, where $h_t \in R^m$ and m represent the number of neurons in hidden layers. During the feedforward pass of the RNN, the hidden state at time step t can be formulated as:

$$\begin{aligned} h_t &= f_H(x_t, h_{t-1}, w_{hh}, w_{xh}) \\ h_t &= f_H(w_{xh}x_t + w_{hh}h_{t-1}) \end{aligned} \tag{2.50}$$

where the $f_H()$ refers to the activation function used in hidden layers. The w_{hh} and $w_x h$ represent the weight of the neurons in the hidden layers and input layer respectively. The corresponding outputs are $\{y_0, y_1, y_2, ..., y_t\}$ and y_t can be formulated as:

$$y_t = f_o(w_{hy}h_t) \tag{2.51}$$

where the f_o represents the activation function in the output layer and w_{hy} represents the weights associated with hidden to output neurons. The network parameters (i.e. weights) are shared temporally during the forward propagation of RNN, which means that x_{hh}, w_{hy} and w_{xh} remain unchanged at each time step. Hence, the value of current output y_t not only depends upon the current input x_t but also on the value of hidden neuron h_{t-1} at the previous time step.

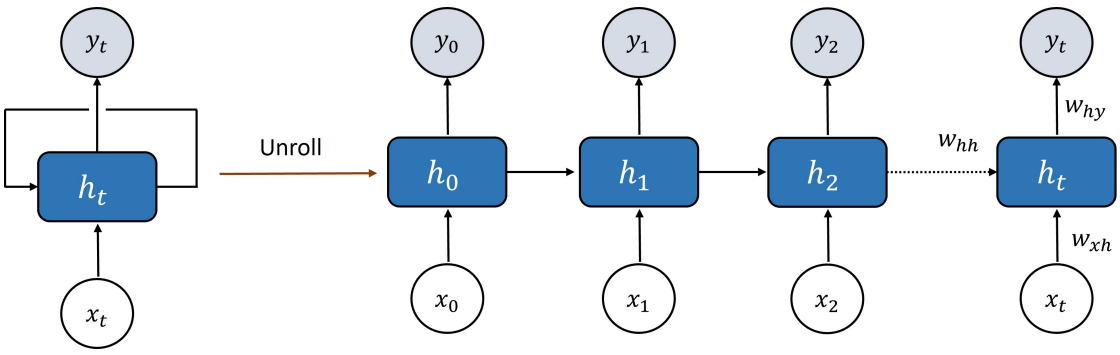

Figure 2.19: Unfold structure of the recurrent neural network

Original RNNs may have gradient problems and difficulty in processing long range dependencies [157]. During the training of RNNs, the network back propagates the

loss function through time (BPTT) [158] to compute the gradient:

$$\frac{\partial E}{\partial W} = \sum_{t=1}^{T} \frac{\partial Error_t}{\partial W} \tag{2.52}$$

where the T refers to the time steps included in a learning task and E is the value of the loss function. The weights of the network can be updated by gradient:

$$W \leftarrow W - \eta \frac{\partial E}{\partial W} \tag{2.53}$$

Then, the gradients of the loss function on the time step n can be formulated via the chain rule:

$$\begin{aligned} \frac{\partial E_n}{\partial W} &= \frac{\partial E_n}{\partial y_n}\frac{\partial y_n}{\partial h_n} \cdots \frac{\partial h_2}{\partial h_1}\frac{\partial h_1}{\partial W} \\ &= \frac{\partial E_n}{\partial y_n}\frac{\partial y_n}{\partial h_n}\left(\prod_{t=2}^{n}\frac{\partial h_t}{\partial h_{t-1}}\right)\frac{\partial h_1}{\partial W} \end{aligned} \tag{2.54}$$

Combined with the precious expression of h_t the derivative of h_t can be computed as:

$$\frac{\partial h_t}{\partial h_{t-1}} = f_H^{'}(W_{xh}x_t + W_{hh}h_{t-1}) \cdot W_{hh} \tag{2.55}$$

Thus, the backpropagated gradient can be expressed as:

$$\frac{\partial E_n}{\partial W} = \frac{\partial E_n}{\partial y_n}\frac{\partial y_n}{\partial h_n}\left(f_H^{'}(W_{xh}x_t + W_{hh}h_{t-1}) \cdot W_{hh}\right)\frac{\partial h_1}{\partial W} \tag{2.56}$$

RNNs usually adapt the tanh function as the activation function, thus $f_H^{'}()$ has a value range from 0 to 1. When the n is large, the gradient tends to 0 and causes vanishing gradients. The exploring gradient problem may occur when the weight W_{hh} is very large. Meanwhile, the vanishing gradient problem causes the forgotten of the previous hidden states, leading to the problem of long-term dependence [159].

Long-short term memory

Long short-term memory (LSTM) [160] is a special class of RNNs, aiming to address the problem of long-term dependence by remaining the previous information for long time steps. Similar to the original RNNs, LSTM composes of a chain of repeating modules while each repeating module contains four interacting layers (figure 2.20).

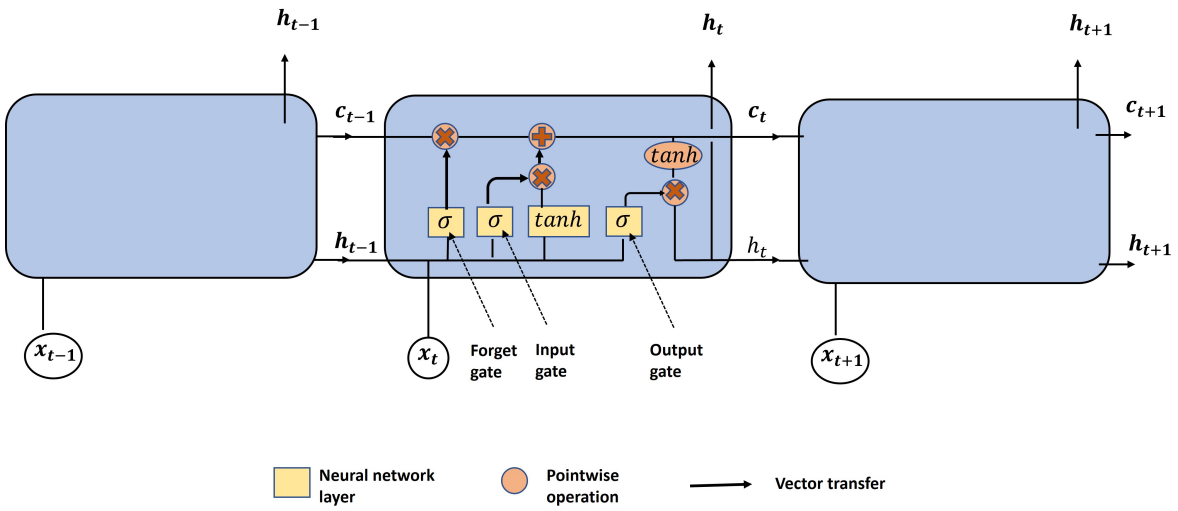

Figure 2.20: Structure of the long short-term memory network at time steps t-1,t,t+1

Figure 2.21 illustrates the structure of a repeating module which is known as the memory cell in LSTM. The core building block of LSTM is cell state c_t which moves forward along the entire chain to transfer the information [161]. The key advantage of LSTM is that LSTM can decides the addition and deletion of the information to the cell state via gates [161]. One Gate consists of a neural network layer with a sigmoid activation function and a point-wise operation, aiming to optionally pass the information through. The output value of the sigmoid function is ranged from 0 to 1, claiming the proportion of the information can be passed through.

There are three types of gates in LSTM: forget gate, input gate and output gate. During the forward propagation, the information is first passed through the forget gate that can remove the unneeded information from the cell state. The state of the forget gate f_t can be expressed as:

$$f_t = \sigma(w_f \cdot [h_{t-1}, x_t]) \tag{2.57}$$

where the σ represents the sigmoid activation function. The second step composes of two parts, deciding what new information can be added to the cell state via the input gate. These two parts have their respective forms:

$$\begin{aligned} i_t &= \sigma(w_i \cdot [h_{t-1}, x_t]) \\ \tilde{c}_t &= tanh(w_c \cdot [h_{t-1}, x_t]) \end{aligned} \tag{2.58}$$

According to the previous computations, the old cell state c_{t-1} can be updated as the

new cell state c_t:

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{2.59}$$

where the $\odot$ refers to the Hadamard product. This step enables the importance and availability of the information added in the cell state. Finally, the output gate works as a filter that regulates the information of the next hidden states (h_t) based on the current cell state. Firstly, the current input x_t and previous hidden state h_{t-1} run through a sigmoid layer for determining the current output. Then, the new cell state c_t is passed through a layer with tanh function, regulating the value range of the new cell state to [-1,1]. Finally, the product of the current output o_t and regulated cell state is the value of the next hidden state.The computation can be written as below:

$$\begin{aligned} o_t &= \sigma(w_o \cdot [h_{t-1}, x_t]) \\ h_t &= o_t \odot tanh(c_t) \end{aligned} \tag{2.60}$$

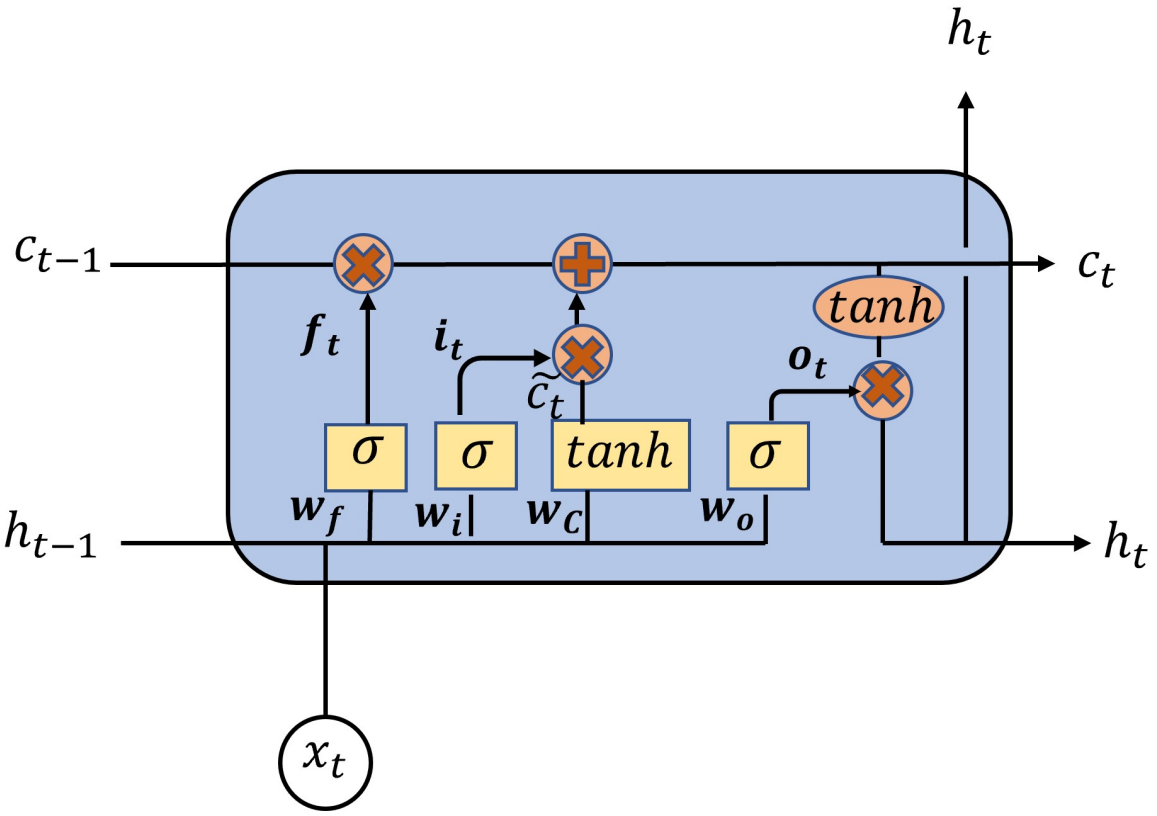

Figure 2.21: Structure of the long short-term memory cell

LSTM utilizes the same backpropagation algorithm (BPTT) [158] during training

process. The gradients of the loss function for time step n have the form:

$$\frac{\partial E_n}{\partial W} = \frac{\partial E_n}{\partial h_n}\frac{\partial h_n}{\partial c_n}\left(\prod_{t=2}^{n}\frac{\partial c_t}{\partial c_{t-1}}\right)\frac{\partial c_1}{\partial W} \tag{2.61}$$

where the $\prod_{t=2}^{n}\frac{\partial c_t}{\partial c_{t-1}}$ causes the vanishing gradients. Different from original RNNs, LSTM adopts the cell state c_t to prevent gradients from vanishing. Notice that c_t can be represented by f_t,i_t and $\tilde{c}_t$, each of these also being a function of c_{t-1} and they are all functions of h_{t-1}. According to the chain rule, the partial derivative of c_t has the form:

$$\begin{aligned}\frac{\partial c_t}{\partial c_{t-1}} &= \frac{\partial}{\partial c_{t-1}}[c_{t-1}\odot f_t + \tilde{c}_t \odot i_t] \\ &= \frac{\partial c_t}{\partial c_{t-1}} + \frac{\partial c_t}{\partial f_t}\frac{\partial f_t}{\partial h_{t-1}}\frac{\partial h_{t-1}}{\partial c_{t-1}} + \frac{\partial c_t}{\partial i_t}\frac{\partial i_t}{\partial h_{t-1}}\frac{\partial h_{t-1}}{\partial c_{t-1}} + \frac{\partial c_t}{\partial \tilde{c}_t}\frac{\partial \tilde{c}_t}{\partial h_{t-1}}\frac{\partial h_{t-1}}{\partial c_{t-1}}\end{aligned} \tag{2.62}$$

According to previous formulations of f_t, c_t, $\tilde{c}_t$ and i_t, the computation of four partial derivative terms are written as:

$$\begin{aligned}\frac{\partial c_t}{\partial c_{t-1}} = f_t \\ &+ c_{t-1}\cdot\sigma^{'}(w_f\cdot[h_{t-1},x_t])\cdot w_f\cdot o_{t-1}\odot tanh^{'}(c_{t-1}) \\ &+ \tilde{c}_t\cdot\sigma^{'}(w_i\cdot[h_{t-1},x_t])\cdot w_i\cdot o_{t-1}\odot tanh^{'}(c_{t-1}) \\ &+ i_t\cdot\sigma^{'}(w_c\cdot[h_{t-1},x_t])\cdot w_o\cdot o_{t-1}\odot tanh^{'}(c_{t-1})\end{aligned} \tag{2.63}$$

Hence, the gradients of the cell state compose of four partial derivative terms that are associated with three gates and the cell state. Thus, LSTM has a better balancing of gradient value during BPTT and addresses the gradient problems during network training. Besides, LSTM utilizes the forget gate to determine which information from the previous hidden state to forget, reducing memory usage and simultaneously enabling the learning of data with long-term dependencies [157]. Moreover, LSTM usually works as a type of layer in deep neural networks. The hyperparameter in the LSTM layer is the units that refers to the number of hidden neurons in this layer and claims the dimensionality of the output vector [162].

Bidirectional long-short term memory

Bidirectional long short-term memory (Bi-LSTM) [163] is a variant of LSTM, enabling data processing in both forward and backward directions. As shown in figure 2.22, the input flows in two directions in Bi-LSTM instead of in the single direction as original LSTM. Note that x refers to the input, h for the hidden states from the forward or backward directions. The forward and backward hidden states are concatenated and passed through the activation function σ, then forming the final output of Bi-LSTM. Through concatenated hidden states, Bi-LSTM stores the information from both future and past input sequences and outperforms original LSTM in processing time sequences [164, 165].

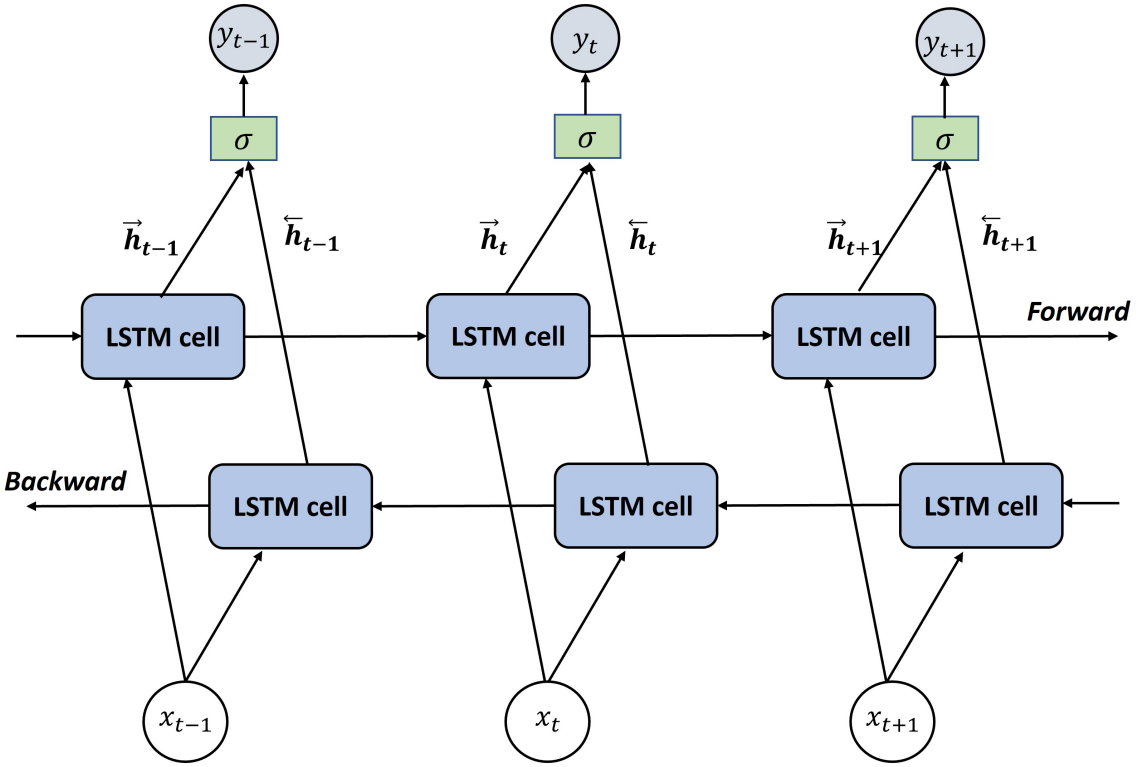

Figure 2.22: Bi-LSTM structure with three consecutive time steps

GRU

As a famous variant of LSTM, Gated recurrent unit (GRU) [166] employs slight changes in memory cell for simplifying the structure. According to figure 2.23, GRU employs a reset gate r and update gate z for controlling the reset and update of the previous information. Initially, the previous hidden state h_{t-1} and current input x_t run through the reset and update gates, obtaining the state of two gates at time step t.

$$\begin{aligned} r_t &= \sigma(w_r \cdot [h_{t-1}, x_t]) \\ z_t &= \sigma(w_z \cdot [h_{t-1}, x_t]) \end{aligned} \tag{2.64}$$

In GRU, the reset gate works as the forget gate that determines the proportion of the previous information to remove, and the update gate decides what previous information to preserve [167]. Thus, GRU utilizes the reset gate to eliminate unneeded information and retain the relevant information in the current memory state $\tilde{h_t}$.

$$\tilde{h_t} = tanh(w_h \cdot [r_t \odot h_{t-1}, x_t]) \tag{2.65}$$

For obtaining the final output of the current unit, GRU uses the state of the update gate to determines how much of the previous hidden state h_{t-1} to remove and the current memory $\tilde{h_t}$ to retain.

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h_t} \tag{2.66}$$

A relatively concise structure contributes to fewer internal parameters in the network than that of the original LSTM, reducing the computational cost. As for the performance in deep learning tasks, LSTM and GRU should be further evaluated depending upon the specific type of tasks.

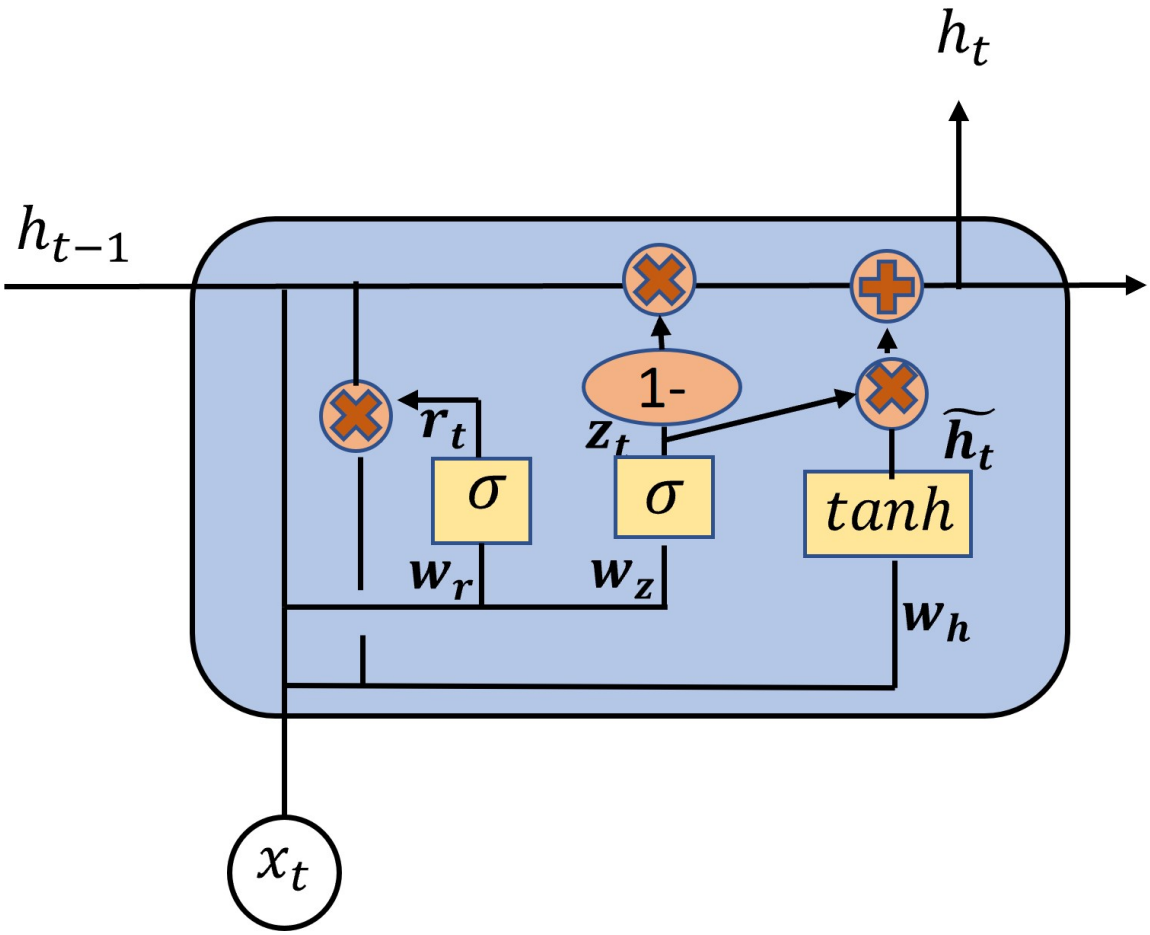

Figure 2.23: Structure of the gated recurrent unit

2.3.5 Residual neural network

The development of deep learning contributes to several state-of-art deep neural networks with deeper structures, such as AlexNet [140], VGG16 [168] and GoogleNet [169]. Evidence illustrates that deep neural networks (i.e. deep CNN) specialize in extracting more complex features and contribute to improved performance. While the very deep structure may cause the vanishing gradient problem, causing a higher training error than that of shallow neural networks [170]. Moreover, in original deep networks is difficult to propagate the information as the increase of network depth, leads to the degradation of the network performance. Residual neural networks (ResNets) address the gradient problems in deeper neural networks via skip connections (shortcuts) between layers [171]. As shown in figure 2.24(b), a typical ResNet composes of multiple residual blocks where each block involves two or three-layer skips. Compared with the plain network structure (figure 2.24(a)), the skip connections transfer the outputs of previous layers to the outputs of each residual block, contributing to a faster approach for gradient updates.

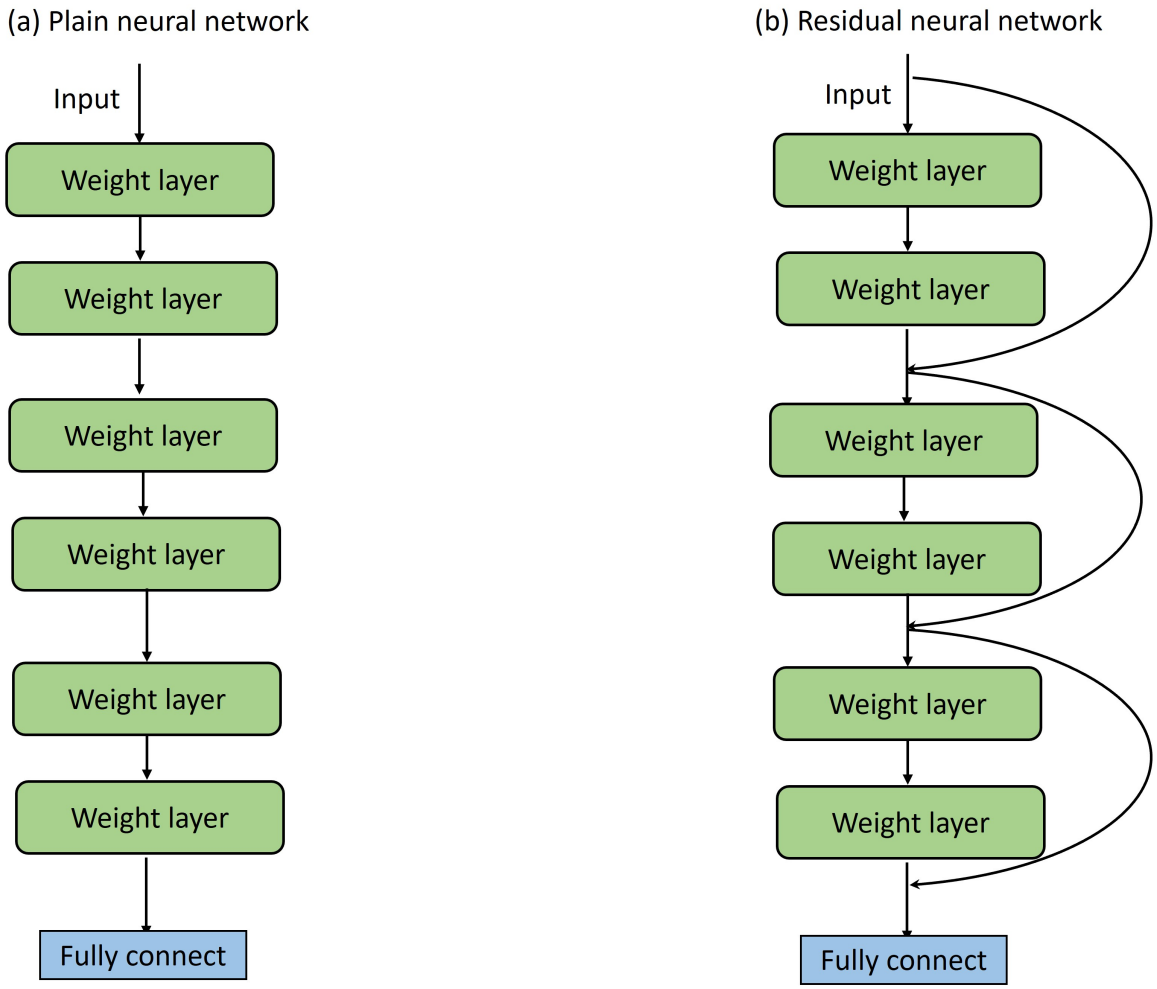

Figure 2.24: (a) A plain neural network (b) Residual neural network

To further describe the algorithms of the ResNet, the typical residual block (figure 2.25) performs the following formulations:

$$x_{l+1} = x_l + F(x_l, W_l) \tag{2.67}$$

where x_l is the input to the $l-th$ residual block, W_l is a set of parameters and $F(\cdot)$ refers to the mapping function contained in the residual block. Suppose there is a $l_2 - th$ block and $l_2 > l$, then recursively unwrap the equation 2.67:

$$x_{l_2} = x_l + \sum_{i=l}^{l_2-1} F(x_i, W_i) \tag{2.68}$$

Thus, the output x_{l_2} of any deeper block l_2 can be mathematically expressed as x_l from any shallower block l plus a residual mapping function. During the forward

propagation, ResNet enables a straightforward transmission of the input from any shallow to any deeper layers, relatively addressing the degradation problem. Denoting the loss function (error) as E, then the backpropagation has the form:

$$\begin{aligned}\frac{\partial E}{\partial x_l} &= \frac{\partial E}{\partial x_{l_2}}\frac{\partial x_{l_2}}{\partial x_l} \\ &= \frac{\partial E}{\partial x_{l_2}}(1 + \frac{\partial}{\partial x_l}\sum_{i=1}^{l_2-1} F(x_i, W_i)) \\ &= \frac{\partial E}{\partial x_{l_2}} + \frac{\partial E}{\partial x_{l_2}}\frac{\partial}{\partial x_l}\sum_{i=1}^{l_2-1} F(x_i, W_i) \\ &= A + B\end{aligned} \tag{2.69}$$

where

$$\begin{aligned}A &= \frac{\partial E}{\partial x_{l_2}} \\ B &= \frac{\partial E}{\partial x_{l_2}}\frac{\partial}{\partial x_l}\sum_{i=1}^{l_2-1} F(x_i, W_i)\end{aligned} \tag{2.70}$$

The gradient of the loss function associated with x_l consists of two additive terms that are represented as A and B. Through the term A, the network directly passes error information back without any computation on weights layers. Even though the weight values of internal layers are small, the gradient hardly approaches zero due to the interaction between the two terms. Therefore, the structure of ResNet enables smoother forward and backward propagation, which effectively resolves the problem of degradation and vanishing gradients.

Besides, the skip connections in different ResNets are various. For example, there are two types of skip connections in ResNet-18 [171]: identity shortcut and projection shortcut. Identity shortcut contains an activation function and batch normalization, straightly transmitting the input to the concatenation operator [172]. While projection shortcut contains a convolution layer combined with batch normalization and ReLu, ensuring that the transmitted input and the output of the residual block have the same size [172]. In general, the structure of ResNets is extensively employed in image processing and classification since it enables the implementation of very deep neural networks with optimal performance [173–175].

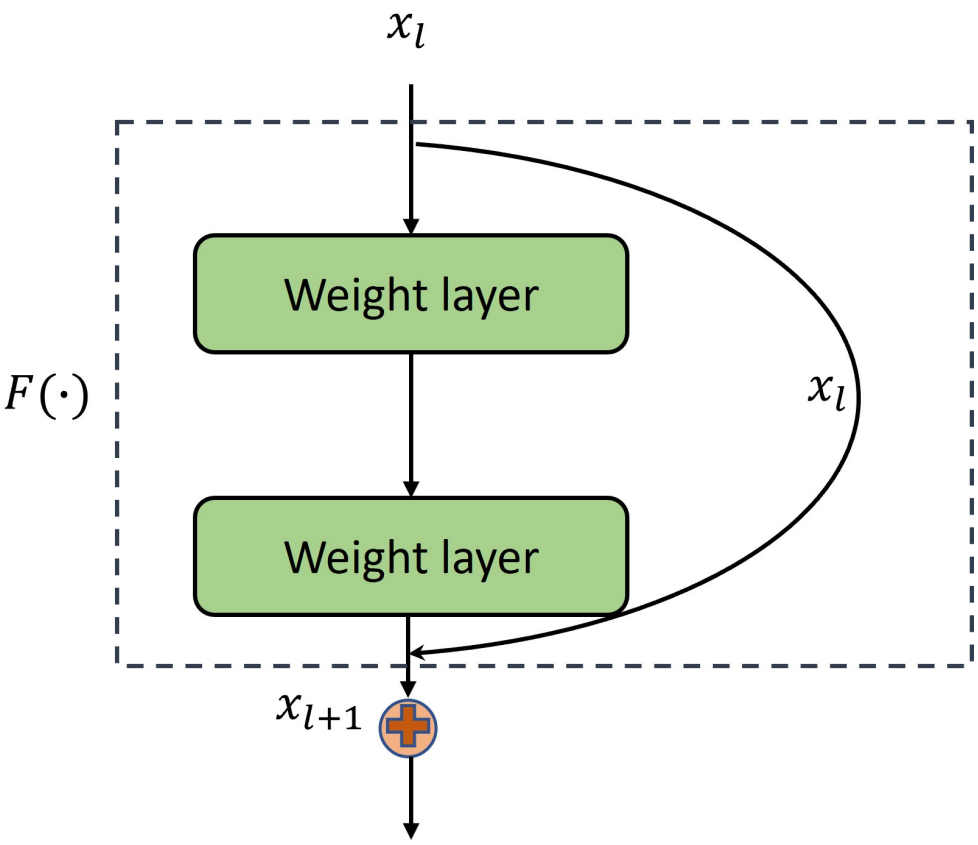

Figure 2.25: Structure of typical residual block

2.4 Experimental Techniques

This section presents the techniques that are adopted in the model implementation showcased later in this thesis.

2.4.1 Python

Python, as one of the most popular programming languages, has recently played a crucial role in data processing and deep learning [176]. Python contains variant built-in functions (i.e. numpy,scipy and matplotlib), allowing users to conduct complex statistical computation, data pre-processing and visualizations. In addition, python supports several types of libraries (i.e. Tensorflow, Keras, Pytorch), enabling the users to write programs for deep learning more efficiently. Compared with C or C++, Python has a simpler syntax and it is friendly to beginners. All models in this thesis are built based on python 3.

2.4.2 Tensorflow /Keras

Tensorflow is an end-to-end open-source framework that is implemented by Google brain team to build machine learning and deep learning models in efficient approach [177]. As for the input data to neural networks, Tensorflow accepts data called tensors which are arrays of data with multi-dimensions, contributing to convenience in processing large amounts of data. Tensorflow provides two types of APIs: high-level APIs and low-level APIs. High-level APIs contain Estimator API and Keras API that provides straightforward approaches to implementation, training and evaluation of the deep learning models. Low-level APIs involve Layers, Metrics, Graphs, Datasets etc. Layers API is the most common-used API in Tensorflow 1, enabling a simpler interface (i.e. tf.layers.Conv2D) for layers in deep neural networks.

To implement deep learning models easier, Tensorflow 2 employed a high-level API-Keras. As one of the most popular Tensorflow APIs, Keras enables fast and easy prototyping in the implementation of deep learning models (neural networks) [178]. Keras is also cross-platform like that Tensorflow, running seamlessly on CPU and GPU.

2.4.3 Colab

Colab or Google Colab is a python-based environment that runs fully in the google cloud [179]. Users can execute python codes and deep learning frameworks (i.e. Tensorflow, Keras, Pytorch etc) in the web browser online. Colab provides users with Python or Jupyter notebook environment and available GPU. The works in this thesis utilize the GPU provided by Colab and the type of GPU is illustrated in the following chapter.

Chapter 3

Literature Survey of ECG Automatic Detection based on Machine/Deep Learning Technologies

According to the background, cardiac diseases have historically been a severe cause of death, and governments devote a large amount of labour and resources to prognosticating various cardiac diseases. Thus, automatic analysis and detection methods have advanced rapidly in recent decades. This chapter introduces the literature review of preprocessing, machine/deep learning techniques for automatic ECG analysis and detection. The whole chapter consists of three subsections: signal preprocessing, machine learning-based studies and deep learning-based studies. The three subsections describe the fascinating advancements made thus far in the area of autonomous cardiac disease diagnosis based on the conventional machine learning and deep learning approaches. This chapter explores the studies or literature related to various machine learning and deep learning techniques for ECG auto-detection, some of which served as inspiration for the works in this thesis.

3.1 Signal Preprocessing

Clinical ECG signals are typically recorded at varied lengths and are noise-filled. To clean raw ECGs and segment signals into the relevant sizes for further feature extraction and classification, preprocessing is an essential step in the ECG classification process. Noise reduction, R-peak, or QRS detection are frequently used in preprocessing stage of both machine and deep learning-based studies.

According to the previous section on ECG, ECG signals can depict the heart rate/rhythm and electrical activities, contributing to the auto-detection approaches for cardiac arrhythmia. These approaches require highly accurate time-domain, frequency-domain and morphological features extracted from ECG signals, which are susceptible to various types of noise such as muscle artefacts (MA), baseline wander (BW), power-line interference etc. Therefore, noise removal is essential for improving classification performance. Several common-used approaches are adaptive filter [180], band-pass filter [181], mathematical morphology [182], weighted averaging filter [183] and independent components analysis [184]. Besides these techniques, wavelet transform (WT) is widely used in traditional machine-learning-based auto-detection for denoising. Using discrete wavelet transform with Daubechies (db6), Sahoo et al. [15] and Martis

et al. [29] decomposed the non-informative frequency components of ECG data and enhanced the significant morphological features of the QRS complex. Moreover, El-Dahshan [185] proposed a hybrid denoising approach based on the wavelet transform (DWT) and genetic algorithm. Through experiments, the proposed approach showed much higher signal-to-noise ratio (SNR) improvements in ECG denoising than pure wavelet-based denoising.

Since the majority of the energy in a heartbeat is contained in the indent QRS complex, determining its precise location and form is essential for ECG auto-detection. Numerous time-domain and morphological parameters, including continuous RR intervals, the amplitude and length of the QRS wave, the morphology and size of the QRS complex, etc., are retrieved from the QRS complex. Initially, the QRS onset and offset points that originate from R peaks are used to calculate the QRS wave's duration. In [186] authors proposed a novel QRS onset and offset detector for ECG signals. This work aimed to compute the indicator that was relevant to the area covered by the QRS complex envelope and obtained an inferior detection rate of 67.5%.

Therefore, an increasing number of researchers have proposed QRS detectors that use digital filters and non-linear transform. Studies of the automatic analysis of ECG signals frequently use the Pan-Tompkins QRS detect algorithm[187]. Researchers employed a series of digital filters for noise removal and periodically adapted thresholds to accurately detect the R-peaks of the filtered ECG signal. The Pan-Tompkins algorithm evaluation result produced a 99.3% correct identification rate. Furthermore, ECG denoising and QRS detection have both made extensive use of the wavelet transform. As early as 1995, Li et al. [188] presented an approach based on wavelet transform for detecting different ECG characteristic points such as QRS complex, P waves, T waves etc. Through the experiments using ECG signals from the MIT-BIH dataset, this approach was robust to the artifacts and baseline wander and obtained an outstanding detection rate of QRS complex of 99.8%. Kadamebe et al. [189] proposed a novel QRS detector based on dyadic wavelet transform. In this work, researchers designed a specific spline wavelet as the mother wavelet of the dyadic wavelet transform, which

is robust to the QRS complex of time-varying ECG signals and also to noise. Additionally, Pal et al. [190] proposed a novel method for detecting ECG characteristic points. They used the multi-resolution wavelet transform to locate the locations of R peaks, Q and S points, and then P and T waves in turn. The manual measurement in the evaluation process may be the main flaw, despite the fact that the test result of the proposed approach on the PTB diagnostic database is above 99%. In addition to wavelet transform, empirical mode decomposition (EMD) can help identify ECG characteristics and reduce noise. In [191] authors presented an EMD-based R-peak detection algorithm which showed a promising result in R-peak detection of the stress ECG. Similarly, Slimane et al. [192] proposed an EMD-based non-linear transform algorithm that increased QRS complex detection specificity and sensitivity by above 99%.

3.2 Machine learning-based studies

For traditional machine-learning-based ECG auto-detection algorithms, preprocessing, feature extraction and classification are three key processes and are processed separately via different tools. The research of machine-learning-based algorithms for ECG detection is explored in the following subsections, which are organised into three steps.

3.2.1 Feature extraction

A crucial procedure in machine-learning-based algorithms is feature extraction. Researchers aim to obtain discriminating features from ECG signals to represent different types of arrhythmia. Some of them proposed purely extracting time-domain features and heart rate features from original ECG signals. While others intend to collect, merge and reduce the dimension of different types of features for better classification performance.

Approaches based on original ECG signals

Time-domain characteristics and other heart rate features were simply taken from the original ECG signals in earlier investigations of auto-detection for arrhythmias without any prior decomposition or wavelet transformation. Earlier in 1997, Yu et al. [193]

formed a 12-dimensional feature vector which involves time-domain features (RR interval, random points at both sides of the R peak). This study used a mixture-of-expert (MOE)[194–196] based classifier to accurately discriminate PVC from Non-PVC ECG beats, with an accuracy of 94%. Fei et al. [197] extracted 8 temporal features, including RR interval, length of QRS wave, length of T wave, QT interval and others. This study proposed a particle swarm optimization-SVM model that had a 96.65% accuracy for detecting the arrhythmia cordis. In this study[198], researchers utilized the Pan-Tompkins algorithm [187] and a proposed P-wave detector, efficiently locating the R peaks and P waves from the continuous ECG signal. Then, with an accuracy of 94.6%, they combined time-domain characteristics (RR intervals, PR intervals) and heart rate information (amplitude of P and R point) to perform continuous Holter monitoring for aberrant ECG beats. Compared with previous studies, this work handled comparatively straightforward feature extraction and classification and also maintained strong performance in binary classification.

Hybrid approaches based on transformed ECG signals

The research stated in the previous section demonstrates that features solely derived from the original ECG signals are constrained and unsuitable for more complex multi-class classification. Additionally, traditional machine learning auto-detection methods demand the comprehensive progress of feature extraction and selection, and the extracted features severely affect the classification accuracy. As a result, a number of hybrid approaches to ECG analysis have been proposed to improve classification performance.

In [199] authors employed temporal features computed from RR intervals of the ECG signal as well as the morphological features extracted from the QRS complex of ECG by wavelet transform. In this study, researchers combined the temporal and morphological features as well as used expert aid to improve classification performance. They were able to classify three different types of arrhythmias with an average accuracy of 98%. similarly, In [200] authors also integrated temporal features (RR intervals) with symmetrical morphological features obtained via discrete orthogonal stockwell transform (DOST). The combined feature sets and SVM optimized by particle swarm

optimization achieved an ideal accuracy of 99.81% in classifying 16 types of ECG beats. Heart rate variability (HRV) signal feature extraction was conducted by Asl et al. [201] in addition to utilising the temporal and morphological properties of ECG data. HRV signal is generated from the ECGs. The HRV accurately depicts the cardiac activities and is a crucial diagnostic tool for cardiac arrhythmia. Initially, researchers collected 15 features obtained from HRV signals via linear and nonlinear analysis. These features include 7 time-domain features, 1 frequency domain feature, and various nonlinear features including approximate entropy, Lyapunov exponent, etc. The dimensionality of the feature set was then reduced by the generalized discriminant analysis (GDA) [202], while keeping the highest discriminating features. The SVM-based classifier finally distinguished 6 types of arrhythmias with an outstanding accuracy of 99.16%.

Several studies [28, 29, 203] have proposed using discrete wavelet transform (DWT) and principal component analysis (PCA) to conduct feature extraction. DWT is a transform which can decompose the input signal into multiple coefficient sets, each of which describes the time evolution of the signal across different frequency bands[31]. A statistical method called principle components analysis (PCA) can make huge datasets more interpretable and retain their useful information while reducing their dimensionality[27]. In aforementioned studies, researchers utilized DTW to decompose the ECG signal into high-frequency and low-frequency components (sub-bands) with corresponding frequency ranges. The required sub-bands are preserved for the subsequent operation in accordance with the frequency range of the ECG (0.1-35hz), and the coefficients in the remaining sub-bands are adjusted to zero. Then, to lessen the dimensionality, PCA was applied to DWT sub-bands. The final step is to feed the obtained features into various classifiers, with the classifier[203] achieving the highest results with a sensitivity of 99.85%.

Besides previous studies, the combination of PCA and other techniques are also efficient. Similar to this, Ince et al. [204] used the PCA to reduce the dimensionality of the extracted features after extracting morphological features from ECG beat signals using the translation-invariant dyadic wavelet transform (TI-DWT). This study also

merged the morphological features with temporal features, achieving an accuracy of 95.58% for 5 classes of ECG beats via the multidimensional particle swarm optimisation classifier. In [205] authors employed Higher Order Statistics (HOS) techniques with PCA to extract the bispectrum features from ECGs and process reduction of dimensionality. For classifying 5 different types of ECG beats, this work used SVM as the classifier and achieved an accuracy of 93.48%. The author[29] used DWT and Independent Component Analysis (ICA) for feature extraction and dimensionality reduction, attaining a greater accuracy of 98.36% on the same types of ECG beats.

3.2.2 Classification

Approaches using Fuzzy-based algorithms

Fuzzy-logic-based machine learning approaches are frequently employed for accurate ECG detection. In [206] authors employed a fuzzy decision tree as the classifier to diagnose the abnormalities of 4 types of heart beats that were gathered from the MIT-BIH arrhythmia dataset. They used 11 features from the morphological, temporal and frequency domains for feature extraction. A further improvement might be achieved to the correctly identified rate, which was 71%. Lei et al. [207] implemented an adaptive fuzzy ECG classifier for diagnosing four major forms of cardiac arrhythmia based on the temporal and morphological features in a similar manner. The proposed classifier, which claimed self-adaptability based on the input ECGs and obtained a higher average accuracy of 88.2%, was assessed using the same database. Moreover, in [208] authors presented a fuzzy-logic classifier based on subtractive clustering. They utilized the subtractive clustering techniques for fuzzy rules parameterization and implemented a novel classification strategy that was robust to noise and arrhythmic outliers. In comparison to the above studies, this fuzzy-logic classifier attained the optimal accuracy of 97.41%. The combination of fuzzy logic and K-Nearest Neighbour also performed well in efficient ECG beat detection. In the work [209], researchers implemented a pruned fuzzy KNN classifier for classifying 6 types of ECG beats. The proposed classifier analysed a large number of ECG beats with less computational time and achieved an accuracy of 97.35%.

Approaches using SVM

Support vector machine (SVM) aims to identify an optimal hyperplane in a N-dimensional space (where N is the number of features) that classifies different types of data points with the maximum distance between each other (in figure 3.1). Automatic diagnostics

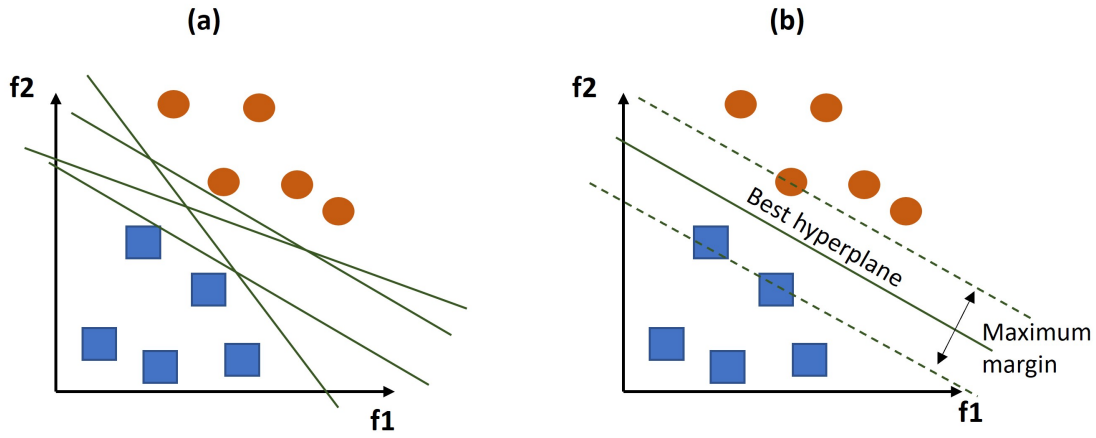

Figure 3.1: (a) Possible hyperplanes in 2 dimensional space. (b) Optimal hyperplane in 2 dimensional space.

based on ECG are frequently performed using SVM, a well-liked classifier. As early as 2009, Kampouraki et al. [17] used SVM to classify ECGs into two different groups with 100% accuracy. This study showed that the SVM classifier has greater noise robustness through comparison with other nonlinear classifiers in ECG classification.

Although the previous studies show satisfying classification accuracy, the simple binary classification is unable to cope with various common cardiac diseases. For multi-class classification of ECGs, Martis et al. [210] employed the least square support vector machine (LS-SVM) for detecting 5 types of different arrhythmias from the MIT-BIH arrhythmia dataset. The LS-SVM classifier is validated through specificity, sensitivity, positive predictive value (PPV) and accuracy. The LS-SVM with RBF kernel achieved the optimal accuracy of 98.11%. Similar to this, the study [19] employed LS-SVM with RBF kernel to identify 6 types of sleep apnea events based on ECGs obtained from Physionet and polysomnographic recordings made at the sleep laboratory in the University Hospital Leuven. The researchers merely used 4 types of features extracted from ECGs, reaching a satisfying accuracy (>85%) on clinical data.

SVM and its variations have been widely used in recent years to increase the accuracy and efficiency of ECG auto-detection. In this study[20], researchers proposed a novel approach based on twins SVM and hybrid swarm optimizers. In comparison to earlier SVM-based studies, the classification accuracy of this study of ECG heartbeats using the same MIT-BIH arrhythmia dataset was ideal (99.44%). Furthermore, Raj et al. [18] proposed a novel approach for feature extraction and classification for 16 types of arrhythmias. The researchers utilized SVM with particle swarm optimisation (PSO), which gradually tunes the learning parameters of the SVM. This study demonstrated the benefit of combining SVM with PSO for multi-class classification by contributing an ideal accuracy of 98.82% for categorising 16 classes of arrhythmias.

Approaches using KNN

K-Nearest Neighbor(KNN) is a well-known and simple classification algorithm in the machine learning domain. KNN algorithm is based on the feature similarity which is decided by the distance (Euclidean, Manhattan, Hamming) between the test sample and train samples. The distance between the test sample and the training set determines the predicted class of the test sample (Figure 3.2). KNN is valuable in

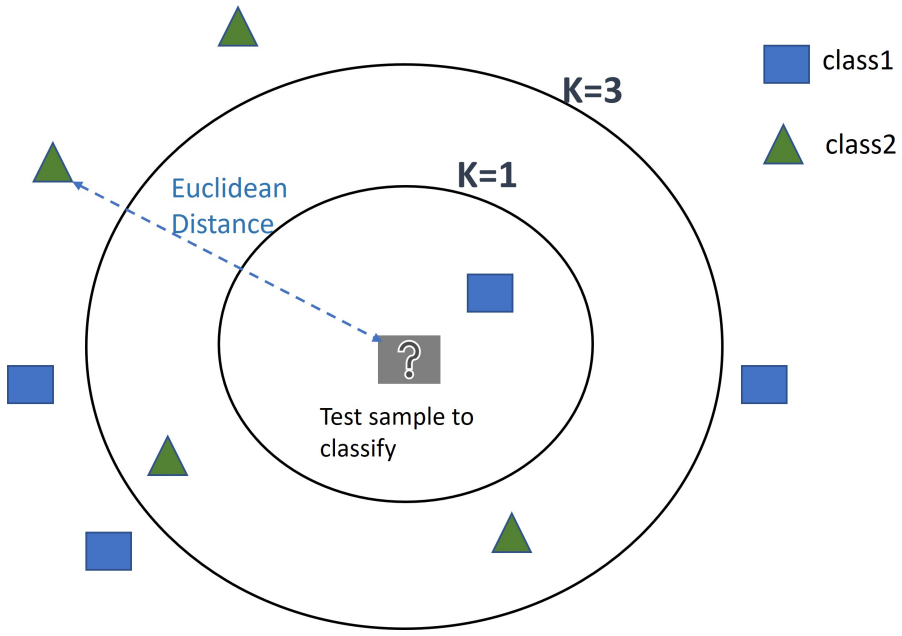

Figure 3.2: KNN working progresses. KNN classifier selects k nearest neighbors based on the calculated Euclidean distance. If k =1, the test sample is assigned to the class 1 because there is only one blue square inside the small inner circle. And if k=3, the test sample is assigned to the second class (2 green rectangles > 1 blue square).

the automatic ECG arrhythmia detection as well. Bong et al. [211] employed KNN to classify the emotional stress based on 3 types of time-domain features that were derived from ECG signals. This method obtained a greater accuracy of 11.65% than that of SVM in binary classification. Yusuf et al. [212] conducted feature extraction using Mel Frequency Cepstrum Coefficient (MFCC) on PTB diagnostic database and used a KNN classifier to diagnose myocardial infarction (MI) based on extracted features, showing an accuracy of 84%. According to the study [23], the KNN classifier reportedly achieved a high accuracy of 97.5% on the MIT-BIH database. For the binary classification of ECG beats, time-domain and frequency-domain features obtained from the HRV signal are applied to the KNN classifier.

Recently, the performance of arrhythmia auto-detection has been enhanced by the integration of KNN and other approaches. In [213] authors proposed a fusion classifier that consisted of KNN, SVM and four MLP classifiers, achieving an average accuracy of 98.2% for diagnosing 7 types of arrhythmias.

Approaches using Decision Tree

Decision Trees (DT) are a non-parametric machine learning algorithm and are popular for classification tasks. The way DT implements classification models is in form of a tree structure (Figure 3.3), where the branches are the decision rules and leaf nodes are the outcomes of those decisions. Since the DT-based classifiers can be visualized clearly and robust to outliers, they are frequently employed in the classification of ECG signals. Using a decision tree with DWT and PCA for feature extraction, Zhang et al. [214] categorised 6 different types of ECG beats with a test accuracy of 96.31%. For improving classification performance, in [215] authors proposed an adaptive boosted optimized decision tree as the classifier which is robust to the data with uncertain features values and labels. The proposed method was tested on the same dataset as the study of Zhang et al. [214], contributing to an outstanding accuracy of 98.77%. Moreover, Mert et al. [26] utilized the decision tree with ensemble learning for ECG beats classification. In the classification process, researchers used a bagged decision tree to analyse the extracted time-domain features and achieved an optimal accuracy of 99.34%. Previous studies demonstrated the satisfying performances of the Decision

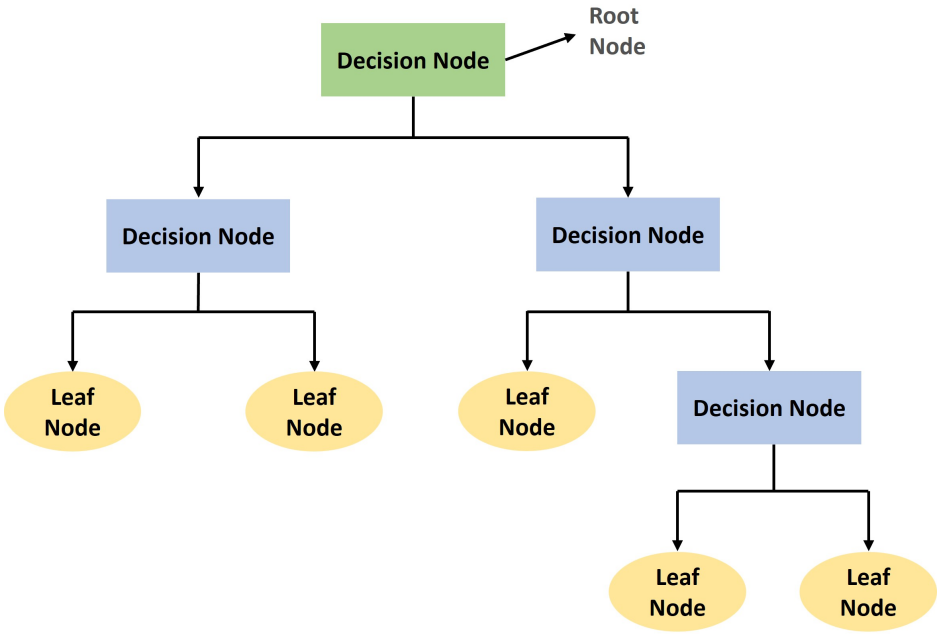

Figure 3.3: Structure of a classification tree.

Tree by evaluating the proposed algorithms on the single lead ECGs from the MIT-BIH arrhythmia dataset. DT-based classifier also performed well in 12-lead ECG signals. Kasar et al. [216] used the J48 decision tree to detect myocardial infarction based on 12-lead ECG classification, and obtained an accuracy of 98.5%.

Critical Analysis

The above literature review introduced various auto-detection approaches based on traditional machine learning techniques in the last decades. Table 3.1 lists examples of feature extraction, classifiers, and classification outcomes. These machine-learning-based arrhythmia detection approaches focused on manual feature extraction and the distinctiveness of features directly affect the classification performance. Despite the high classification accuracy of these machine-learning-based approaches, manual feature extraction is still complicated and time-consuming. Furthermore, these approaches mainly analysed 1-D ECG signals for binary or multi-class classifications, whereas multi-label classification of multi-lead ECG was only occasionally used[217, 218]. Thus, traditional machine-learning-based approaches were inefficient in more complex arrhythmia classification tasks.

Table 3.1: Comparison of the ECG auto-detection methods in traditional machine learning domain.

Literature	**Feature Extraction**	**Classifier**	**Types of ECG beats**	**Accuracy**
Yu et al.[193]	QRS complex + PCA	MOE	2	94%
Fei et al.[197]	time intervals	Particle swarm optimization-SVM	5	95.65%
Park et al.[198]	QRS complex, p wave	Decision Tree	2	94.6%
Llamedo et al.[199]	RR interval, QRS complex	LD	3	98%
Raj et al.[200]	DOST	PSO tuned SVM	16	99.81%
Asl et al.[201]	HRV features	GDA+SVM	6	99.16%
Martis et al.[28]	Cumulant + PCA	LS-SVM	5	-
Martis et al.[29]	WT + ICA	SVM	5	98.36%
Martis et al. [210]	QRS complex + PCA	LS-SVM	5	98.11%
Martis et al.[205]	HOS + PCA	SVM	5	93.48%
Ince et al.[204]	DWT +PCA	SVM	5	95.58%
Omar et al. [206]	HRV features	Fuzzy decision tree	4	71%
Lei et al.[207]	timing, morphological ECG features	adaptive fuzzy	4	88.2%
Homaeinezhad et al.[208]	DWT	fuzzy-logic+ subtractive clustering	4	97.41%
Arif[209]	WT+PCA	Purned fuzzy KNN	6	97.35%
Kampouraki et al.[17]	HRV features	SVM	2	100%
Houssein et al.[20]	EMD	Swarm-TWSVM	16	99.44%
Bong et al. [211]	HRV features	KNN	2	77.69%
Yusuf et al.[212]	MFCC	KNN	2	84%
Venkatesan et al.[23]	HRV features	KNN	2	97.5%
Homaeinezhad et al.[213]	geometrical features	KNN+ SVM+MLP	7	98.2%
Zhang et al.[214]	DWT+PCA	DT	6	96.31%
Kumari et al.[215]	timing, morphological ECG features	adaptive boosted optimized DT	6	98.77%
Mert et al. [26]	timing features	bagged DT	6	99.34%
Kasar et al. [216]	timing, morphological ECG features	J48DT	2	98.5%

3.3 Deep learning based studies

Deep learning approaches are commonly called deep neural networks that are developed rapidly with a crucial impact on the classification performance for various medical application, including arrhythmia detection. End-to-end deep neural networks now provide more straightforward approaches for ECG auto-detection. Deep neural networks can conduct autonomous feature extraction and classification, handling a vast volume of datasets with superior robustness to traditional machine-learning-based approaches. The preprocessing procedure of deep learning methods is similar to that of machine-learning-based approaches. Thus, the following subsections mainly explore the arrhythmia detection approaches of different types of deep neural networks.

3.3.1 Feature extraction and classification

Multilayer Perception

According to the preceding section, MLP consists of multiple layers (input, hidden, output) of neurons fully connected with each other in a feed-forward manner and is trained with back propagation. Sannino et al. [219] used the deep neural network to classify 5 types of heart beats from the MIT-BIH arrhythmia dataset. Researchers conduct the noise removal and segment filtered ECG signals into heart beats during the preprocessing stage. The proposed MLP contains 7 hidden layers with the ReLu activation function, with 5, 10, 30, 50, 10 and 5 neurons in each layer, respectively. The training and testing accuracy of this study were 100% and 99.09% respectively. The MLP with one (arrhythmias) vs. rest (normal) strategy was proposed by Raut and Dudul[220] and effectively categorised 16 different types of heart states gathered from the UCI Machine Learning database. In this work, three different classifiers were examined, and the proposed MLP with two hidden layers surpassed the others with an accuracy of 88.24%.

Convolutional Neural Network and its Variants

CNN is the one that is most frequently utilised in various deep learning application of arrhythmia detection. For binary-classification tasks, Pourbabee et al. [221] proposed a 5-layers CNN model as the feature extractor, directly extracted features from

raw ECG signals and successfully screened the paroxysmal atrial fibrillation patients. These studies [222, 223], which showed a high accuracy of about 98%, also utilised 1-dimensional CNN as end-to-end models for differentiating between two separate heart beats. Acharya et al. [224] utilized an 11-layers 1-dimensional CNN to distinguish between ECG beats of normal and myocardial infarction (MI). Notably, the average accuracy of the filtered ECG beats, which was higher than that of the noise-filled ECG beats, was 95.22%. Additionally, these authors used a 9-layers 1-dimensional deep CNN that classified 5 types of ECG beats [39]. The CNN model achieved 93.47% and 94.03% accuracy using ECG beats with and without noise respectively, and demonstrated its insensitivity to the ECG signal quality. A popular variation of CNN called residual neural network (ResNet) increases the depth of neural network while also enhancing classification performance. Hannun et al. [225] implemented a 34-layers CNN model based on the structure of the ResNet, classified 12 ECG classes with an average area under the receiver operating characteristic curve of 0.97.

The frequency components of the ECG signals were disregarded in the experiments mentioned above, which used 1-dimensional CNN to extract features from the ECG data. A number of studies [226–228] proposed using time-frequency representations to depict the changes of ECG signals in both time and frequency domains. Xia et al. [226] employed stationary wavelet transform (SWT) to obtain time-frequency spectrograms for 5-second ECG segments. The spectrograms are fed into a 5-layer, 2-dimensional CNN as the inputs suited for a deep learning classifier, and this achieved an accuracy of 98.63% for the detection of atrial fibrillation. The short-term Fourier transform, in addition to the wavelet transform, is frequently employed in time-frequency analysis. Diker et al. [228] utilized discrete short-time Fourier transform, transformed 1-dimensional ECG signals into spectrograms. Researchers used three different pre-trained DCNN models for training and testing on the PTB dataset, where the AlexNet[140] obtained the highest accuracy of 83.82%. In this study, there were merely 160 and 80 ECG samples for training and testing, whereas the DCNNs were not suitable for less training data, probably leading to over-fitting.

The authors [229] used STFT to create spectrograms for 5 types of arrhythmias from

the MIT-BIH arrhythmia database without any noise removal. They evaluated the classification accuracy of the proposed 2-dimensional shallow CNN with that of 1-dimensional CNN, finding that the 2-D CNN achieved a higher accuracy of 99% and demonstrated its robustness to noise. Recently, Rashed et al. [227] also utilized a pre-trained model which demonstrated excellent performance on image classification and reduced the computational complexity during the training process. Researchers segmented the long-term ECG signals into 2.4s-long chunks, where each chunk includes 3 beats. This study achieved an optimal classification performance (99.09%) for 5 types of arrhythmias cases, where the chunks were processed as 2-dimensional time-frequency scalograms by continuous wavelet transform (CWT) and then fed into VGG16-based DCNN.

Aforementioned studies introduced CNN-based approaches for analysing single-lead ECG signals. While the 12-lead ECG signals can provide more useful features for arrhythmia detection. Based on the 12-lead ECG, Lodhi et al. [230] created a 20-layer CNN to distinguish between normal and MI. A voting technique was utilised to make the final prediction using the CNN outputs, and it achieved a classification accuracy of 93.53%. Although the accuracy is relatively satisfying, adding more CNN layers may increase the computational complexity. Baloglu et al. [231] implemented a 10-layers 1-dimensional CNN model and correctly classified 10 types of MI ECG beats from the PTB dataset. Based on the technique of QRS detection, researchers in this study separated each lead ECG into numerous ECG beats. The proposed CNN model achieved the highest accuracy on the lead V4 (99.78%) and an average accuracy of 99.6% among all ECG leads. In a similar manner, Park et al. [232] conducted an accurate pulse segmentation via different QRS detectors for each lead and voted the correct location of the QRS complex. Then, they processed the MI detection using the VGG16-based model and the ResNet34-based model, where the ResNet34-based model achieved the outperforming specificity and sensitivity of 89.6% and 93.2% respectively. More studies for processing the 12-lead ECG using the hybrid models discussed in the section below.

Recurrent Neural Networks and its Variants

Recurrent neural network and its variants were initially implemented for sequential data, specifically applied to machine translation[233], text generation[234] and speech recognition[235]. Unlike CNN, RNN neurons in the hidden layers can store relevant memory and utilize previous information for prediction. RNN and its variants are now frequently employed in bio-signal processing. Zhang et al. [236] implemented a patient-specific ECG classification model for distinguishing between VEBs, SVEBs and other ECG beats. The morphological features including the T waves between two adjacent heart beats were fed into a 4-layers RNN to learn the potential features and classify the ECG beats automatically. Through the evaluation on MIT-BIH dataset, the proposed model achieved an accuracy of 99.4% and 98.7% for VEBs and SVEBs, respectively. As the variant of traditional RNN, LSTM is equipped with more efficient memory backup system and solves the problem of long-term dependencies. Yildirim[237] proposed a novel classification model which consists of wavelet sequence (WS) layers and Bi-LSTM. Without using handcrafted features, this model automatically extracted features from the WS layer using the wavelet transform. The proposed model achieved a high accuracy of 99.39% for 5 types of ECG beats when evaluated on the MIT-BIH arrhythmia dataset. The combination of WS layer and Bi-LSTM amply demonstrated its superiority in classification performance when compared to other state-of-art CNN-based models. Along with ECG signals, the LSTM model also worked successfully in the RR intervals sequence. Faust et al. [238] extracted the RR intervals sequences from the original ECG signal in the MIT-BIH atrial fibrillation dataset and then divided the RR sequences into overlapping sub-sequences. The proposed Bi-LSTM used sub-sequences as input and obtained an accuracy of 99.77% in AF detection.

GRU reduced the number of gates on the basis of LSTM, which helped to reduce memory usage. Some studies of arrhythmia detection utilized the traditional RNN, LSTM and GRU to conduct the control experiment. In [239] authors normalized and segmented raw 1-dimensional ECG signals into chunks where each chunk contains almost 3 beats. The accuracy of the 3-layer LSTM, RNN, and GRU models used to classify five different types of ECG beats was 88.1%, 85.4%, and 82.5%, respectively.

Sujadev et al. [240] directly adopted the raw ECG signals as the input of RNN, LSTM and GRU models. The evaluation was placed on the same dataset as the study of Singh [239], and both LSTM and GRU achieved an satisfying accuracy of 100% for AF detection. From the standpoint of results, LSTM is marginally better than GRU.

Hybrid approaches

Faced with non-stationary and non-linear ECG signals, various hybrid deep learning models were implemented for better classification performance and robustness. Andersen et al. [241] proposed an end-to-end model which consisted of CNN and LSTM. That model detected the ECG beats of AF during the training and testing on three different ECG datasets. The presented model contained 2 convolutional layers, one max-pooling layer and one LSTM layer, scored accuracy of 87.4% on the test set. In a similar manner, Petmezas et al. [242] implemented a hybrid deep learning model which combined CNN and LSTM for detecting 4 types of arrhythmias that integrated CNN and LSTM. The proposed model contained three 1-dimensional Convolutional layers, 3 Max-pooling layers and one LSTM layer, which obtained an accuracy of 97.87%. The performance of the model is improved to some extent by correctly raising the neural network's depth in accordance with the input samples. In [40] authors built a novel deep model using CNN and LSTM that consisted of two sub-networks and was successful at classifying 6 types of ECG signals. In preprocessing stage, the raw ECG signal was denoised and divided into 10s segments. Unlike previous studies, the proposed model needed two inputs to feed the 10s ECG segments and its RR intervals sequences into two different sub-networks. Evaluated on the MIT-BIH arrhythmia database and the unseen database, the hybrid model achieved an optimal accuracy of 99.32% and 97.15% respectively.

However, mindlessly increasing the depth of the CNN may lead to the anomaly of gradients and over-fitting. Thus, some proposed models utilized the residual network structure, simultaneously smoothed the gradient descent and prevented the model from over-fitting. Chen et al. [243] built a novel deep learning model based on the architecture of ResNet for classifying 5 classes of ECG. The proposed model accepted two parallel inputs that converge the time-based segments and beat-based segments

to obtain both the temporal and morphological features simultaneously. Then, the parallel inputs were separately fed into the combination of ResNet and LSTM for automatic feature extraction, fusion and classification. The model attained a better accuracy of 99.56% and 96.77% under the inner-patient and intra-patient paradigms, respectively, during the evaluation of the MIT-BIH arrhythmia dataset. ResNet-based deep learning models are widely used in the analysis of 12-lead ECG signals. Yao et al. [244] implemented a time-incremental CNN combined with a varied-length LSTM sub-network to distinguish 9 types of ECG based on 12-lead ECG signals. Researchers used the ECG data from China Physiological Signal Challenge 2018 dataset for training and testing. With an F1 score of 0.773, the suggested model outperformed previous VGG-based models. Furthermore, Chen et al. [245] presented a novel model that merged CNN,Bi-LSTM and attention mechanism. The presented model successfully classified 9 types of ECG and ranked first in the China Physiological Signal Challenge 2018 (CPSC2018). In a similar manner, in [246] author further proposed a combination of ResNet-based CNN and bi-LSTM. The proposed model achieved an overall F1 score of 0.806 on the hidden test dataset from the CPSC2018 and placed third in the challenge. The advantages of LSTM to store historical data and handle long-term dependencies were merged with the capability of CNN to precisely extract critical features from ECG inputs, which resulted in a hybrid model with stratifying accuracy and robustness.

Researchers confronted the difficulty of data imbalance, which results in subpar performance in minority classes, due to the paucity of clinical data on some rare arrhythmias. Previous studies adopted cross-validation, resampling and copy samples for addressing the problem of data imbalance. Recently, researchers utilized Generative Adversarial Networks (GAN) to address imbalanced ECG datasets. GAN [247] usually consists of a generator and discriminator. As for ECG analysis, the generator tries to trick the discriminator by producing the fake signals, while the discriminator is trained to distinguish between the real and fake signals. Through adversarial training, the discriminator lapses and the generator is responsible for producing new realistic signals. In [248] authors built a hybrid model using a CNN-based GAN and LSTM for distinguishing the normal and abnormal ECG signals. Researchers implemented a single

LSTM model and modified the discriminator as a One-vs-Rest binary classifier. The input ECG was fed into Bi-LSTM and GAN models in parallel, and then the outputs of the two models are concatenated for final classification. The proposed GAN-LSTM model was evaluated on the MIT-BIH dataset and the PTB dataset, obtained an average accuracy of 99.3%. Additionally, Golany et al. [43] proposed to used a CNN-based GAN in ECG generation, which significantly improved the classification accuracy of 5 types of ECG beats. A simple LSTM model worked as the classifier in this study. Experiment results showed that the incorporating GAN-generated data increased test accuracy by 4%. Additionally, Zhu et al. [249] proposed a novel GAN composed of CNN and Bi-LSTM for ECG generation. Two Bi-LSTM layers served as the generator and two convolutional layers and two max-pooling layers served as the discriminator in the proposed GAN model. Through the evaluation between the original signal and generated signal, the proposed GAN model had the lowest percent root mean square difference (PRD) among other GAN models. Previous studies have shown that GAN was effective at ECG generation and greatly expanded the imbalanced database. However, the utilization of GAN-based models increases the difficulty of training due to the problems of non-convergence, mode collapse and being highly sensitive to the hyperparameter selections. For better use in ECG data augmentation, the unstable training procedure and additional computation of GAN might be further improved.

Critical Analysis

The aforementioned studies provide an overview of various deep learning-based auto-detection approaches. Table 3.2 illustrates feature extraction, classifiers and results. Since these approaches employed different pre-processing and datasets, the simple horizontal comparison between deep learning approaches and machine learning approaches is not objective. In the general aspect of classification performance, the deep learning-based approaches outperformed beat traditional machine learning-based approaches. Nevertheless, some manual feature extraction still existed in aforementioned deep-learning-based studies[236, 238], leading to extra time cost. Additionally, it may be not efficient for extracting discriminated features from complex ECG signals purely using traditional CNN and training algorithms. Different types of ECG have similar morphological ECG representations. Besides, broadening the scope of the research is

also important. Some techniques and algorithms in other domains (speech recognition, computer vision) can be used in ECG automatic detection domain, contributing to better classification performance and efficiency. Thus, the researcher aims to adopt advanced preprocessing techniques, neural networks and training algorithms for more discriminative ECG feature extraction. Additionally, the rareness of some types of arrhythmias can lead to extremely insufficient training samples, further affecting classification performance. Unlike previous studies to handle the imbalanced dataset, the researcher intends to investigate a novel approach to address this problem from the source forward.

Table 3.2: Comparison of the ECG auto-detection methods in traditional deep learning domain.

Literature	**Feature Extraction**	**Classifier**	**Types of ECG beats**	**Result**
Sannino et al.[219]	end-to-end	MLP	5	Accuracy:99.09%
Raut et al.[220]	end-to-end	MLP	16	Accuracy: 88.24%
Pourbabaee et al.[221]	end-to-end	CNN	2	-
John et al.[222]	end-to-end	CNN	2	Accuracy: 99.56%
Acharya et al.[224]	end-to-end	CNN	2	Accuracy: 95.22%
Acharya et al.[39]	end-to-end	CNN	5	Accuracy: 94.03%
Hannun et al.[225]	end-to-end	ResNet	12	ROC: 0.97
Xia et al.[226]	SWT+end-to-end	CNN	2	Accuracy:98.63%
Diker et al.[228]	DSTFT+end-to-end	AlexNet	2	Accuracy: 83.82%
Huang et al.[229]	STFT+ end-to-end	CNN	5	Accuracy:99%
Rashed et al.[227]	CTW+end-to-end	VGG16	5	Accuracy:99.09%
Lodhi et al.[230]	end-to-end	CNN	2	Accuracy: 93.53%
Baloglu et al. [231]	end-to-end	CNN	10	Accuracy: 99.6%
Park et al.[232]	end-to-end	ResNet34	2	specificity:89.6% sensitivity:93.2%
Zhang et al.[236]	T-wave+RR interval	RNN	3	Accuracy:99.05%
Yildirim[237]	end-to-end	WSBi-LSTM	5	Accuracy:99.39%
Faust et al.[17]	RR intervals	Bi-LSTM	2	Accuracy:99.77%
Singh et al.[239]	end-to-end	LSTM	5	Accuracy:88.1%
Sujadev et al.[240]	end-to-end	LSTM,GRU	2	Accuracy:100%
Anderson et al.[241]	end-to-end	CNN+LSTM	2	Accuracy:87.4%
Petmezas et al.[242]	end-to-end	CNN+LSTM	4	Accuracy:97.87%
Chen et al.[40]	end-to-end	CNN+LSTM	6	Accuracy:99.32%
Chen et al.[243]	end-to-end	ResNet+LSTM	5	Accuracy:99.56 %
Yao et al.[244]	end-to-end	CI-CNN	9	F1 score:0.773
He et al.[246]	end-to-end	ResNet+Bi-LSTM	9	F1 score:0.806
Chen et al.[245]	end-to-end	CNN+Bi-LSTM	9	F1 score:0.84
Rath et al.[248]	end-to-end	GAN+LSTM	2	Accuracy:99.3%
Golany et al.[43]	end-to-end	GAN+LSTM	5	Accuracy:94.25%
Zhu et al.[249]	end-to-end	GAN+Bi-LSTM	-	-

Chapter 4

Automatic Detection for Multi-Labeled Cardiac Arrhythmia Based on Frame Blocking Preprocessing and Residual Networks

Study presented in this chapter was published in:

Li, Z. and Zhang, H., 2021. Automatic Detection for Multi-Labeled Cardiac Arrhythmia Based on Frame Blocking Preprocessing and Residual Networks. Frontiers in cardiovascular medicine, 8, p.616585.

4.1 Introduction

Cardiac arrhythmias refer to irregular heart rhythms, representing abnormal cardiac electrical activities associated with abnormal initiation and conduction of excitation waves in the heart [250]. Cardiovascular diseases in association with cardiac arrhythmias can cause heart failure, stroke, or sudden cardiac death [251]. Early detection and risk stratification of cardiac arrhythmias are crucial for averting severe cardiac consequences. With their ability to represent useful information regarding the electrical activity of the heart, electrocardiograms (ECG) measured via electrodes placed on the body surface played an important role in diagnosing cardiac abnormalities [252]. Recently, artificial intelligence-based algorithms [253, 254] have shown promises in screening abnormal features of ECG to achieve an automatic diagnosis of cardiac arrhythmias with high accuracy but less labor demand.

In previous studies, several auto-detection algorithms have been developed [15, 255]. These algorithms focus on extracting physiological features of ECGs, such as heart rate variation (calculated from the time interval between two consecutive R peaks), the width of the QRS complex, and QT intervals. However, these algorithms do have limitations for practical application, as ECG features were merely extracted from RR or QT intervals, providing insufficient information for multiple types of cardiac event classification. To extract sufficient features automatically and achieve high classification accuracy, recent advancements in deep neural network [256] helped to develop several improved auto-detection algorithms [225, 254, 257] for ECG analysis and classification. These studies illustrated that the deep-learning-based algorithms have the advantages of extracting and processing ECG features automatically.

However, the algorithms discussed earlier are mainly focused on processing single-lead ECG rather than the 12-lead ECG, which is commonly used in the clinical setting for providing more diagnostic information than a single-lead ECG on cardiac excitations [258]. Also, it is still a challenge to auto-detect multi-types of cardiac diseases based on 12-lead ECG due to:

1. similar morphological features of ECG among different types of diseases, such as between atrial fibrillation (AF) and premature atrial contraction [259].

2. imbalanced ECG data for various heart diseases in some training datasets, which

may result in excessive bias or over-fitting of the neural network for diagnosis.

3. unequal recording length of clinical ECG recordings, which may result in loss of some essential signals in the process of preprocessing for training the neural network.

Therefore, this study aims to develop a novel method for preprocessing raw ECGs and design an appropriate neural network for classifying 12-lead ECG data with multi-labeling and varied lengths.

4.2 Related Works

Previous works into ECG auto-detection are mainly focused on manual feature extraction via the analysis in the time domain, frequency domain, and ECG morphology. After feature extraction, machine learning methods, such as Support Vector Machine [260] and linear discrimination analysis [261], are usually used for classifications. Compared with the algorithms mentioned earlier, ECG auto-detection based on deep neural networks focuses more on automatic feature extraction from ECG signals.

Hannun et al. [225] developed a deep CNN model for auto-detection of 12 classes of cardiac rhythms, achieving an averaged F1 score of 0.837. Besides, models based on LSTM have also been developed for processing ECG data with varied recording lengths and long-term time dependence to avoid the loss of valid features [262]. For multiple label classification, the combined use of different neural networks demonstrates a better performance than the network structure purely based on the convolution layer. For example, the algorithm of multi-information fusion neural networks [243] consisting of BiLSTM and CNN has the advantages of simultaneously extracting the morphological features and temporal features, yielding an accuracy of 99.56%. Moreover, a similar BiLSTM–CNN model has been introduced to process data with long-term correlation, which could sufficiently extract features [241] to achieve high sensitivity and specificity of 98.98 and 96.95%, respectively.

The ECG auto-diagnosis algorithms discussed earlier demonstrated the advantages of deep learning algorithms in classification accuracy but were less focused on processing

the 12-lead ECG with multiple diagnosis labels. Thus, it is in demand to develop an effective and auto-diagnostic algorithm to classify 12-lead ECG data for multiple cardiac arrhythmias.

4.3 Methodology

The proposed algorithm for classifying 12-lead ECG with multi-labeling consists of components of data denoising, framing blocking, and dataset balance for data preprocessing and a neural network structure based on ResNet in combination with attention-based bidirectional long short-term memory (BiLSTM). The general structure of the proposed algorithm is shown in Figure 4.1.

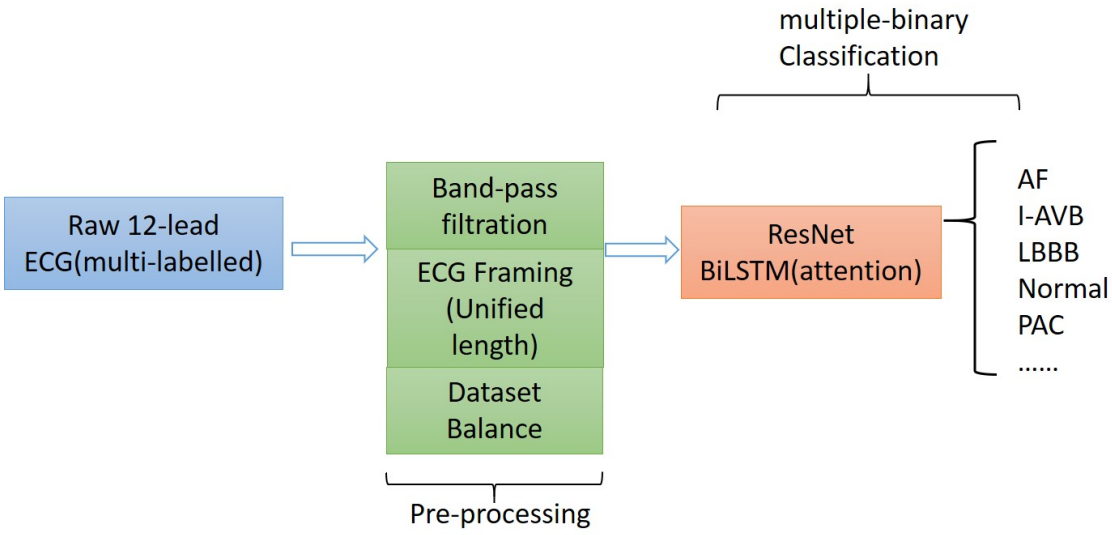

Figure 4.1: Flow chart diagram of the algorithm for multi-type cardiac arrhythmia classification

4.3.1 Dataset Description

China Physiological Signal Challenge in 2018

The China Physiological Signal Challenge (CPSC) 2018 dataset consists of 6,877 (females: 3,178; males: 3,699) recordings of 12-lead ECG data collected from 11 hospitals. Each recording is saved as a MAT file with a hea file presenting labels and relevant information of the ECG recording at the end of the file. The ECG recordings are sampled at 500 Hz with different recording lengths, ranging from 6 to 60 s. The

dataset contains ECG recordings for nine types of cardiac states, including AF, intrinsic paroxysmal atrioventricular block, left bundle branch block (LBBB), normal heartbeat (Normal), premature atrial contraction (PAC), premature ventricular contraction (PVC), right bundle branch block (RBBB), ST-segment depression (STD), and ST-segment elevation (STE).

To illustrate the morphological variation of the ECG among different cardiac states, the visualization of ECG lead II waveforms for nine types of cardiac states and a multi-labeled ECG recording can be found in Appendix Figures A1, A2, respectively. Among the 6,877 recordings, 476 of them have two or three different labels. Table 4.1 lists the numbers and distribution of eight-type cardiac arrhythmias in the 476 multi-labeled recordings of the CPSC 2018 dataset.

Table 4.1: Numbers and distribution of ECG recordings with multiple labels [245] for eight different types of abnormalities in CPSC2018.

	AF	**I-AVB**	**LBBB**	**RBBB**	**PAC**	**PVC**	**STD**	**STE**
AF	0	0	29	172	4	8	33	2
I-AVB		0	8	10	3	5	6	4
LBBB			0	0	10	6	3	4
RBBB				0	55	51	20	19
PAC					2	3	6	5
PVC						0	18	2
STD							0	2
STE								0

China Physiological Signal Challenge in 2020

An independent dataset, the CPSC 2020 dataset, is also used for testing the robustness of the proposed model. The dataset from CPSC 2020 contains two subsets of annotated recordings, one with 6,877 (males: 3,699; females: 3,178) recordings and the other with 3,453 (males: 3,453, females: 1,610) recordings of 12-lead ECG data, each of which was collected by a sampling frequency of 500 Hz.

Furthermore, the dataset from CPSC 2020 contains public and unused datasets from

the CPSC 2018 dataset for seven common types of cardiac states, details of which are listed in Table 4.2 for the total number and distribution of cardiac abnormality in the CPSC 2020 dataset. Except for normal heart rhythm, the numbers and distribution of six types of abnormalities in multi-labeled recordings of the CPSC 2020 dataset can be found in Appendix Table B.1. In the experimental process, the total recordings for seven common types of cardiac states in CPSC 2020 were used for robustness testing.

Table 4.2: The records numbers of 7 types of abnormalities in CPSC2020.

Abnormalities	CPCS2020 Training set1	CPCS2020 Training set2	Total
AF	1221	153	1374
I-AVB	722	106	828
LBBB	236	38	274
Normal	918	4	922
RBBB	1857	1	1859
PAC	616	73	689
PVC	0	188	188

PTB XL

To demonstrate the universality and robustness of the proposed algorithm, the cross-validation of the algorithm was processed on the PTB XL dataset. The PTB XL dataset comprises 21,837 clinical 12-lead ECG records from 18,885 patients (males: 9,820, females: 9,064) of 10-s length. As a multi-labeled dataset, the ECG records were annotated by two cardiologists based on the Standard Communication Protocol for Computer-Assisted Electrocardiography standard [263]. Table 4.3 illustrates the distribution of diagnosis, where the diagnostic labels are aggregated into superclasses.

Table 4.3: Recording numbers of distribution of 5 types of diagnostic labels in PTB XL.

Superclass	Description	Record Num
Norm	Normal ECG	9,528
MI	Myocardial Infarction	5,486
STTC	ST/T Change	5,250
CD	Conduction Disturbance	4,907
HYP	Hypertrophy	2,655

4.3.2 Preprocessing

Noise Processing

Most ECG signals have a frequency range between 0.1 and 35 Hz and are non-stationary in the low-frequency range [264]. Noises normally contaminate them from sources of power-line interference, muscle movement, and baseline wander, which blur the features of the ECG signals for classification. For minimizing possible effects of noise on model classification, raw ECG data in the two databases were denoised by using an eight-order Butterworth lowpass (35 Hz) filter for eliminating noise and removing baseline wander.

Frame Blocking

Clinical ECG data are normally collected with non-uniform duration, ranging from 10 s to 24 h, causing difficulties for training and testing neural networks. For unifying the length of each of the ECG recordings, a frame blocking method adapted from speech recognition [265] is utilized in the present study. In speech recognition, frame blocking is used to segment speech signals into short frames with overlapping, enabling a smooth transition between adjacent frames that maintains the continuity of the signal.
As there is a similarity between speech signals and ECG time series [266, 267], the frame blocking method can be implemented in ECG data for unifying their recording length. Figure 4.2 A illustrates the implementation of the frame blocking method on the cardiac signal. In the figure, F_s, the frameshift, denotes the time lag of the frame

(from the starting time of the ECG recording), and f_o denotes the overlapping part between adjacent frames. Thus, the length of each frame, F_l, can be expressed as:

$$F_l = F_s + f_o \tag{4.1}$$

For a raw ECG recording with a total length of Sl, given the number of frames Fn and frame length Fl, then the framing equation can be represented as:

$$F_s = (S_l - F_l)/(F_n - 1) \tag{4.2}$$

The length of each ECG recording in the CPSC 2018 dataset is variable, of which 6,634 recordings have their length shorter than 40 s (i.e., 20,000 sampling data points). To retain the available ECG signals for each record as much as possible, researcher sets F_l and F_n as a constant of 2,000 (sampling points) and 10, respectively, but F_s variable for fitting the required length and number of frames. As for some records that has length longer than 40 s, the researcher retained the 40 s signal segment from the start of the record and dropped rest of signals. Figure 4.2B illustrates an example of a 12-lead ECG recording processed by the frame blocking, with each ECG recording can be transformed into a frame-block with a uniform size [i.e., (F_n, F_l, leadnum)]. As such, the frame blocking acted on each lead of the signals and divided them into 10 frames with a frame length of 2,000 sampling points.

Dataset Balance

In the present study, the multi-labeled dataset was converted into multiple types of sub-dataset classes, each of which represented one of the multiple types of cardiac states. The length of the ECG data for each type of cardiac abnormalities is imbalanced, leading to over-fitting and weak generalization of the proposed neural network. Due to the high volume of the dataset, random over-sampling method may cause duplicated samples then further lead to the oversize of training data and over-fitting problem. Thus, Random under-sampling method [268] is used to address imbalanced dataset. For training and testing each binary-classifier, data samples are selected randomly from the dataset until a 2:1 ratio of samples in the majority class to the minority class is obtained.

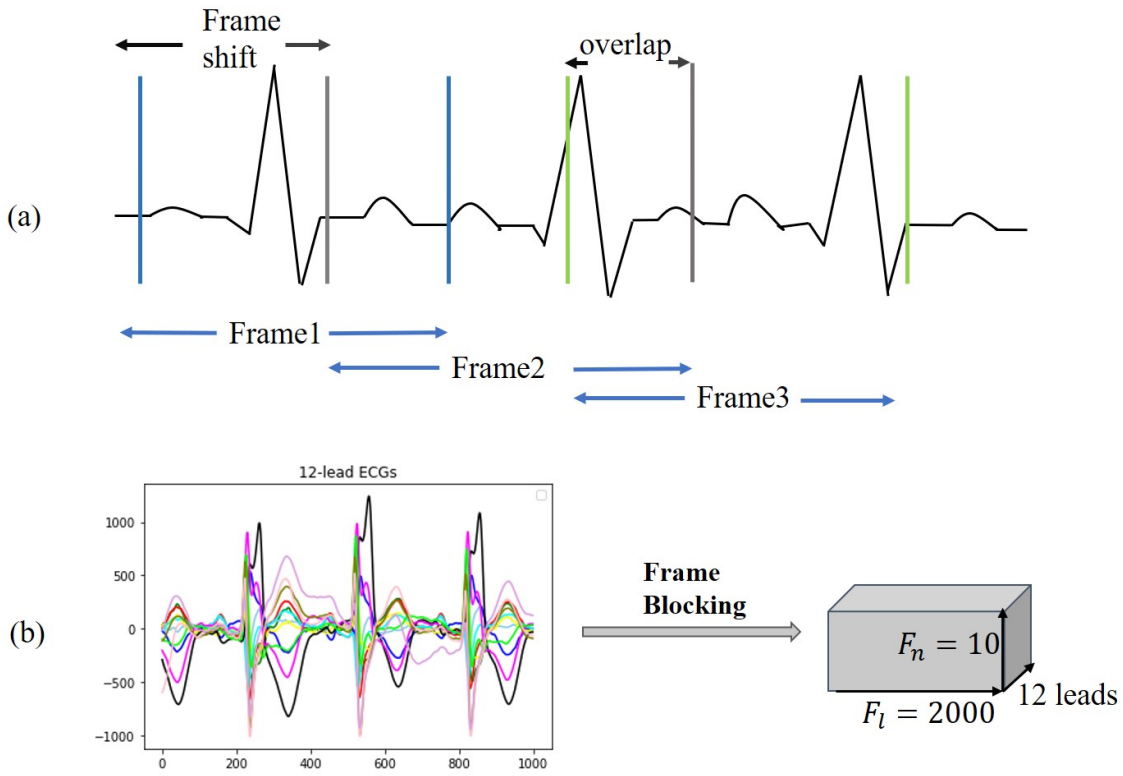

Figure 4.2: Illustration of frame blocking for pre-processing ECG signal. (A) Method of frame blocking. (B) Example of 12-lead ECG data segments after frame blocking processing.

4.3.3 Construction of the Model

Residual Convolution Neural Network

Residual convolutional neural network (CNN) [171] has shown excellent performance on image recognition for addressing the degradation problem of a deeper neural network, and it is believed to be useful for analyzing time-series signals, such as ECG. Here, researcher implemented one-dimension residual CNN with 13 layers based on the structure of ResNet. As shown in Figure 4.3 for the general structure of the network, both dense blocks 1 and 2 belong to the residual block, and the shortcut connection simplifies the optimization of the deep neural network.

Attention-Based Bidirectional Long Short-Term Memory

In the proposed model, the residual blocks primarily focus on extracting features from ECG signals, and the attention-based BiLSTM structure focuses on learning and analyzing the feature map produced by the residual blocks. The bidirectional structure

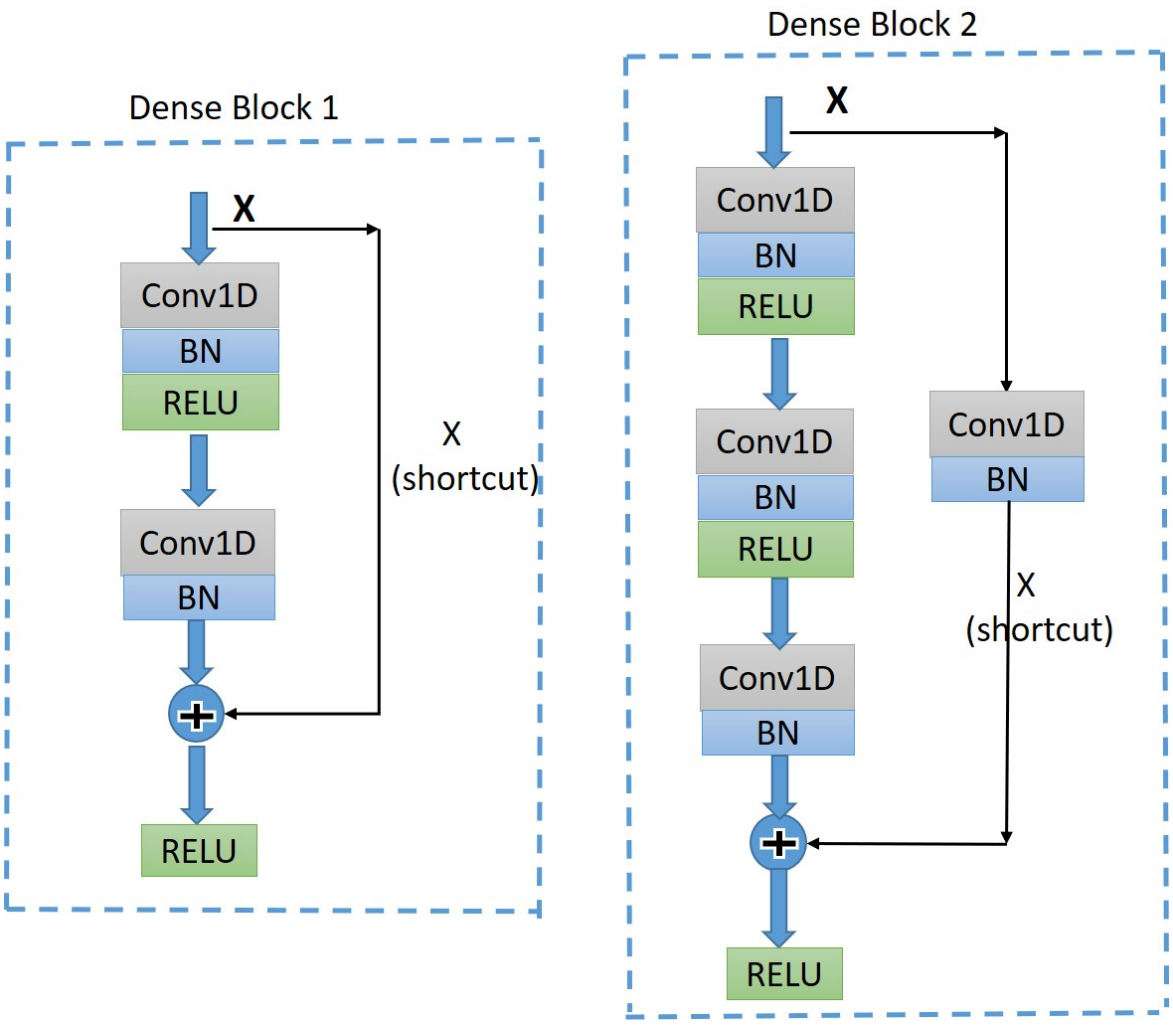

Figure 4.3: Diagram of the structure of dense block1 and dense block2. BN, batch normalization; ReLu, rectified linear units; Conv1D, one-dimension convolutional layer.

provides contextual information in the forward and backward directions for the output layer, providing more prediction information [269]; thus, in this study, a BiLSTM [266] is used to catch some essential information from a long-distance correlation of the ECG data.

The proposed model implements the Attention Mechanism (Figure 4.4) to allocate different attention values to each input query, which assists BiLSTM to precisely identify valid information and reduce the loss of key features. The attention-based BiLSTM can focus on the essential part of the input, meanwhile, it catches global, and local connection precisely because of the weight and attention allocation for the input time sequences.

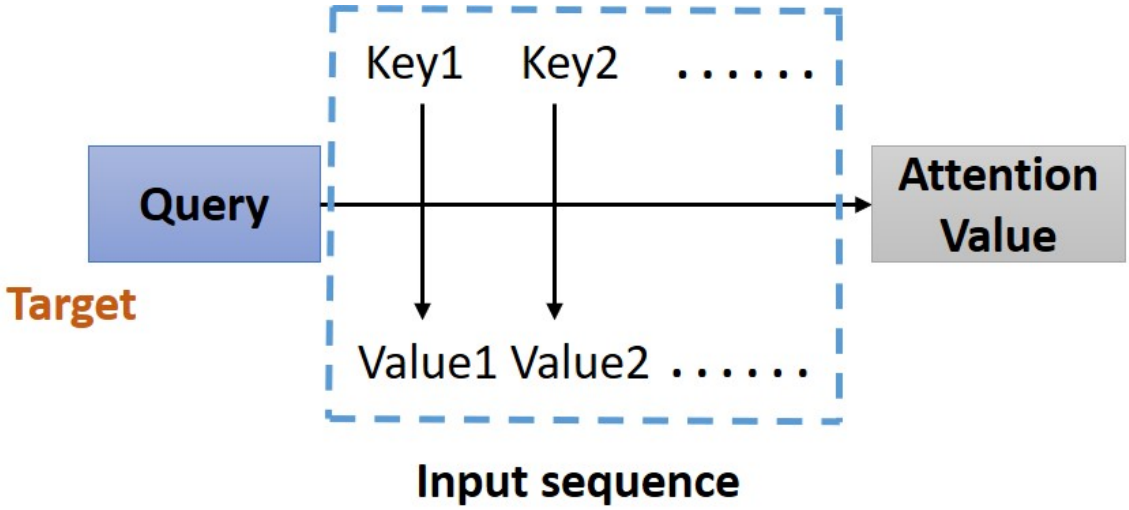

Figure 4.4: Structure of Attention Mechanism. Correlation between the key-value pairs of the input time sequences and the query (a condition value) is evaluated, based on which the weight of each value is calculated. Through the weighted summation, the attention value for each element in the input time sequence can be assign

4.3.4 Structure of the Proposed Network

The whole structure of the proposed model was inspired by the He et al.[246] which model also consists of a ResNet and Bi-LSTM. The two types of dense blocks are built based on the structure of ResNet18[171], aiming to perform a faster and smooth approach for gradient updates in a deep neural network. Additionally, the attention-based Bi-LSTM is skilled in identifying the key part of the input so that it extracts more discriminative features than traditional Bi-LSTM. Finally, the softmax layer produces the probability distribution for each binary classifier.The overall structure of the proposed model is shown in Figure 4.5. The dense block 1 shown in Figure 4.3 is a standard residual block in ResNet[171]. It consists of two one-dimension convolutional layers (Conv1Ds), two Batch Normalization (BN) [270], and rectified linear units (ReLu) [139] for the activation function layers, as well as a shortcut connection that transmits the input to output directly before applying the second ReLu nonlinearity. As for the structure of dense block 2, a Conv1D layer and BN are added in a shortcut for adjusting channels or stride to fit the desired shape of output.

Following the Conv1Ds are the BN and ReLu layers, which help to simplify the parameter adjustment, improve the learning speed, and address the vanishing gradient

problem of the model. Then a 1D max-pooling layer is used to down-sample the feature map by computing and extracting maximums of every three values in the feature map matrix, thus retaining the most valuable features and avoiding unnecessary memory usage during the training process. After the max-pooling layer, dense block 2 is connected to dense block 1 to fulfill a complete residual CNN. Before processing the attention-based BiLSTM for feature analysis, the global average pooling layer [271] is used to process the regularization of the global structure of the network, preventing it from overfitting.

Appendix Table B2 lists a set of optimal parameters for each layer and the residual blocks. Among different residual blocks in different positions (first or second), convolution kernels have different sizes and numbers. For classification, sigmoid activation with binary cross-entropy [272] is used to convert the output sequence from the last LSTM layer into a probability for a specific label, based on which classification is determined with a given threshold. Additionally, The training algorithm of the proposed model is listed below:

Algorithm 1 Pseduocode for training algorithm of proposed model. number of classes M

Input: Training Dataset $D = \{x_1, x_2, ..., x_n\}$. labels$Y = \{y_1, y_2, ..., y_n\}$. Classifiers set $C = \{c_1, c_2, ...c_m\}, c_i \in proposed\_model$, an embedded function f which represents the proposed classifier.

Output: The loss L for the mini-batch training

for i in $\{1, 2, ..., M\}$ **do**

$Train\_classifier(c_i, (x_i, y_i))$ ▷ training process

$L \leftarrow L + \left[\frac{1}{n}\sum_{i=1}^{n} y_i log(p_i) + (1 - y_i) log(1 - p_i)\right]$ ▷ Loss update

end for

Validation and Test procedure

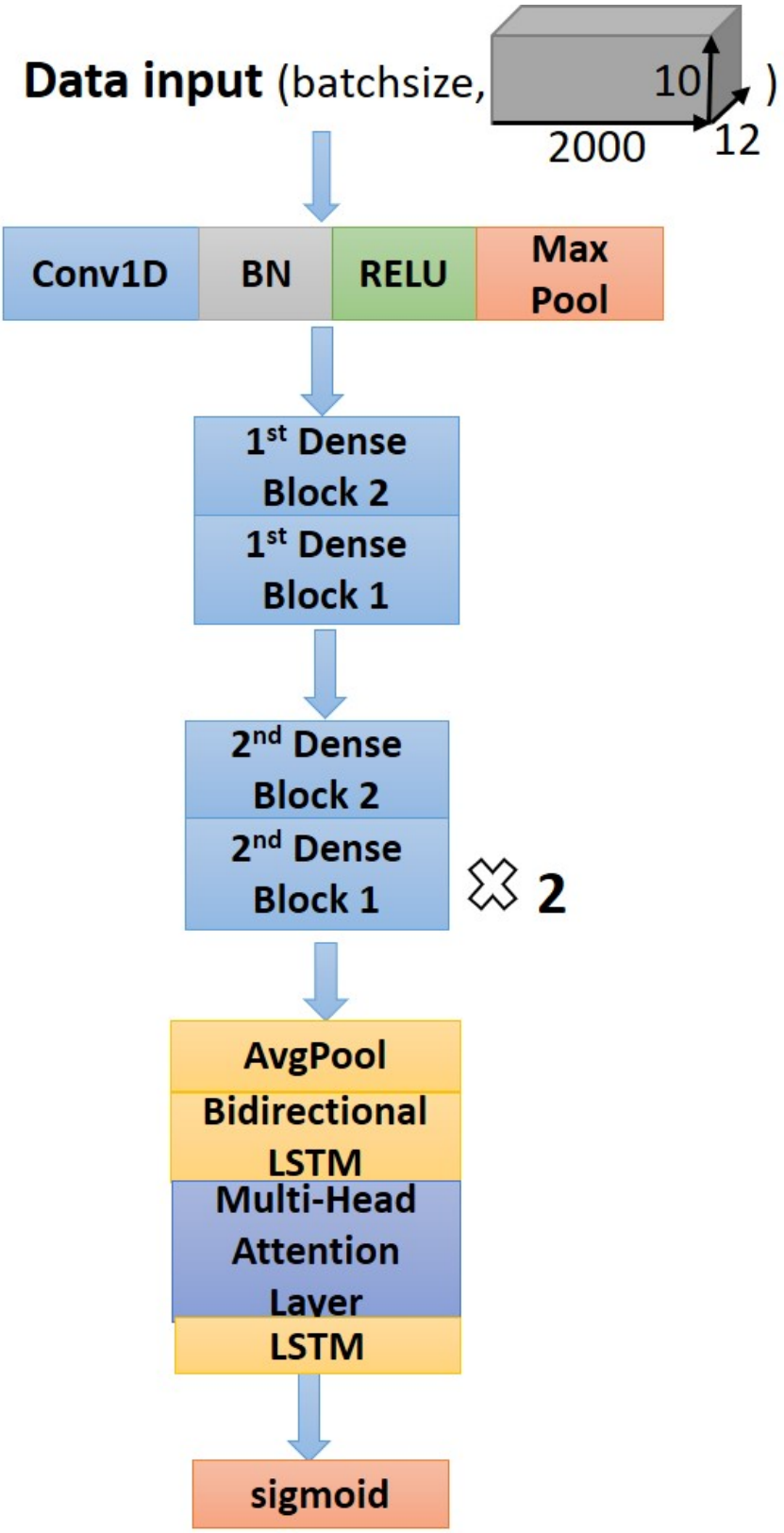

Figure 4.5: Structure of the proposed neural network.

4.3.5 Experimentation Details and Evaluation Matrix

The proposed model is initially trained and implemented using the CPSC 2018 datasets and run on Tesla T4 GPU with Keras frameworks [176]. As described in the section of dataset balance, the positive and negative samples of each cardiac abnormality with the ratio of 1:2 are randomly selected and combined as input datasets for the model. For each binary classifier, the input data were divided into three subsets: 64% for training, 16% for validation, and 20% for testing. The 5-fold cross-validation was also implemented for training and validation. The test dataset was used purely for evaluating the performance of the model and was not involved in training and validation of the proposed model.

The classification performance can be comprehensively evaluated by precision, Recall, F score, receiver operator characteristic (ROC) curve, and area under the curve (AUC). These evaluated measures are calculated by the following equations:

$$Precision_i = \frac{TP_i}{TP_i + FP_i} \tag{4.3}$$

$$Recall_i = \frac{TP_i}{TP_i + FN_i} \tag{4.4}$$

$$F1score_i = \frac{2 \times (Precision_i \times Recall_i)}{Recall_i + Precision_i} \tag{4.5}$$

In these equations, researcher denotes each of the types of cardiac arrhythmias. TPi and TNi represent the number of correctly predicted positive and negative samples, respectively. On the other hand, FP and FN are the values of false prediction for positive and negative samples separately. The ROC curve measures the performance of the model via plotting the trade-off between sensitivity and specificity, and the AUC is the value of the area under the ROC curve. A ROC curve is closed to the top-left corner and has the AUC close to 1 indicates the good performance of the classification model.

4.4 Result

4.4.1 Adjustment of Hyperparameter

Hyperparameters such as learning rate, regularization, optimization and kernel parameters of hidden layers can be tuned for balancing the memory usage and model performance. This section aims to tune the chosen hyperparameters which involves kernel size and numbers of two types of dense blocks, the number of BiLSTM units, dropout rate and the number of units in the multi-head attention layer. Learning rate is an adjustable parameter in the optimization algorithm, controlling the step of backpropagation and optimization time. The kernel parameters of convolutional layers determine the model performance and the computation load.

In the proposed model, dropout layers are added between the average pooling layer, bidirectional LSTM layer and the attention layer. Dropout rate represents the percentage of the activation of neurons which has been deactivated. Shielding neurons properly helps to reduce complex coadaptation between neurons and the number of parameters thereby resolving the problem of over-fitting. In the multi-head attention layer, the input sequences are divided into a given number (number of heads) of segments, and then Attention Mechanism is applied on the corresponding segments. Therefore, the model can attend to relevant information simultaneously from different representation subspace at a different position.

After evaluating the model performance via cross-validation, the most proper setting of hyperparameters is shown in Table 4.4 and evaluated result is shown in Appendix Table B3 in details. The mean squared error on the validation set is used as the standard of evaluation. Procedure of hyperparameter tuning is discussed in the following.

1) Learning rate

   This study uses Adam optimizer with a learning rate of 0.001 as suggested (Kingma Ba, 2014). The decay rate for 1st and 2nd moment estimates was set as default, resulting in fast convergence and less memory requirement.

2) Kernel parameters of residual neural network

The kernel size and numbers of the convolutional layers are key hyperparameters which should be evaluated. In the proposed model, there are two Conv1D layers in the dense block 1 and three Conv1D layers in the dense block 2. As shown in Appendix Table B.3, the mean square error is lowest when the kernel numbers of the Conv1D layers in the first residual block 1 and 2 are [32,64] and [32,64,64] respectively, meanwhile, kernel numbers of Conv1D layers in the second residual block 1 and 2 are [64,128] and [64,128,128] respectively. Moreover, the optimal kernel size of Conv1D layers in dense block 1 and 2 are [1,1] and [3,3,7] separately.

3) Units of BiLSTM and multi-head attention

After the proper tuning of the kernel size of the dense blocks, the hidden units of bidirectional LSTM layer, multi-head attention layers, and rate of dropout were set as 128, 256 and 0.5 separately, obtaining the lowest mean squared error.

Table 4.4: The optimal hyperparameters of the proposed model.

Hyperparameters	**Values**
Learning rate	0.001
DenseBlock 1 kernel number	1^{st} : $[32, 64]$ 2^{nd} : $[64, 128]$
DenseBlock 1 kernel size	1,1
DenseBlock 2 kernel number	1^{st} : $[32, 64, 64]$ 2^{nd} : $[64, 128, 128]$
DenseBlock 2 kernel size	3,3,7
BiLSTM units	128
Dropout rate	0.5
Attention units	256
Kernel regularizer.l1l2	[0.01,0.01]

4.4.2 Comparison of Model Performance to Different Model Structures

To compare the performance of the proposed model to others, results obtained here were compared with those obtained from multiple models with different network structures, which included (i) the plain CNN with attention-based BiLSTM; (ii) Plain CNN + LSTM; and (iii) Challenge-best deep neural network model.

i) Plain CNN + attention based BiLSTM

Appendix Table B.4 lists the architecture of plain CNNs and attention-based BiLSTM. Except for the structure of shortcut, the convolutional layers, batch normalization layers, and ReLu layers of this model are similar to those of the proposed model. Multiple dropout layers were added to this structure, which could reduce the complexity of coadaptation between hidden neurons and improve the robustness of the neural network [138].

ii) Plain CNN + LSTM

Similar to the plain CNN + attention-based BiLSTM model, the structure of the plain CNN + LSTM model contains plain CNNs without shortcut. Moreover, the attention-based BiLSTM is replaced by LSTM layers with a simpler structure for feature analysis.

iii) Challenge-best deep neural network model

Appendix Table B.5 depicts the structure of the first prize model [245] for the automatic diagnosis of cardiac abnormalities in the CPSC 2018 dataset. The model consists of five CNN blocks and attention-based bidirectional GRU. Each block includes two convolutional layers, with one pooling layer appended for reducing the over-fitting and the amount of computation. To achieve optimal performance of classification, the bidirectional GRU layer followed by an attention layer is connected to the last convolutional block. Moreover, the hyperparameters (i.e. kernel number/size) of the challenge-best model have been modified based on our proposed model, enabling a direct comparison.

Figure 4.6 plots computed F1 scores achieved by the proposed model, which are compared with results from other comparable models using the same dataset. As shown

in the figure, the F scores of six labels in the proposed model are notably higher than others. The proposed model achieved the highest F score of 0.965 for the RBBB case, followed by 0.959 and 0.958 for AF and LBBB, respectively. The probability results illustrated by the confusion matrix (Appendix Figure A.3) demonstrated a low probability of misclassification by our proposed model; especially, the probability of false positive and false negative for AF, LBBB, PVC, and RBBB is closed to zero.

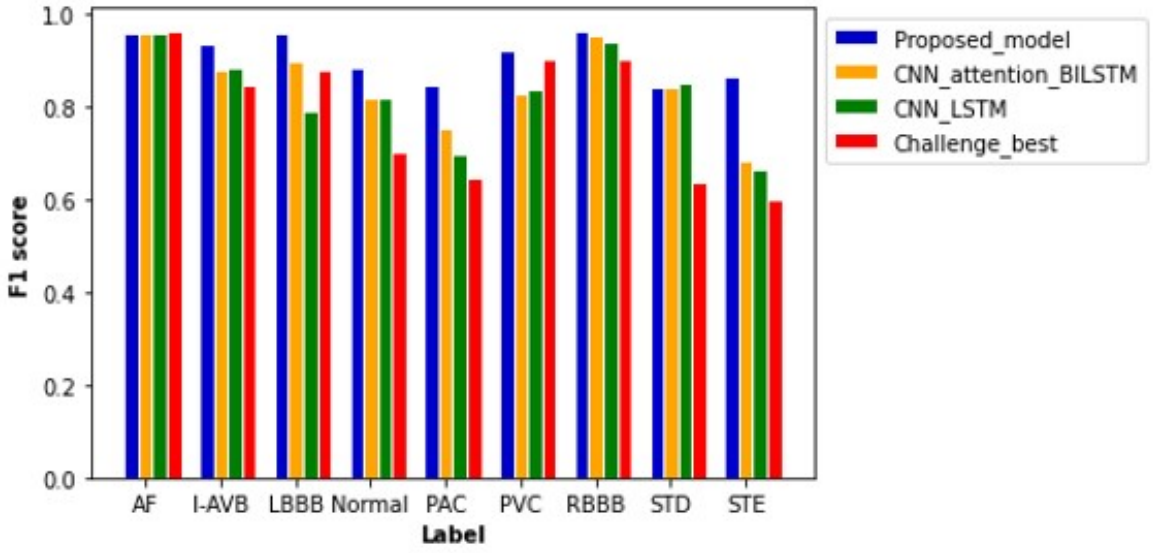

Figure 4.6: Comparison of F1 scores between different models based on the same test samples. F1 scores of the proposed model show the best performance of the model as compared with others with values of 0.959 for AF, 0.937 for an intrinsic paroxysmal atrioventricular block (I-AVB), 0.958 for LBBB, 0.885 for Normal, 0.848 for PAC, 0.920 for PVC, 0.965 for RBBB, 0.841 for STD, and 0.868 for STE. Specific values of F1 scores of the other three models are shown in Appendix Table B.6.

The plain CNN with attention-based BiLSTM ranks second with an average F score of 0.846. The computed F scores from the model for PAC, PVC, and STE are much smaller than those of the proposed model. Thus, the replacement of residual networks reduced the performance of the model. The performance of plain CNN with the LSTM model is not optimal for each type of cardiac abnormalities, especially for the cases of LBBB, PAC, and STE, for which F score is <0.800.

Although the Challenge-best model achieved the highest F score for the AF case, its performance for other abnormalities is relatively poor. Over-fitting occurred when the Challenge-best model was implemented for the data input of PAC, STD, and STE, leading to undesired F scores. Though the architecture and hyperparameters of the

Challenge-best model are similar to the model shown in Appendix Table B.4, the computed average F score of the challenge-best model is much lower as compared with the presented model.

Appendix Figure A.4 shows the computed ROC curve from different models for each type of the nine cardiac states. Comparing with other models, the ROC curve of the proposed model is closer to the top left corner, with an averaged AUC at 0.974, suggesting out-performance to the other models.

4.4.3 Performance on Different Preprocessing

To illustrate the advantage of the frame blocking for pretreatment of the data, the performance of the proposed model was compared with that using a common preprocessing method [273–275], which uses direct cutting and zero-padding protocol to unify the length of ECG signals. As for a fixed length of 40-s ECG data (i.e., 20,000 sampling data points), the common method can either truncate the exceeding signal samples when the length of original records exceeds 40 s or pads zeros to the data when the length is <40 s.

As shown in Figure 4.7, the higher median and minimum of the common method illustrate an improved model performance of the proposed frame blocking method. Moreover, the distribution of F1 scores by the common method is discrete, reflecting the instability of the performance of the classification model. To further evaluate the significant difference of this observation, the Wilcoxon signed-rank test is done on the two paired of F1scores. The p-value is 0.028 (<0.05), revealing the difference between F1scores produced by two pretreatments is significant.

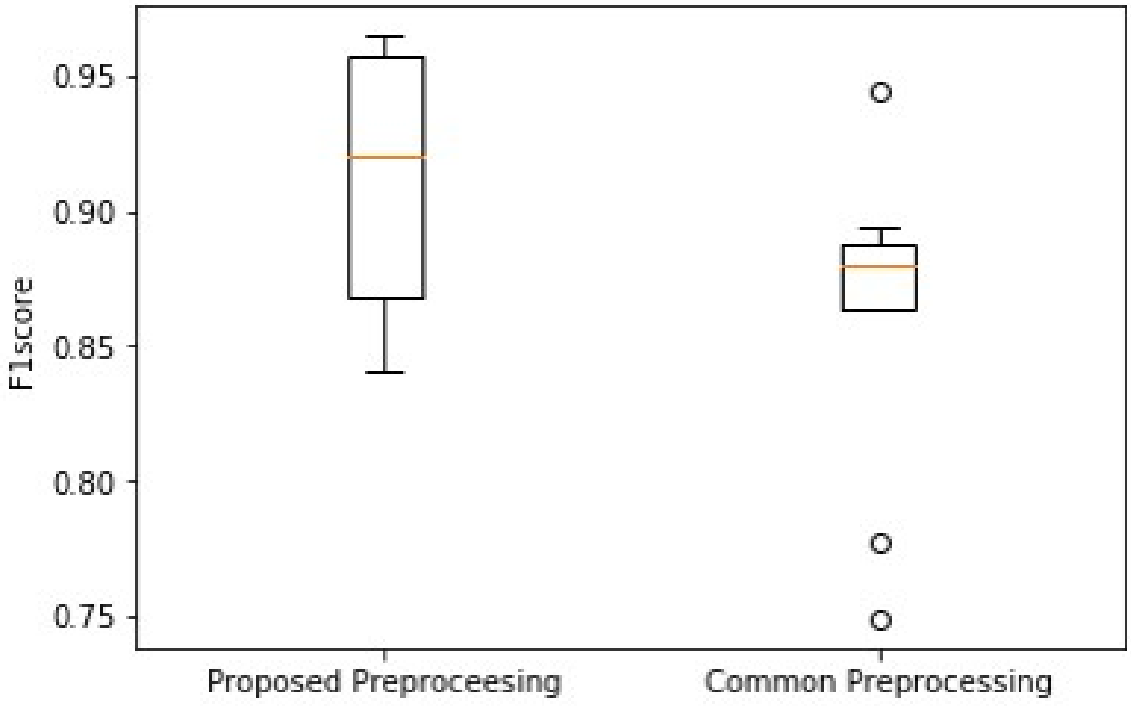

Figure 4.7: Comparison of overall F1 scores between using the proposed block framing and the common padding method for pre-processing ECG data for classification.

4.4.4 Robustness Testing

Being tested on the CPSC 2020 dataset, the proposed model shows F1 scores significantly higher than those of the Challenge-best model [245] for all seven types of cardiac arrhythmias (Figure 4.8). The computed ROC curve and AUC (shown in Appendix Figure A.5) also demonstrate the better performance of the proposed model (with an averaged AUC of 0.951) than the challenge-best model [245]. It is interesting to note that the Challenge-best model is much harder to converge on the CPCS 2020 than those of CPSC 2018. Also, the performance of the challenge-best model varies dramatically for different types of cardiac abnormalities with the use of the CPSC 2020 dataset as indicated by low values of F1 score for LBBB, Normal, PAC, and PVC conditions.

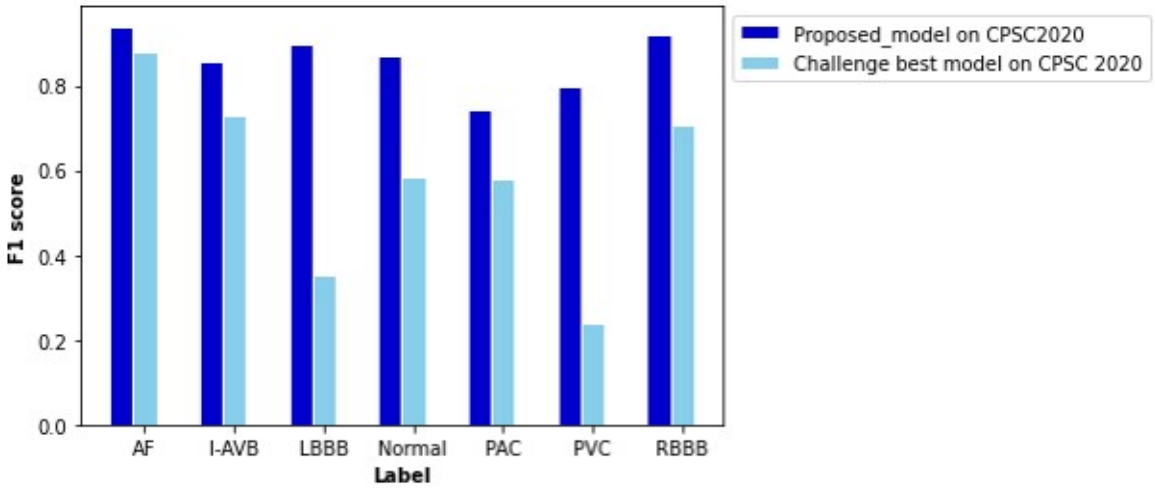

Figure 4.8: Comparison of performance between the proposed model and the Challenge-best model tested on the CPSC 2020 dataset for various types of arrhythmias. F scores of the proposed model are 0.940 for AF, 0.856 for intrinsic paroxysmal atrioventricular block (I-AVB), 0.898 for LBBB, 0.870 for Normal, 0.743 for PAC, 0.798 for PVC, 0.922 for RBBB, 0.841 for STD, and 0.868 for STE. Comparison of the F1 score between them is listed in Appendix Table B.7.

4.4.5 Cross-validation

Besides the CPSC datasets, the PTB XL dataset was adapted for cross-validation of the proposed novel algorithm for preprocessing and classification. As shown in Figure 4.9, the F1 scores of four diagnosis labels are higher than 0.800, achieving an average F1 score of 0.838 for all diagnosis labels in that dataset. The computed ROC curve and AUC (shown in Appendix Figure A.6) also illustrated a satisfying performance of the proposed algorithm on an external dataset with an average AUC of 0.950.

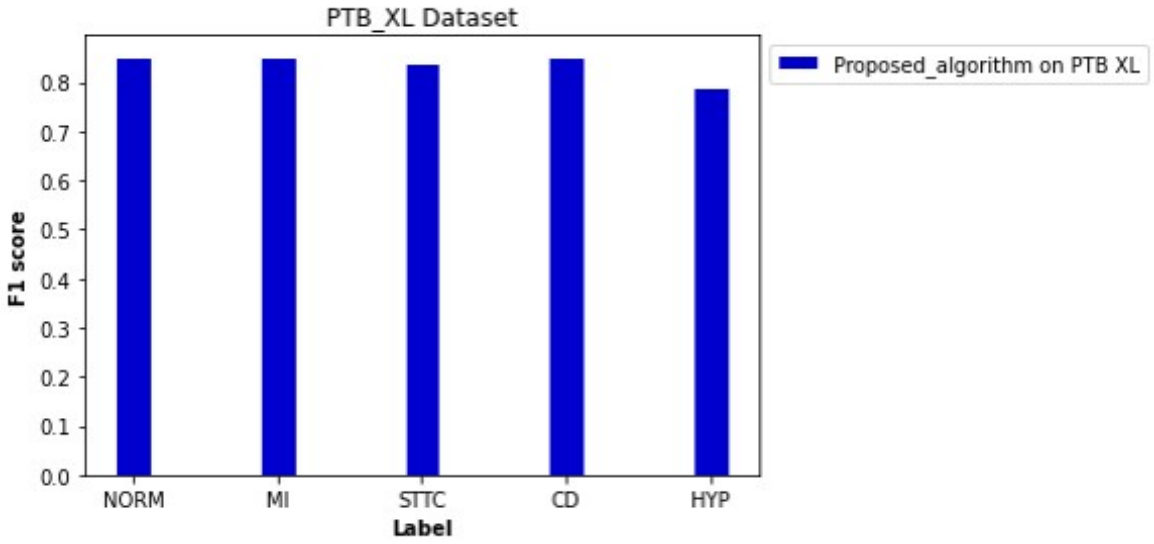

Figure 4.9: Performance of the proposed algorithm on PTB XL dataset for five diagnosis labels. F scores of each label are 0.853 for NORM, 0.852 for MI, 0.842 for STTC, 0.853 for CD, and 0.791 for HYP.

4.5 Discussion

The novelty and major contributions of the present study are the following:

1) Researcher proposed a preprocessing algorithm of frame blocking adapted from speech recognition, which decomposes ECG signals into overlapped frames. The proposed frame blocking method minimizes the loss of valid signals while maintaining the continuity of ECG signals in the process of unifying the length of variant ECG recordings.

2) Researcher developed a neural network based on the residual networks [276] with attention-based BiLSTM. As compared with the previous algorithms mentioned earlier for ECG detection, the presented network can extract and analyze ECG features automatically, thereby improving the model performance. It also alleviates the vanishing and exploding gradient problem as seen in deep neural networks

3) By training and testing the model using three independent datasets of 12-lead ECG signals provided in CPSC [277] and PTB XL [278], the proposed algorithm demonstrates superiority and robustness in classifying 12-lead ECGs with multi-labeling.

In recent years, numerous automatic detection methods for ECG analysis and classification have been developed. These methods are mainly based on and tested using the open-source MIT-BIH database [279], which are mainly single lead ECGs with single labeling. Thus, the general applicability of these algorithms for automatic stratifying multi-leads ECG and multiple types of arrhythmias is unclear. In this study, researcher developed a new algorithm based on frame blocking and the structure of ResNet, in combination with attention-based BiLSTM.

Initially, the novel algorithm was trained and evaluated on the datasets of CSPC for classifying 12-lead ECG for nine types of arrhythmia labeling. By comparing the performance of other model structures (Figure 3.6), the superiority of the proposed model was confirmed. Comparing with the common preprocessing method (Figure 3.7), the frame blocking method reduces the number of zeroes padded at the end of the signal recording, enhancing the valid part of ECGs, as well as the autocorrelation of ECG records. Thus, the proposed preprocessing method is more conducive to feature extraction for further classification.

The proposed algorithm demonstrated its robustness and clinical value via robustness testing and cross-validation. Through the robustness testing, the proposed algorithm shows a consistent performance on the two datasets and various types of abnormalities, illustrating the robustness of the proposed algorithm and hyperparameters. Considering the cross-validation, both the frame blocking method and classification model are also applicable to the PTB XL dataset with a vast number of clinical records.

Regarding the model structure, the proposed model adopts a similar neural network structure as the Challenge-best model. Both are based on a bidirectional recurrent neural network with a attention mechanism, but the proposed model used residual networks to avoid gradient explosion and vanishing. The strength of ResNet has also been demonstrated by several studies [246, 276]. In their studies, He et al. [246] showed the deep residual networks achieved an overall F1 score of 0.806. Rajpurkar et al. [276] utilized a 34-layer residual neural network to classify 65,000 multi-lead ECG records with 14 classes of cardiac disease and achieved an average accuracy and F1

score of 0.800 and 0.776, respectively.

Due to the differences between the original dataset and preprocessing method, the crosswise comparison of classification models is not persuasive. The studies mentioned earlier processed the classification via complex network structures and a large amount of annotated data. Although the deeper neural networks with sufficient training data achieved high classification accuracy, the computation of the model also increased and required expensive hardware support. Our model adapted a similar structure as the studies mentioned earlier but simplified the network structure, raising the computational efficiency of training and the probability of clinical practice.

As for the challenge-best model proposed by Chen et al. [245], its whole structure contains 10 plain convolutional layers and 5 pooling layers. The use of unnecessary multiple layers in the CNN layers may reduce the model performance on a small and unbalanced dataset due to over-fitting, causing difficulties in parameter tuning. Thus, the occurring of the internal covariate shift slows down the training process when the input distribution changes, impairing the convergence ability of the model. Different preprocessing methods may also affect the performance of the model. Compared with the commonly used method, the frame blocking method used in this study demonstrated its advantage in retaining maximum valid cardiac signal, which contributed to signal enhancement. Therefore, it proved to be a feasible preprocessing method to help the model extract more available features that are useful for model classification.

As for algorithms for multi-label classification [280], they can fall into problem transformation and algorithm adaption. With the development of neural networks, more studies [246, 259, 276, 281] designed an adaptive algorithm for multi-label classification. However, algorithm adaption has a high demand for sufficient training data and effective parameter adjustment to reduce misdiagnosis for multi-labeled ECG. Additionally, algorithm adaptation requires a complex model with proper parameters, increasing training cost and difficulties in data interpretation. In this study, each abnormality is considered as an independent binary problem, improving the interpretation of the features extracted. Although the binary relevance method cannot provide

information about label correlation and interdependence directly, it still demonstrated some advantages for multi-label classifying performance and efficiency.

Regarding several recent studies [282–284], the risk stratification is in high demand to prevent sudden death or stroke caused by cardiac diseases. Inspired by the present algorithm, the risk prediction of cardiac diseases can be automated based on the clinical data collected from the ECG or electronic heart records. The shortcut connection in the residual network saved the computing time of the model and accelerated the convergence of the model, which is friendly to the clinical research setting. Thus, the model has the potential to automatically identify the patients at a high risk of cardiac diseases, process early clinical interventions and therapy.

Furthermore, the application of a warning system of cardiac arrhythmias can be implemented based on the risk stratification and auto-detected algorithm. The ECG and electronic heart records can be stored and processed via cloud infrastructure and the internet, realizing the real-time monitoring system for cardiac arrhythmias and improving the early warning for the patients suffered from cardiac diseases.

4.6 Limitation of Study

There are a few potential limitations in this study. Firstly, random under-sampling was used to address the imbalanced datasets of MIT-BIH [279], PTB XL [278], and CPSCs 2018 [277] and 2020. However, some potentially important and information-rich data might be discarded from the majority class, causing difficulties in fitting the decision boundary between majority and minority samples [268]. Although the proposed model demonstrated good performance on two CPSC datasets (2018 and 2020) and PTB XL for 9, 7, or 5 different rhythmic abnormalities, it still needs to be further tested and improved by using other ECG datasets with more types of rhythmic abnormalities. However, as the types of rhythmic abnormalities increase, it would be expected that the required training time and GPU memory usage will be substantially increased.

In addition, the proposed neural network algorithm is heavily dependent on a large amount of annotated training data, which is labor expensive. For some rare types of cardiac abnormalities, it is difficult to collect such a large ECG dataset with annotation. In following-up works, it warrants to study further how algorithm adaption method [285] and other neural network architectures [42, 286, 287] help to deal with multi-labeled data directly and reduce time-demand for training. Moreover, unsupervised and semi-supervised learning can also be tested for addressing the lack of enough annotations.

4.7 Conclusion

This study proposed a new framing preprocessing method that can minimize the loss of ECG signals to enhance the features of signals. The proposed algorithm can diagnose multiple types of cardiac arrhythmias with promising accuracy, clinical value, and robustness, which may be potentially useful in assisting risk stratification, clinical diagnosis, and real-time ECG monitoring. Furthermore, researcher has shown that the residual neural network helps to extract deep features while saving computing time via processing the convolutional layers in parallel. For feature analysis, the attention-based BiLSTM demonstrated its advantage in addressing problems of long-distance dependency, allowing focus on the most significant features based on the assigned attention values.

Chapter 5

Fusing Deep Metric Learning with KNN for 12-lead Multi-labeled ECG Classification

5.1 Introduction

Cardiovascular diseases (CVD) with various complications are the leading cause of worldwide mortality [288]. Early diagnosis and prevention of cardiac abnormalities is essential for timely treatment and averting worse consequences. As a popular approach for detecting cardiac abnormalities, the electrocardiogram (ECG) provides sufficient information that indicate the electrical activity and dysfunction of the heart [289]. However, the interpretation of ECG for diagnosing cardiac dysfunctions is empirical and subjective, relying on the experience of the ECG reader. It is a challenge to develop algorithms for auto-analysing ECG signals.

In previous studies, some traditional auto-detection algorithms [290–292] have been developed to manually extract physiological features of ECG using time-frequency analysis. Recently, the deep learning methods provided a powerful alternative to the traditional auto-detection methods for more efficient ECG diagnosis. To date, a variety of deep learning approaches [225, 241, 256] have been tried for automatic ECG classification. While traditional auto-detection methods rely on manually extracted features or linear analysis of ECGs, deep learning algorithms concentrate on automatic extraction and analysis of ECG features, allowing for fewer false positives and missing features as a result of the selection of the analysing methods.

Various algorithms based on the deep network structure for ECG automatic classification have been developed [293, 294]. Chen et al. [40] combined the convolutional neural network (CNN) with long short-term memory networks(LSTM), achieving 99.32% accuracy on multi-class ECG classification based on MIT-BIH database. He et al. [246] further proposed a combination of deep residual network and bidirectional LSTM and achieved a overall F1 score of 0.806 on the test dataset from China Physiological Signal challenge Dataset.

In the study of [276], a 34-layer convolution-based residual network is used for 14 types of arrhythmias auto-detection, exceeding the performance of cardiologists. These studies achieved automatic feature extraction and good classification accuracy by mapping

the input ECG into specific types of abnormalities in end-to-end learning structures of deep neural networks (i.e., convolutional neural network, recurrent neural network). However, the robust deep-learning models and optimal classification accuracy proposed by these studies require a very deep CNN structure with massive computation. Additionally, the CNN-based feature extractor can only concentrate on the morphological features of the input signal while ignoring the temporal information.

Thus, challenges still remained for current deep learning methods to auto-detect 12-lead multi-labelled ECGs due to:

(i) Various heart abnormalities could present similar morphological features on ECG, thus multi-types of abnormalities are difficult to be classified accurately by a single model.

(ii) Algorithms employing deep convolution and recurrent structures requires large amounts of memory consumption of servers, impeding their applications in practice.

(iii) Deep neural networks merely extract morphological features of the input but ignore the temporal features on ECG, which impose some important information essential for ECG classification.

(iv) Most clinical ECG records have unequal lengths (Park et al. 2019), thus improper pre-processing may lead to loss of the important information embedded in ECG signals.

In this study, researcher developed an ECG classification approach based on the deep metric learning model to address some of the aforementioned issues. This method assesses the mapping from input ECG to features and reflects the similarity between input samples via the distance between features. Even though the deep metric model uses multiple convolutional layers, the addition of residual network structure [171] avails the backpropagation of gradients and accelerates the convergence of the model. The deep CNN with residual blocks dramatically reduces the memory consumption in the training process whilst maintaining satisfied classification accuracy [295]. As shown in previous studies [296–298], it is possible to use prior knowledge of ECG signals in classification tasks to avoid bottlenecks during model training.

Therefore, the first main contribution of this study is at that the proposed model combined the advantages of residual network and distance metric learning, processing the feature extraction more efficiently. Another key benefit of the study is that researcher concatenated the morphology features and temporal features of RR intervals from ECGs to further improve the classification accuracy. Additionally, employing frame blocking for the segmentation and length unification of raw ECG recordings, researcher implemented an novel pre-processing method inspired by speech recognition[265] to unify the length of raw ECG records.

5.2 Methodology

The pipeline of our proposed algorithm for 12-lead ECG classification is shown in Figure 5.1. Firstly, the raw ECG records are pre-processed via signal denoising and frame blocking. All ECG records are formatted as frames with uniformed size and then transmitted into a deep metric learning model (encoder) to produce features embeddings. The temporal features extracted from RR intervals concatenates with features embedded, based on which the Supervised Nearest Neighbours (KNN) algorithm [299]was finally utilized as a decoder to processed classification. Details of each of the process will be elaborated in the following subsections.

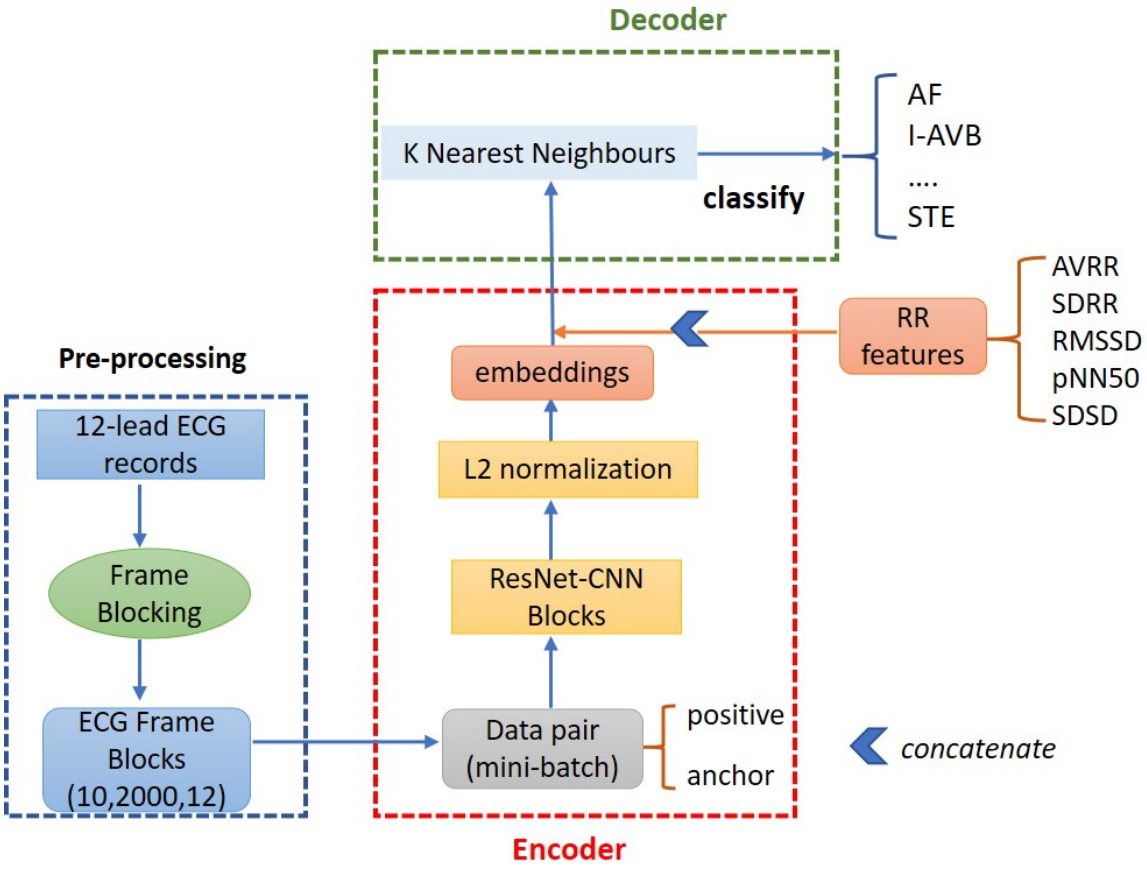

Figure 5.1: Flow chart diagram of the proposed algorithm for classifying 12-lead ECGs with multiple types of heart abnormalities (e.g., Atrial Fibrillation (AF), First-degree atrioventricular block (I-AVB) and ST-segment elevated (STE)). There are five time-domain measurements selected based on the standard of heart rhythm variability: Mean of RR intervals (AVRR), standard deviation of RR intervals (SDRR), root mean square of successive difference between RR-intervals (RMSSD), percentage of adjacent RR intervals that differ by more than 50ms (PNN50) and Standard deviation of the differences between successive RR intervals (SDSD).

5.2.1 ECG Datasets

China Physiological Signal Challenge 2018 (CPSC 2018)

China Physiological Signal Challenge in 2018 [277] comprises 6,877 12-leads ECG recordings collected from 11 hospitals. The sampling rate of all ECG recordings is 500Hz while each recording was collected with different duration lengths, ranging from 6 to 60 seconds. Table 5.1 shows the classes and the recordings number of each cardiac states in the CPSC2018 dataset. Furthermore, Appendix Figure A.1 shows the lead II ECG waveform of 9 types of cardiac states to illustrate their morphological profiles.

Table 5.1: Data profile of ECG recordings for 9 types of heart states in CPSC2018

ECG Type	**Number of records**
Atrial Fibrillation(AF)	1098
First-degree atrioventricular block (I-AVB)	704
Left bundle branch block(LBBB)	207
Normal(N)	918
Premature atrial contraction(PAC)	556
Premature ventricular contraction (PVC)	672
Right bundle branch block (RBBB)	1695
ST-segment depression (STD)	825
ST-segment elevated (STE)	202
Total	6877

PTB XL Dataset

To cross-validate the developed deep metric model to demonstrate its robustness and cross-applicability, the PTB XL dataset was used for testing. The PTB XL dataset contains 21,837 12-lead ECG recordings collected from 18,885 (male: 9,820, female: 9,064) patients with the sampling rate of 500Hz. For all of the recordings, two cardiologists manually classified and annotated them based on the SCP-ECG standard [300]. Table 5.2 lists the recording number for 5 different cardiac conditions, where the diagnostics statements are aggregated into 5 different superclasses.

Table 5.2: Records numbers and distribution of 5 types of diagnostic labels (superclass) in PTB XL.

Superclass	Description	Record Num
Norm	Normal ECG	9,528
MI	Myocardial Infarction	5,486
STTC	ST/T Change	5,250
CD	Conduction Disturbance	4,907
HYP	Hypertrophy	2,655

PTB diagnostic dataset

To further evaluate the generalization ability of the proposed model, the PTB diagnostic dataset was also adopted. The PTB diagnostic dataset includes 549 ECG recordings collected from 290 subjects (male: 209, female: 81), with various duration of around 2 minutes. Each ECG record contains 15 simultaneously measured signals (12-leads and 3 Frank leads ECGs), where each signal has the sampling rate of 1000 Hz. The diagnostic classes with corresponding recording number are summarized in Table 5.3.

Table 5.3: Records numbers of 8 types of diagnostic classes in PTB diagnostic database.

classes	Records
Myocardial infarction	368
Healthy control	80
Bundle branch block	17
Cardiomyopathy/Heart Failure	20
Dysrhythmia	16
Valvular heart disease	6
Myocardial hypertrophy	7
Others	35

5.2.2 Pre-processing

Denoising

The ECG is commonly contaminated by various noise from the source of base-line wander, power-line interface and muscle movements, resulting in blurred features extracted from ECG [301]. To minimize the negative effects of noise for auto-classification, an 8 order Butterworth low pass filter (cut-off 35Hz) was applied for eliminating the noises following the study of Qaisar and Dallet [302].

Frame blocking

The clinical ECG recordings with diverse length are not conducive to the training and testing of the classification model. To unify the length of input ECG recordings, a frame blocking method inspired by signal segmentation in speech recognition [303, 304] was proposed for pre-processing. Figure 5.2 demonstrates the implementation of the frame blocking on the clinical ECG signal. The raw ECG signals was divided into several blocks which consist of frames with a constant frame length F_l and an overlap f_o and a frame hop F_s. The frame hop shown in Figure 5.2 can be expressed as:

$$F_s = F_l - f_o \tag{5.1}$$

Furthermore, the overlap between frames of M can be mathematically formulated below, where the input signal length of S_l, number of frames N and frame length F_l are given.

$$f_o = \frac{N \cdot F_l - S_l}{N - 1} \tag{5.2}$$

Occasionally the signals in the last frame were padded by zero so that all samples in frames have the same length. Thus, the number of the samples for padding in the last frame R can be expressed as:

$$R = F_l - [(S_l - f_o) mod (F_l - f_o)] \tag{5.3}$$

The ECG recordings in the CPSC dataset have various length ranging from 6 to 60 seconds, and the length of most recordings is shorter than 40 seconds (i.e., 20,000 sampling points). Considering the retention of available ECG signals, the F_l and N were fixed values that are set to 2,000 and 10 respectively, but H and M are variable

depending on the input signal length. Thus, a set of 12-lead ECG recordings were formatted as frame blocks, each of which had a size of (10,2000,12) by frame blocking.

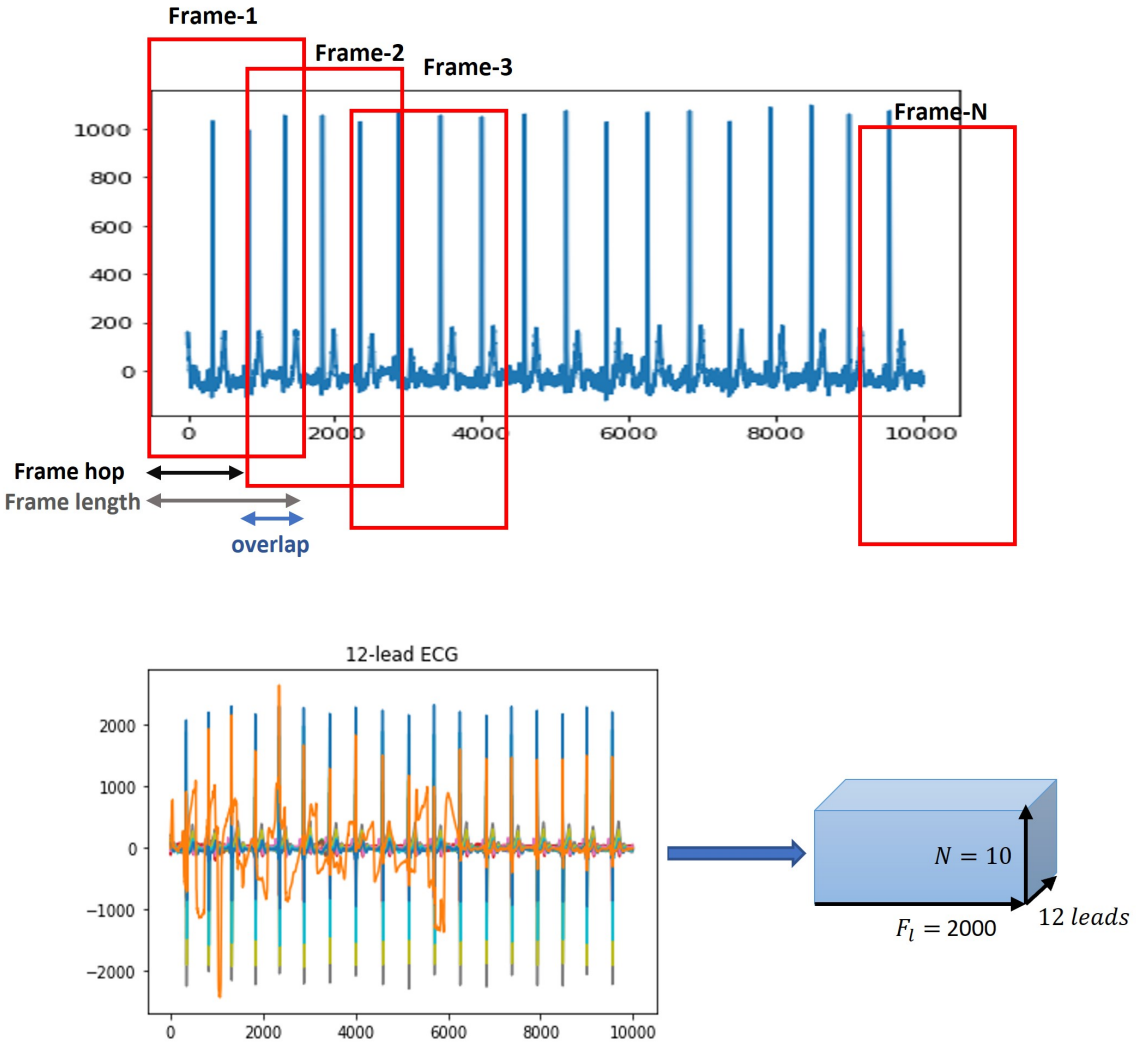

Figure 5.2: Schematic diagram of frame blocking for pre-treating raw ECGs.

Balanced Dataset

The class imbalanced problem commonly occurs in clinical datasets, leading to the bias and skewness in classification problem. In our study, researcher employed Synthetic Minority Over-sampling Technique (SMOTE) [305] to address the dataset class imbalance. SMOTE, as an oversampling technique, aiming to generate synthetic samples for solving the imbalanced dataset. Researcher used SMOTE to oversample the minority classes to have the same size of the majority class with the most samples. In our experiment, 20 percent of real samples are adopted as test set and the remaining 80 percent samples are regraded as training set, then using SMOTE for data augmentation.

5.2.3 Model Details

Encoder with Residual Convolution Blocks

In this study, The encoder model was used for feature extraction. The structure of the encoder is inspired by ResNet[171, 173] due to its outstanding performance on image recognition and classification. Similar to the previous project, two types of dense blocks also originates from the structure of ResNet. Though the deeper neural network is conducive to extracting ECG features more precisely, however, it may result in overfitting and vanishing or explosion gradient. Therefore, researcher chooses to use the residual network that provides a smoother approach for information processing during forward and back propagation, accelerating network convergence and addressing the shattering gradient problem.

An encoder with two-dimension residual blocks that involved 11 Convolutional layers to process feature extraction was developed. Figure 5.3 shows the general structure of the encoder network ((a)) and the consisting two types of residual blocks ((b): A and B).

Both two types of blocks involved the use of two-dimension convolution layers (Conv2D), and batch normalization (BN) and rectified linear units (ReLu) as activation function. A Conv2D and BN were involved in the shortcut of the residual block B to adjust the input dimension to match the output dimension, however, a single shortcut in

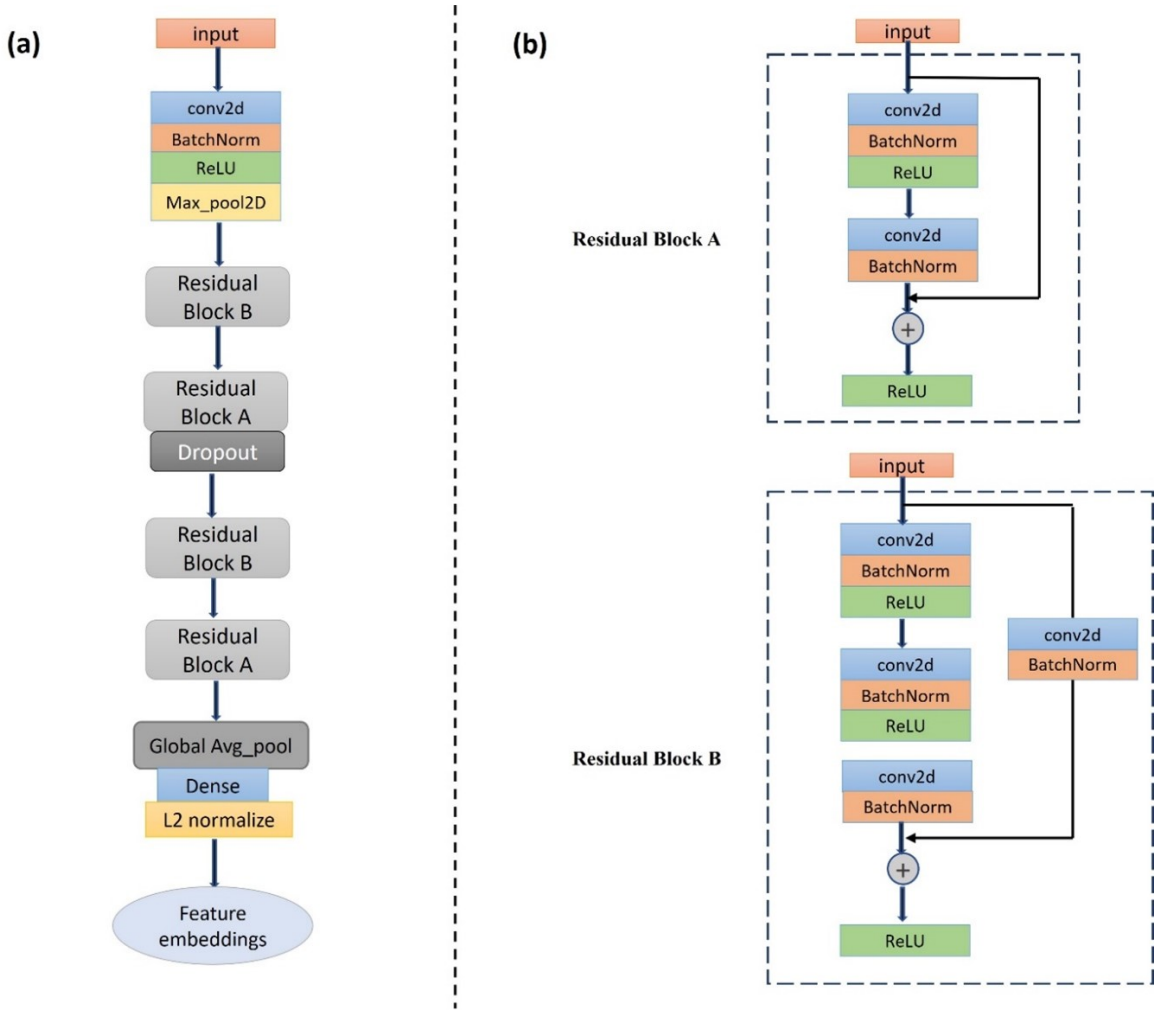

Figure 5.3: (a) Illustration of the structure of full encoder architecture. (b) Illustration of two types of residual blocks (Residual Block A, Residual Block B) for the encoder network.

residual block A can deliver the input to the following layer directly, reducing computational complexity. Following the last residual block are the global average pooling layer (GAP) [271] , reducing the spatial dimension of the output and generating a feature map according to each corresponding abnormalities. The last dense layer and L2 normalization are used to generate the feature embeddings for the corresponding input record. Furthermore, the configuration and parameters of all CNN layers in the residual network can be found in Table 5.4.

Table 5.4: Parameters details for all layers in encoder

Layers	**Output shape**	**Kernel, Strides**
Input1	(Batch, 10,2000,12)	
conv2d1	(Batch, 5,1000,32)	3×3×32, 2
Maxpool2d	(Batch,3,500,32)	3×3, 2
Conv2d2	(Batch, 2 ,250,32)	3×3×32, 2
Conv2d3	(Batch, 2,250,64)	3×3×64, 1
Conv2d4 (shortcut)	(Batch,2,250,64)	7×7×64, 2
Conv2d5	(Batch, 2,250,32)	1×1×32, 1
Conv2d6	(Batch,2,250,64)	1×1×64, 1
Conv2d7	(Batch,1,125,64)	3×3×64, 2
Conv2d8	(Batch,1,125,128)	3×3×128, 1
Conv2d9(shortcut)	(Batch,1,125,128)	7×7×128, 2
Conv2d10	(Batch,1,125,64)	3×3×64, 1
Conv2d11	(Batch,1,125,128)	3×3×128, 1
Globavgpool2d	(Batch,128)	
dense	(Batch,9)	9
L2norm	(Batch,9)	

Feature Fusion

It has been shown in previous studies [306, 307]the fusions of prior information and extracted features can effectively extend the margin of different classes, contributing to improved classification accuracy. For electrophysiological signals, it is insufficient to process the classification relying solely on morphological features. Thus, the features of RR intervals were analysed for representing rhythm information of ECG signal and then merged with morphological features.

Figure 5.4 shows a boxplot of RR intervals of the nine types of cardiac states in the CPSC dataset. Through the middle line and interquartile ranges of the box, the nine types of heart states can be distinguished, especially AF, I-AVB, Normal and STE. According to a guideline for measurement standard of heart rhythm variability (Shaffer Ginsberg, 2017), researcher chooses several time-domain measures as signal prior information based on RR intervals. The details of the measures have shown in Table 5.5.

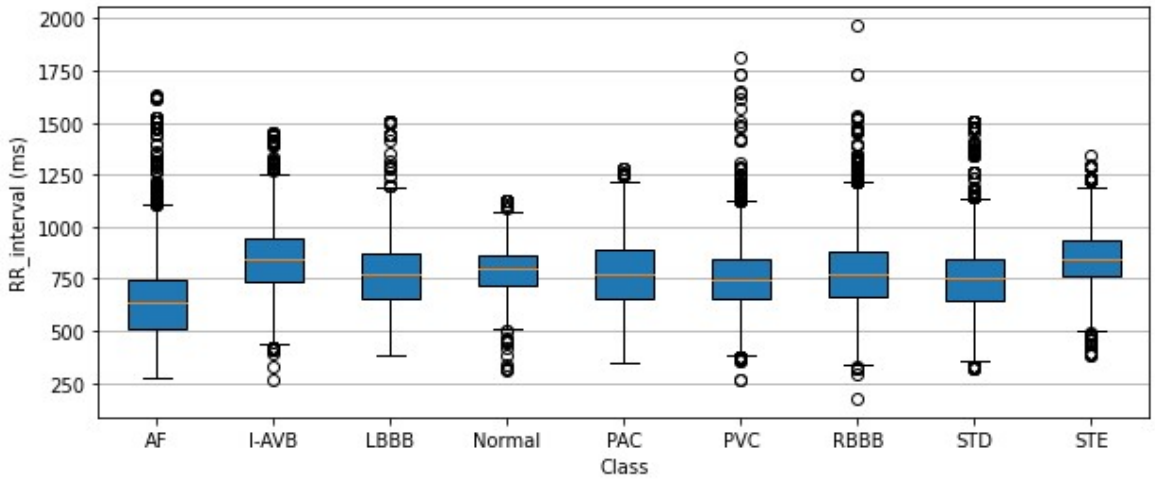

Figure 5.4: Boxplot for RR intervals of the 9 types of heart state of ECG records in CPSC2018. The middle line and black circle represent the middle value and outliers of RR intervals of that class, respectively. The interquartile range (blue) of box indicates where the RR intervals lie.

Table 5.5: Profile of five time-domain measures bases on RR intervals

Measures	**Description**
AVRR	Mean of RR-intervals
SDRR	Standard deviation of RR-intervals
RMSSD	Root mean square of successive difference between RR-intervals
pNN50	Percentage of adjacent RR intervals that differ by more than 50ms
SDSD	Standard deviation of the differences between successive RR intervals

Metric Learning

Metric learning firstly aims to map the input data to feature embeddings and then proposes a measure based on the similarity between feature embeddings, bringing the similar data closer and extend the margin between different data. In our study, the deep metric learning utilized the deep learning model as feature extractor, automatically learning discriminative features via mini-batch processing and pair-based training procedure.

Researcher adopted (anchor, positive) pairs to express the similarity where each pair consists of an anchor sample (randomly selected sample) and a positive sample (random selected sample from the same class). In training procedure, each mini batch contained C pairs of data $((A_i, P_i), i \in C)$ where the C denoted the number of classes. The residual convolution network worked as a feature encoder $f_{:\ R^D \to R^d}$, generating feature embeddings of pairs of samples $(f_\theta(A_i), f_\theta(P_i))$. The pairwise dot product was utilized for measuring the similarity S between anchor embeddings and positive embeddings with the following equation:

$$S_{a,p} = \sum_e f_\theta(A_i)_{a,e} f_\theta(P_i)_{p,e} \tag{5.4}$$

where a and p represent the row numbers for anchor embeddings and positive embeddings, and e denoted the number of columns of feature embeddings. Based on the sparse labels and the similarity of embeddings, SparseCategoricalCrossentropy loss function [308] was employed to minimize the distance between the embeddings for

anchor/positive pairs, as well as to separate the pairs from other different classes. The calculation of the loss in a batch can be written as:

$$L_{SCCE} = -\sum_{c=1}^{c} y_c log(S_{a,p}) \tag{5.5}$$

where y_c is the sparse labels and C is classes number. The training algorithm of deep metric learning model with fused features is given below.

Algorithm 2 Deep metric learning model with fused features. Batch size C,number classes M and number of images per classes N, RR represent the ECG temporal features extracted from RR intervals

Input: Training Dataset $D = \{(x_1, y_1), (x_2, y_2), ..., (x_i, y_i)\}$. An embedded function f which represents the ResNet encoder, epoch n

Output: The loss L for the mini-batch training

Split the Training Dataset D into data pairs (A_i, P_i), each mini-batch involves C pairs of samples $(A_i, P_i) \in C$

for i in $1, ..., n$ epochs **do**

for j in $1, ..., C$ mini batches **do**

$fused(A_i) = [f_\theta(A_i), RR_i]$

$fused(P_i) = [f_\theta(P_i), RR_i]$ ▷ process concatenation for feature fusion

$S_{a,p} = \sum_e fused(A_i)_{a,e} fused(P_i)_{a,e}$ ▷ calculate similarity

$L \leftarrow L - \sum_{M=1}^{M} y_m log(S_{a,p})$ ▷ Loss update

end for

end for

5.3 Experiment Result and Evaluation

5.3.1 Evaluation Metrics

In this study, the widely used evaluation matrix (i.e., precision, recall, F1 score, confusion matrix), were used, which can be calculated by the following equations:

$$Precision_n = \frac{TP_n}{TP_n + FP_N} \tag{5.6}$$

$$Recall_n = \frac{TP_n}{TP_n + FN_N} \tag{5.7}$$

$$F1_n = \frac{2 * (Precision_n * Recall_n)}{Precision_n + Recall_n} \tag{5.8}$$

The variant n denotes the types of cardiac abnormalities. TP, FP, TN and FN represent the number of samples which are true positive, false positive, true negative and false negative separately. Besides the F1 score, the confusion matrix can also reflect the true and false prediction rates on a classification model.

5.3.2 Training Settings

The proposed algorithm was trained and tested on a server with Tesla P100 16GB GPU and Keras framework 2.4.3 [309]. Initially, the CPSC dataset was randomly divided into training and testing datasets with a ratio of 4:1, then utilized SMOTE to balance the training set. To further prevent overfitting, the remaining training data was further split into training data and validation data via 3-repeated 5-fold cross-validation and each fold contains 40 epochs for training and validation.

After pre-processing, the proposed model and other reference models will be tested on the test set to show the classification performance. In the training process, one mini-batch consists of multiple data pairs (anchor, positive) among 9 classes. Batch size is set to 200 based on hyperparameter tunning, requiring less memory occupation. Adam optimizer [131] with a learning rate of 1e-3 was applied for updating the weights of the residual network. Also, researcher sets the decay rate of the 1^st and 2^nd moment as default for fast convergence. HeNormal initializer and L2 kernel regularizer are employed to accelerate the net convergence, as well as preventing the overfitting.

5.3.3 Classification Performance

Experiment Result

The experiment was firstly conducted on the CPSC2018 dataset based on the training setting. Table 5.6 shows the comparison of the classification performance of our algorithm and other reference models on the test set split from 6,877 ECG recordings.

The F1 scores obtained here were compared to those obtained from other state-of-art models based on deep learning, which includes (i) CNNLSTM [244]; (ii) Residual CNN with bidirectional GRU and attention mechanism [310]; (iii) Challenge-best deep neural network [245]. The parameters and other details about these reference model can be found in the Appendix (Table B.8-Table B.10).

As shown in Table 5.6, the F scores of 7 classes in our proposed algorithm are higher than those of others, especially for Normal, PAC and STE. In the aspect of arrhythmias, the AF, LBBB and RBBB remain high classification performance among all the models probably due to the highly discriminative morphological features (Appendix Figure 206) and temporal features of these types of arrhythmias. The classification performance of STD and STE is not optimal since the abnormal waveform morphologies of the ST segments may correspond to various arrhythmias such as Left bundle branch block and left ventricular hypertrophy[70]. The performance of CNN with LSTM model ranks at the last; their F scores of PAC and STE are lower than 0.6, significantly smaller than those of others. The second reference model with residual network achieves the highest F score of 0.967 for the RBBB while the performance of PAC, Normal and STE is not optimal. Compared to the performance of the challenge best model in CPSC2018, our algorithm achieves an approximately 0.08 score increase. There was a large gap in the F score of the two models for the Normal and STD cases. Besides the F score, the classification report (Precision, Recall) for each class in CPSC dataset were calculated for the proposed model and presented in Appendix Table B.11. Appendix Figure A.7 shows the accuracy plot of the proposed model on training and validation sets from CPSC dataset. Both training and validation accuracy have similar uptrend, and then remain flat in the last 40 epochs.

Table 5.6: Classification performance for proposed algorithm and other reference models on the test set split from 6877 recordings from CPSC dataset

Class	**F1score**			
	CNN+LSTM[244]	ResidualCNN +BiGRU +Attention[310]	CNN +LSTM +Attention [245]	Our proposed
AF	0.906	0.865	0.916	0.904
I-AVB	0.802	0.739	0.819	0.879
LBBB	0.895	0.923	0.879	0.959
Normal	0.728	0.755	0.744	0.868
PAC	0.590	0.614	0.702	0.794
PVC	0.799	0.911	0.831	0.929
RBBB	0.847	0.967	0.903	0.899
STD	0.722	0.783	0.725	0.869
STE	0.567	0.681	0.667	0.773
Average	0.761	0.804	0.798	0.874

Visualization of Extracted Features

To evaluate our proposed model visually, the t-distributed stochastic neighbour embeddings (t-SNE) [311] were used to visualize the high dimensional feature embeddings into two-dimensional plots. The t-SNE first captures similarities between features embeddings in the initial high dimension, then clustering the feature samples in lower-dimensional space based on their similarities.

Figure 5.5 plots the t-SNE visualization for the performance of the proposed model; the points of different colours indicate 9 classes separately. In general, the colour-coded points of 9 classes are decentralized obviously, even for the small number of

STE and LBBB. A slight overlap of points is in the classification of Normal, I-AVB PAC and STE, and the overlap in distinguishing the rest of the classes is lesser. Thus, the features extracted from the proposed deep metric learning model are discriminable for classifying 9 types of heart states.

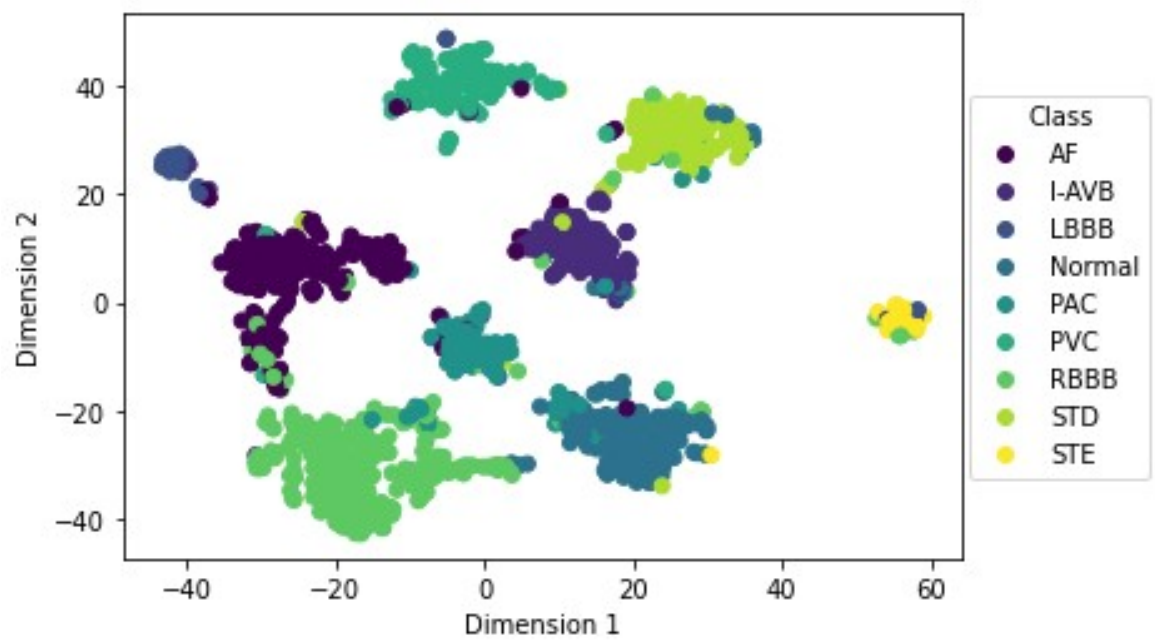

Figure 5.5: The t-SNE visualization for feature embeddings into two dimensional plots (dimension 1&2).

5.3.4 Contribution of features fusion

In this subsection, researcher processed the training and testing to explore the contribution of the fusion of temporal features from RR intervals and morphological features of signals, while keeping hyperparameters and model structures unchanged. Results shown in Appendix Table B.12 depicts the specific F1scores of 9 classes achieved by our proposed algorithm and that without adding the temporal features of RR intervals. Comparing to the model with temporal RR-features, the F1scores of 8 classes decreases to varying degrees and the average F1scores fell to 0.825 from 0.874.

As shown in Figure 5.6, the false positive and false negative instances of Normal, PAC and PVC in Figure 5.6 (a) are greater than those in Figure 5.6 (b). The rising error rate implies the importance of temporal RR features for classification accuracy; thus,

the absence of temporal RR features weakens the model performance.

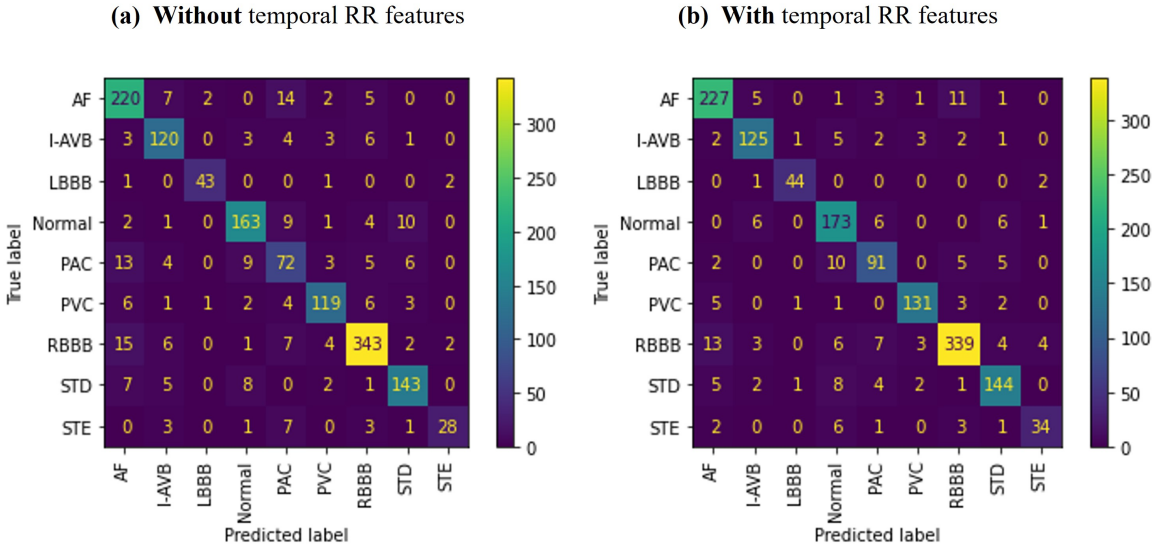

Figure 5.6: Confusion matrix for classification performance. (a) confusion matrix for the model without temporal RR features; (b) confusion matrix for the model with temporal RR features.

5.3.5 Cross-validation

In addition to the CPSC2018 dataset, researcher evaluated the classification performance of our proposed method on the and PTB XL dataset and PTB diagnostic dataset for cross-validation. 8,000 ECG recordings from PTB XL were randomly selected as experimental data, where the ratio of training and testing set was also 4:1. Table 5.7 depicts the classification performance of the proposed method on PTBXL dataset, where the average of all metrics is higher than 0.800. Specifically, three types of evaluation metrics of NORM achieved 0.927,0.939 and 0.915 respectively, ranking at the top.

Comparing to PTB XL dataset, the PTB diagnostic dataset belongs to small dataset with limited ECG recordings. To uniform the ECG signals during cross validation, researcher resampled the signal frequency of PTB diagnostic dataset to 500 Hz. Table 5.8 displays a classification report of the proposed method on PTB diagnostic dataset with same training setting as before. All evaluation metrics among all classes are higher

than 0.850, where the F1score of Dysrhythmia achieved 1.000, followed by that of Myocardial infarction (0.947) and Cardiomyopathy (0.923). Furthermore, two confusion matrices (shown in Supplemental Figure A.8) also showed satisfactory performances with low error rates of our proposed method on PTB XL and PTB diagnostic dataset.

Table 5.7: Classification report of cross validation on PTB XL dataset

Superclass	**F1 scores**	**Precision**	**Recall**
CD	0.806	0.771	0.844
HYP	0.820	0.792	0.849
MI	0.775	0.790	0.760
NORM	0.927	0.939	0.915
STTC	0.762	0.770	0.754
Average	0.818	0.812	0.824

Table 5.8: Classification report of cross validation on PTB Diagnostic dataset.

Superclass	**F1 scores**	**Precision**	**Recall**
Myocardial infarction	0.947	0.954	0.941
Healthy control	0.883	0.844	0.926
Bundle branch block	0.889	0.889	0.889
Cardiomyopathy/Heart Failure	0.923	0.857	1.000
Dysrhythmia	1.000	1.000	1.000
Valvular heart disease	0.889	1.000	0.800
Myocardial hypertrophy	0.875	1.000	0.778
Others	0.857	0.857	0.857
Average	0.902	0.925	0.898

5.4 Discussion

The major contribution and novelty of the proposed method are the followings:

(*i*) Inspired by computer vision image classification tasks [312–314], the developed model combines metric learning with residual network for better performance. As shown in Table 6, the F1 scores of the deep metric model is the highest among others reference models that merely rely on deep learning, highlighting its advantages in classification performance.

(*ii*) The frame-blocking worked as pre-processing method, reducing the loss of valid ECG signals during unifying the length of different ECG recordings.

(*iii*) Researcher proposed to concatenate the morphological features with temporal features extracted from RR intervals. As shown in Figure 6, the merge of temporal and morphological features helps to discriminate heart abnormalities more efficiently.

(*iv*) Advantages in classification of subtle morphological differences of the deep metric model was required as compared to other state-of-arts models based on deep neural networks.

(*v*) Our algorithm has been trained and tested on three independent datasets (CPSC2018, PTBXL and PTB diagnostic dataset). Through the cross-validation, the proposed deep metric model demonstrated its robustness and efficiency of auto-detection for 12-lead ECGs with multi-labelling.

As the experiment result showed, the deep metric model ranks at the top in classification performance. The other three reference models [244, 245, 310]used different test dataset in their research, thus their classification performances proposed in the original text is unable to compare. In our study, these reference models and our proposed model are tested on the unified test set for comparison. All three reference models fail to diagnose STD and STE may since the changes in ST-segment of these arrhythmias is subtle. In contrast, our algorithm shows improved performance in several classes (i.e., STD:0.869, STE: 0.773), thus claiming its advantages in classification under the high similarity between different classes.

Regarding the model structure, the deep metric model adopts the structure of ResNet to eliminate the gradient vanishing and explosion. The average F1 scores of the second reference model [310]are higher than the models with plain CNN, illustrating the advantages of ResNet in deep learning. Serval studies [44, 315, 316] also claim the strength of ResNet in ECG classification. Both Han Shi and Hao et al.(Han Shi, 2020; Hao et al. 2020) [315, 316] utilize the residual neural network to detect the myocardial infarction (MI) automatically on the 12-lead ECGs, achieving F1scores of 93.79% and 96.92% respectively. Furthermore, in their studies, Wang et al. [44] proposed a parallel GAN model for data augmentation and a classification model which is a very deep residual network, achieving the average F1 scores of 0.883 via training and testing on the CPSC 2018 hidden dataset. Above studies demonstrates the advantages of the residual networks in classification performance. As the classification model, residual neural network helps to extends the depth of the model and simultaneously maintains the classification performance.

Table 5.9: Comparison of computation complexity between the proposed algorithm and other deep learning models.

Model	**Inputsize**	**Model size(Params)**	**GFLOPs**
CNN +LSTM	(10,2000,12)	4,949,097	1.03
Challenge-best	(10,2000,12)	28,035	0.07
ResidualCNN+BiGRU+Attention	(10,2000,12)	5,594,569	1.16
ResNet50	(244,244,3)	23,587,712	3.91
VGG16	(244,244,3)	138,357,544	15.5
Our proposed	(10,2000,12)	800,169	0.33

As shown in Table 5.9, the model size and GFLOPs of ResNet50 is smaller than that of VGG16 with same input size. In comparison with plain CNN, the ResNet can effectively reduce the memory usage even with deeper ConvNets. Though the deeper ResNet networks show their advantages, their computational cost may hurdle practical industrial applications, especially with wearable devices. However, our proposed

method adopts a similar ResNet structure with relatively shallow CNN, combining with distance metric learning to preserve accuracy whilst reducing both the model size and GFLOPs.

As for the algorithms for feature fusion, some studies [317, 318]integrate morphological features with temporal features (i.e., RR intervals at the front and rear of a single heartbeat: [RRpre, RRpost]) to process the heartbeat classification. Inspired by the previous studies, researcher utilized and improved the feature fusion to adapt the classification on 12-lead ECG records. Using 5 temporal measures to represent continuous and long ECG rhythm, which is not confined to the RR intervals between two adjacent heartbeats. Although the extraction of RR features increase the time-consuming, the proposed feature fusion still achieved higher classification performance and illustrated the value of temporal features.

As mentioned in serval studies [319–321], deep metric learning has been widely used in image classification, face verification and recognition but rarely used in ECG signal classification. Compared with pure deep learning, deep metric learning is more efficient in these tasks that involve multiple classes but limited samples per class. In our study, deep metric learning can handle data pairs (i.e., [anchor, positive]) at the same network structure, thus increasing the amount of data training and partly averting the influence of insufficient samples.

Furthermore, the loss function can minimize the inner-class distance based on the similarity samples in data pairs and further optimize the model. As for classification, the deep metric learning model transforms input ECG records into corresponding feature embeddings, and then the supervised nearest neighbours classifier predicts the label of the test samples through majority voting. This KNN classifier takes simply distance measure without tuning parameters for model training. Moreover, KNN directly evaluates the distances between feature embeddings and performs well on the classes whose features are not evident, addressing the issue caused by the similar morphological features between different classes of records.

5.5 Limitation

There are a few limitations in our study. Firstly, manually extracting temporal features of RR intervals from ECG records increases time-consuming of training. Being different from FaceNet [321], our proposed deep metric model simply minimizes the inner-class distance while ignores the inter-class distance. Considering recent studies [322–324], triplet network and proxy loss might be conducive to improve classification effectiveness in following-up works.

Although the proposed model claimed a consistent performance on different datasets (CPSC2018 and PTBXL) and efficiently diagnosed 9 or 5 types of heart abnormalities, its performance still requires further evaluation and improvement using other ECG datasets with various types of heart abnormality. However, it is challenging to collect sufficient ECG records with professional annotations for rare types of abnormalities. Although deep metric learning proved its advantages, semi-supervised learning can be further investigating for addressing the shortage of ECG records and annotation. Additionally, using KNN classifier for classification stage is intuitive and simple, where as the test stage is slow and computationally inefficient. The extra time and computational cost should also be considered.

5.6 Conclusion

In this paper, researcher presented a deep metric model with feature fusion to screen multiple types of heart states on 12-lead ECGs. The proposed approach consists of three procedures for conducting abnormalities classification: pre-processing via frame blocking, feature extraction based on deep metric learning and KNN classification. Concerning the experiment results and cross-validation, our approach diagnosed efficiently multiple types of abnormalities with satisfying accuracy and robustness. Noteworthy to point out that the deep metric model with ResNet structure is able to extract discriminative features from various types of arrhythmias. Consequently, the residual neural network contributes to a lightweight deep metric model which has potential usage for real-time processing.

Chapter 6

Parallel Multi-scale Convolution based Prototypical Network for Few-shot ECG beats Classification

Study presented in this chapter was published in:

Z. Li and H. Zhang, Parallel Multi-scale convolution based prototypical network for few-shot ECG beats classification. *2022 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)* Ioannina, Greece, 2022, pp. 1-4

6.1 Introduction

Cardiac arrhythmias are a typical and serious group of cardiovascular diseases (CVD), leading to various complications and high mortality rates [325]. Early detection and proactive treatment of cardiac arrhythmia are extremely essential for averting worse consequences. The electrocardiogram (ECG) is a common clinical diagnosis approach for cardiac arrhythmia, recording the signal electrical activity of the heart [326]. Manual beat-by-beat ECG interpretation is time-consuming, subjective, and greatly dependent on the expertise of cardiologists in practise. Therefore, the automatic detection of ECG signals is crucial in improving the diagnostic efficiency of arrhythmia. Diverse deep-learning-based approaches, such as DCNNs [40, 231] and RNN [239], have recently achieved impressive progress in ECG auto-detection.

Baloglu et al. [231] used a 10-layer CNN to detect the myocardial infarction with 12-lead ECG signals, achieving an impressive accuracy of about 99%. In the study of Singh et al. [239], the long-short term memory (LSTM) directly extracted the information from one-dimensional ECG and showed a satisfying accuracy of 88.1% on a large number of ECG beats. The model proposed by Chen et al. [40] demonstrated the benefit of combining CNN and LSTM in signal abstraction by achieving 99.32% accuracy in multi-class ECG classification. To further improve the accuracy and address the imbalanced database, several state-of-art models are utilised in ECG classification. Rajpurjar et al. [276] proposed a 34-layer convolutional network based on the residual network structure, surpassing the performance of cardiologists in diagnosing 14 types of arrhythmias. Studies with Generative adversarial networks (GAN) [43, 327] illustrated the advantages of GANs in data augmentation and classification performance on the imbalanced MIT-BIH arrhythmia dataset. These studies utilized deep neural networks to perform automatic feature extraction and analysis on ECG signals, achieving high accuracy of binary or multi-class arrhythmia classification.

In these studies, the robust deep learning model and high classification accuracy are closely related to sufficient labelled instances. However, limited data is collected for some types of arrhythmias due to the rare heart abnormality and individual variation.

Furthermore, the models employing deep networks based on a huge amount of training dataset demand massive computation and memory consumption, which impeding their practical use. Thus, it is still challenging to detect arrhythmia automatically under insufficient ECG instances.

Few-shot learning (FSL) was presented along with the advancement of computer vision as a solution to the data scarcity. FSL aims to pre-train the model on a related training set and then uses the pre-trained model for auto-classification using a small number of training data [328]. Some metric-based FSL studies [329–331] modelled the distribution of the distance between samples to expand the inter-class distance and shrink the inner-class distance, showing the progress of the few-shot learning in image recognition. In addition, to improve the signal feature extraction, authors [332] employed the relation network with attention mechanism for EEG-based classification. They also highlight the correlation between the support set and query set, generalizing better on unseen classes. In [333] authors leveraged a ResNet as a baseline model for signal feature extraction and addressed multiple few-shot time-series classification tasks via meta-learning. For time-series classification, they proved that the model can be trained on few-shot tasks from diverse classes and generalized rapidly on unseen classes. Few-shot learning proved its potential in time-series classification while rarely used in arrhythmia detection based on ECG.

Inspired by the aforementioned studies, researcher employed few-shot learning in the automatic arrhythmia classification for arrhythmia auto-classification with limited ECG samples. Researcher proposed a parallel multi-scale CNN (PM-CNN) based prototypical network for arrhythmia classification, realizing a fusion of different scales of CNN and processing few-shot tasks based on the learning of metric space. In our study, researcher used wavelet transform to convert one-dimension signals to three-dimension RGB images in order to better signal feature extraction. Therefore, the multi-scale CNN can extract the comprehensive feature representation via convolutional layers with different kernel sizes. The fusion of the features contributes to better performance for unseen classes recognition than the conventional prototypical network.

6.2 Related works

Few-shot learning (FSL) is a branch of machine learning that can only use a small number of training samples to classifying the unseen data. Combined with the N-way K-shot training strategy of the meta-learning, FSL lessens the reliance on a large amount of labelled data and prevents the network from overfitting that is caused by very few training data. Like the learning of humans, few-shot learning utilizes the combination of both the few samples and prior knowledge obtained from pre-training. According to the different usage of the prior knowledge, deep learning-based FSL learning approaches can be broadly divided into three groups: models-based approaches, optimization-based approaches and similarity(metric)-based approaches.

Using the prior knowledge, Model-based approaches (i.e. Memory-Augmented Neural Networks (MANN)[334], Meta networks [335]) rely on the structure of deep learning models for fast generalization of FSL tasks. For instance, Meta networks consist of a base learner and a meta learner, learning the meta-level knowledge across tasks and realizing the rapid generalization. Meta networks introduce external memory in the meta learner that stores the generalized information produced by the base learner, learning model parameters (i.e. weights) in a fast approach. Additionally, gradient descent algorithms (optimizers) may fail to optimize the model parameters on a very small amount of dataset. To quickly optimize the parameters of deep neural, optimization-based approaches [336, 337] strive to modify the optimization methods. For example, Model Agnostic Meta Learning (MAML) [337] is one of the most popular approaches. Through the adjustment of the gradient descent and computation, MAML contributes to rapid optimization and is compatible with any models based on the gradient descent algorithm.

Recently, similarity-based approaches has developed rapidly in the computer vision domain, particularly in image classification. These approaches firstly map the input samples into representation or embeddings using a feature extractor that usually is a neural network. Through the measurement of the similarity (i.e. cosine similarity, euclidean distance) among the embeddings of samples, similarity-based approaches

aim to find the labelled sample with the highest similarity of unseen samples. The early development stage of similarity-based approaches focus on pairwise comparators such as Siamese networks[329] and Triplet networks[320], judging whether the samples in data pairs are from the same or different classes. Following training, pairwise comparators narrow the inner-class distance while expand the between-class distance, efficiently discriminating between different classes. Compared with pairwise comparators, multi-class comparators [330, 331, 338] are more compatible with N-way k-shot training and testing settings; here the embeddings and final classification are obtained in an end-to-end manner. For instance, Matching networks [331] utilize the neural network as a feature extractor and measure the similarities among embeddings via cosine similarity. Unlike aforementioned studies, Matching networks train and test on N-way K-shot tasks, predicting the unseen samples in an end-to-end manner. Additionally, Matching networks adopt bidirectional LSTM with attention mechanism to encode input samples, which enables more robust feature embeddings.

As the FSL develops in image processing, recent studies propose novel approaches such as graph-network-based approaches [339, 340] and GAN-based approaches [341, 342]. Therefore, state-of-art neural networks and deep learning technologies can further optimise few-shot learning, contributing to effective methods for image classification, object detection, etc.

6.3 Methodology

The pipeline of our proposed few-short learning approach for ECG beats classification is shown in Figure 6.1. The major four modules of the methodology are as follows: (i) problem formulation; (ii) data pre-processing; (iii) original prototypical networks; and (iv) multi-scale CNN based prototypical network.

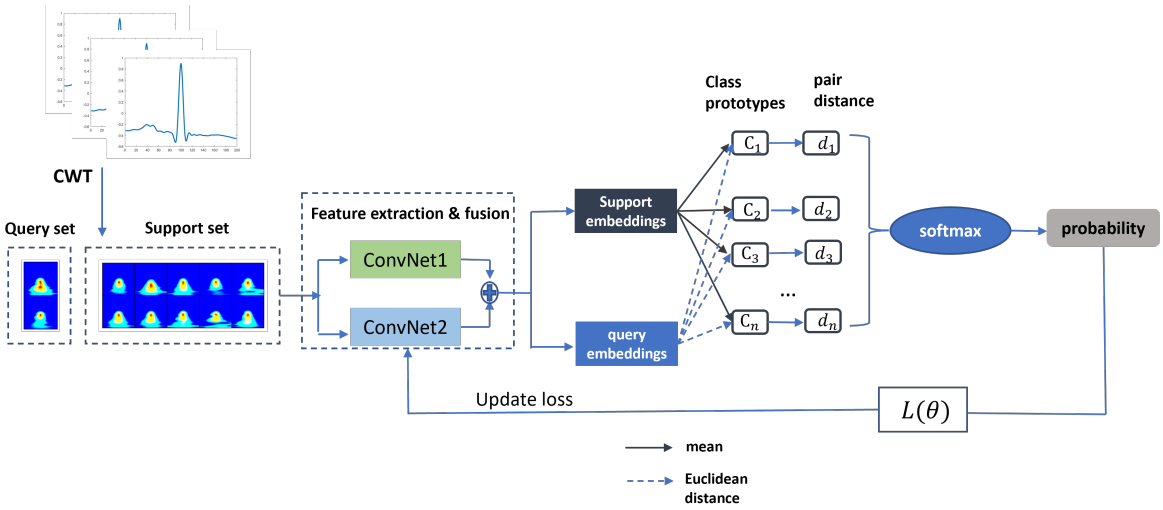

Figure 6.1: The pipeline of the proposed model for few-shot ECG beats classification with the structure of the Parallel multi-scale convolution based prototypical network.

6.3.1 Problem formulation

In our study, researcher addresses the ECG based arrhythmia classification as a few-shot learning problem. In the FSL dataset, researcher splits the original dataset $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ as the train set D_{train} and the test set D_{test}. The label space of the train set and test set is disjoint ($D_{train} \cap D_{test} = \varnothing$). Researcher trains the model in an episodic approach and define an N-way K-shot problem for each episode. In each episode, researcher randomly selects N unique unseen classes where each class involves K labelled samples as a support set S. The query set Q contains unlabelled samples, sharing the same label space with S.

Through the training on the D_{train}, FSL aims to predict the label of query samples based on the support set S and minimize the loss of prediction during training episodes.

6.3.2 Data pre-processing

To minimize the disadvantages of noise for arrhythmia classification, an 8-order Butterworth low pass filter (35Hz) is adopted for eliminating noise. Then, the filtered signals are segmented into beat by beat based on the R-peak locations marked on the

annotation file. Each segment as a signal beat consists of 99 samples (i.e., data points) before the R-peak and 100 samples after it. To extract features on both temporal and spectral dimensions of ECGs, researcher characterizes one-dimension signals as time-frequency spectrograms via continuous wavelet transform (CWT) [226]. The wavelet transform utilizes multiresolution analysis, changing the size of the sliding window on different frequencies. The adoption of wavelet transform contributes to a smoother feature observation of the nonstationary signal at different scales. The continuous wavelet transform of ECG signal $f(t)$ is:

$$C\left(a,b;f(t),\varphi(t)\right)=\int_{-\infty}^{\infty}f(t)\frac{1}{a}\varphi*(\frac{t-b}{a})\,dt \tag{6.1}$$

where φ is a wavelet as analysing function, a is the scale parameter and b is the position parameter. Through CWT, the 1D ECG beat signals are transformed into time-frequency spectrograms with and then converted as three-channel RGB via ind2rgb from Matlab, forming three channel inputs for the convolutional 2D CNN feature extractor.

6.3.3 Original Prototypical network algorithm

The prototypical network firstly aims to create a feature embedding of each image sample via feature extractor $f_\theta : R^D \rightarrow R^M$. In the embedding space R^M, the inner-class distance is small and the inter-class distance is large. Then, the mean of support feature embeddings S_i is computed as the class prototype for class i:

$$P_i=\frac{1}{|S_i|}\sum_{(x_n,y_n)\in S_i}f_\theta(x_n) \tag{6.2}$$

where the θ represents parameters of the feature extractor and i represents a class in the support set. For classifying query samples, researcher computes the Euclidean distance between the query features embeddings and prototypes, then use a softmax function to predict the label for query samples. The probability of the query sample q_i with a predicted class c can be formally written as follows:

$$p_\theta(y_n=c|q_i)=softmax(-d(f_\theta(q_i),p_i)) \tag{6.3}$$

Where the $d(\cdot)$ is the Euclidean distance function. In the original prototypical network, Snell et al. [330] process the learning phase by minimizing the log-softmax loss, which

can be defined as:

$$L_\theta = -logp_\theta(y_n = c|q_i) \tag{6.4}$$

6.3.4 Parallel Multi-scale based convolutional network

Considering the high efficiency of the prototype network in the few-shot learning domain. The design concept of the proposed model is based on the original prototypical network. In few-shot learning studies[330, 343], prototypical networks use a CNN as a feature extractor. However, the traditional CNN structure consists of sequentially connected convolutional layers and pooling layers, extracting image features layer by layer. The features extracted via that structure are relatively simple. Furthermore, using single-size kernels intensively can lead to filter adaptation and over-fitting [344], which is not conducive to identifying the waveform details of ECG beats. According to these studies [344–346], multiscale CNN (MCNN) can extract robust features from images by different sizes of convolution kernels, where the larger kernel size is more conducive to extracting global feature and the smaller kernel size capture the detailed features better.

Thus, the researcher proposes to process feature extraction via two parallel convolutional networks with different kernel sizes. As shown in Figure 6.2, the PM-CNN consists of two streams of 16-layers CNN followed by a Flatten layer and a Dense layer for feature dimension reduction. Since the deeper CNN caused over-fitting problem, there are 4 convolutional layers in each stream. Table 6.1 illustrates the structure details of PM-CNN. Inspired by studies[344, 345], the kernel size of the convolutional layer in convnet1 is 3*3 and that of convnet2 is 7*7, obtained the robust features of different scales from ECG beats. Also in these studies, the approach of feature fusion normally uses concatenate operations. Using concatenate operation can result in excessive dimensions of extracted features while increasing the computation. Thus, the researcher uses a linear combination to fuse the features $(f_1(x, \theta_1), f_2(x, \theta_2))$ separately extracted from convnet1 and convnet2. With the introduction of a learnable parameter σ, PM-CNN can adaptively adjust the weight of features those learning from the convnets with different sizes of kernels. The formulation of the feature fusion can be

defined as follows:

$$Pro_\theta(x) = \mu f_1(x, \theta_1) + (1 - \mu) f_2(x, \theta_2) \tag{6.5}$$

where $f_1(\cdot)$ and $f_2(\cdot)$ represent different convnets, and then $Pro_\theta(\cdot)$ is the ultimate embedding function for the prototypical network. The fused embedding features are averaged to form a class prototype for the support set, which is formulated below:

$$P_i = \frac{1}{|S_i|} \sum_{(x_n, y_n) \in S_i} Pro_\theta(x_n) \tag{6.6}$$

The training algorithm of Parallel multi-scale CNN is given below.

Algorithm 3 Pesudocode for training of PMCNN Prototypical Network. K is the number of training set samples, N is the number of training set classes, $K_c <= N$ is the number of classes per episode, K_s is the number of support set samples per class, K_Q is the number of query set samples per class. Random(S,N) denotes a set of N samples randomly chosen from set S. convnet1 and convnet2 denote two strems of PMCNN respectively.

Input: Training set$D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_K, y_K)\}$, where each $y_i \in \{1, ...N\}$.D_c denotes the subset of D containing all samples(x_i, y_i) such that $y_i = c$

Output: The loss L for the training episode.

T←Random({1,..,N},K$_c$) ▷ select class indices for episode
for k in $\{1, ..., K_c\}$ **do**
  $S_k \leftarrow Random(D_{T_k}, K_s)$ ▷ select support samples
  $Q_k \leftarrow Random(D_{T_k} \setminus S_k, K_Q)$ ▷ select query samples
  $S_{K1} \leftarrow convent1(S_k)$ ▷ obtain support set embeddings from convent1
  $S_{K2} \leftarrow convent2(S_k)$ ▷ obtain support set embeddings from convent2
  $Q_{K1} \leftarrow convent1(Q_k)$ ▷ obtain query set embeddings from convent1
  $Q_{K2} \leftarrow convent2(Q_k)$ ▷ obtain query set embeddings from convent2
  $Pro_\theta(S_{kfused}) \leftarrow \mu S_{k1} + (1 - \mu) S_{k2}$ ▷ feature fusion for support set
  $Pro_\theta(Q_{kfused}) \leftarrow \mu Q_{k1} + (1 - \mu) Q_{k2}$ ▷ feature fusion for query set
  $P_k \leftarrow \frac{1}{K_c} \sum_{(x_i, y_i) \in S_{kfused}} Pro_\theta(x_i)$
end for
$L \leftarrow 0$ ▷ initialize loss function
for k in in $\{1, ..., K_c\}$ **do**
  for (x, y) in Q_{kfused} **do**
    $L \leftarrow L + \frac{1}{K_c K_Q} [d(Pro_\theta(x), P_k) + log \sum_{K'} exp(-d(Pro_\theta(x), P_{k'}))]$ ▷ loss update
  end for
end for

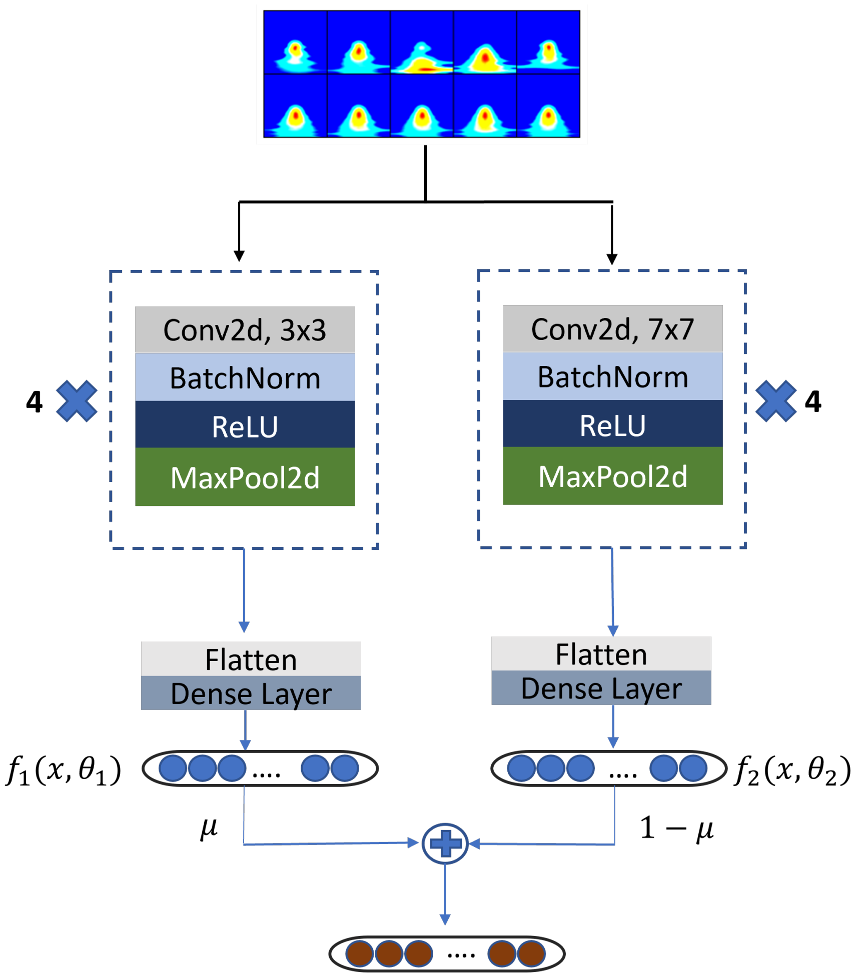

Figure 6.2: Illustration of the structure of the parallel multi-scale CNN.

Table 6.1: Parameter details of all layers in parallel multi-scale CNN

Convnet1		**Convnet2**	
Layers	***Kernel number/size***	**Layers**	***Kernel number/size***
Conv2D+BN+ReLu	64, 3*3	Conv2D+BN+ReLu	64, 7*7
MaxPool2D	2*2	MaxPool2D	2*2
Conv2D+BN+ReLu	64, 3*3	Conv2D+BN+ReLu	64, 7*7
MaxPool2D	2*2	MaxPool2D	2*2
Conv2D+BN+ReLu	64, 3*3	Conv2D+BN+ReLu	64, 7*7
MaxPool2D	2*2	MaxPool2D	2*2
Conv2D+BN+ReLu	64, 3*3	Conv2D+BN+ReLu	64, 7*7
MaxPool2D	2*2	MaxPool2D	2*2
Flatten		Flatten	
Dense	512	Dense	512

6.4 Experiment and Result

Researcher implements and evaluates the proposed model on the MIT-BIH arrhythmia database. The arrhythmia database contains 48 two-leads ECG recordings of 30 minutes duration obtained from 47 subjects. The signal in each record is pre-processed by a bandpass filter in the 0.1-100 Hz range and sampled at 360 Hz. The beats labels and R-peak location of the signals are annotated by cardiologists. The details of 15 types of ECG beats in MITDB are shown in Table 6.2. Researcher random selected 8,000 samples from those of type N due to its excessive number of samples. Only 14 types of beats are adopted in experiments since the number of beats S is not applied to N-way K-shot tasks.

Table 6.2: Numbers and labels of beats in MIT-BIH arrhythmia dataset

Heartbeat types	**♯ of beats**
Normal(N)	74476
Left Bundle Brunch Block (L)	8043
Right Bundle Brunch Block (R)	7225
Atrial Escape (e)	16
Nodal (junctional) escape(j)	224
Atrial Premature (A)	2521
Aberrant Atrial Premature (a)	67
Nodal (Junctional) Premature (J)	82
Supra-Ventricular Premature (S)	2
Premature Ventricular Contraction (V)	5341
Ventricular escape (E)	106
Fusion of Ventricular and Normal (F)	784
Paced (/)	7001
Fusion of Paced and Normal (f)	968
Unclassifiable (Q)	25

6.4.1 Experimental Setting

In our experiment, 14 types of arrhythmias from MIT-BIH are employed for training and testing. The composition of the train set greatly affects the model performance on the test set due to the limited number of classes. Moreover, the classification performance based on a single test set cannot demonstrate the robustness of the model. Thus, researcher randomly selects 9 classes from 14 types of arrhythmias as the prototypical train set and regarding the remaining 5 classes as the adaption test set. The random selection for train-test splits will repeat 5 times and the mean of 5 test accuracies combined with 95% confidence interval is the result of model performance. To evaluate the effectiveness of multi-scale structure for the prototypical network, researcher performs 2-way and 4-way classification tasks, corresponding with 1-shot, 5-shot and 10-shot. In the comparison experiment between our model and other metric-based FSL models, the training and evaluation are based on 4 tasks are, 2-way 1-shot, 2-way 5-shot, 4-way 1-shot and 4-way 5-shot. For all experiment, the training episodes is 500 and the test episodes is 200.

The proposed model is implemented and evaluated on Keras 2.3.0 based TensorFlow 2.1.0 while running on the Tesla P100 GPU. Researcher adopts the Adam optimizer to optimize the model during training and set the initial learning rate as 0.002. The loss function utilized in our model is categorical cross-entropy.

6.4.2 Contribution of Parallel Multi-scale CNN

In this section, researcher aims to explore the contribution of PM-CNN based prototypical networks. For the experiment of the original ProtoNet, researcher uses a feature extractor with the same structure as convnet 1 in Table 6.1, simultaneously remaining the hyperparameters unchanged. The comparison of results between PM-CNN ProtoNet and original ProtoNet in N-way K-shot tasks is shown in Table 6.3. In most of the classification tasks, the performance of PM-CNN ProtoNet is better than that of the original ProtoNet. Especially in 2-way 1-shot setting with only one support sample per class in training, 1.92% improvement of accuracy is obtained compared with the original ProtoNet. The gap between the PM-CNN ProtoNet and the

original ProtoNet in 2-way tasks is greater than that in 4-way tasks.

Table 6.3: Comparative results of PM-CNN prototypical network and original prototypical network for N-way k-shot classification tasks. The performance is regarded as mean accuracies (%) with 95% confidence interval. The prototypical network is denoted as ProtoNet

Model	**2-way tasks**			**4-way tasks**		
	1-shot	5-shot	10-shot	1-shot	5-shot	10-shot
PM-CNN ProtoNet	77.50 ± 6.50	87.50 ± 4.80	91.60 ± 3.70	59.72 ± 6.28	75.66 ± 6.45	74.83 ± 7.75
Original ProtoNet [330]	75.58 ± 7.50	85.80 ± 6.30	90.40 ± 4.80	58.79 ± 5.90	75.02 ± 6.30	75.77 ± 7.40

To concatenate the features from multi-scale CNN, researcher introduces a learnable parameter μ which illustrates the proportion of different convnets. In experiments, researcher initializes the value of μ as 0.5 and monitor its changes during N-way K-shot tasks. As shown in Table 6.4, the value of μ changes slightly in different few shot tasks. Thus, both convents with different kernel sizes occupy considerable proportion in feature extraction, demonstrating the availability of the multi-scale CNN.

Table 6.4: Value of μ in PM-CNN ProtoNet

Setting	**2-way tasks**			**4-way tasks**		
	1-shot	5-shot	10-shot	1-shot	5-shot	10-shot
Value of μ	0.5	0.51	0.53	0.49	0.48	0.50

Although the increase of N-way may disturb the performance of the proposed model, the PM-CNN structure still contributes to higher accuracies with only one support sample. The comparative results demonstrate the contribution of the proposed model to classification performance. Furthermore, the PM-CNN can extract and fuse different scaled features without rising the depth of the network, partly preventing overfitting.

6.4.3 Comparison between current method

To further assess the performance of the PM-CNN prototypical Network, researcher compares the proposed model with three metric-based FSL models: Siamease Network [329], Matching Network [331] and Relation network [338]. All models are trained and tested on the same dataset for comparison. The comparative results of three state-of-art models and PM-CNN ProtoNet in N-way K-shot tasks are illustrated in Figure 6.3. The performance of the proposed model surpasses the Siamese network and Relation network in most cases. Especially in the 4-way 5-shot task, the PM-CNN ProtoNet achieves a mean accuracy of 75.66% which is higher than those of the other three models at 14.4%, 5.0% and 49.6% respectively.

The relation network shows poor performance in all cases, where merely achieves 51.6% on 2-way 5-shot task. Different from other metric-based few-shot models, Relation network adopts a shallow CNN as a relation module to evaluate the distance between query samples and support samples. The ECG beats spectrogram may not fit the relation module, and then fail to obtain valid relation scores between the support samples and query samples. The mean accuracy of the proposed model are obviously higher than those of Matching Net in 2-way tasks and slightly lower than those of Matching Net in 4-way tasks. Although the performance of the proposed model is not steaming ahead in all cases, the smaller confidence uncertain interval illustrates its performance certainty. Besides, the imbalanced dataset and CWT-induced redundancy may also limit the performance of the proposed model, which can be improved in the future work

6.5 Limitation

There are a few potential limitations in this study. Firstly, The one-dimensional ECG signals are transformed as three-channel RGB images via CWT and RGB converter, whereas the CWT is computationally intensive and adds redundancy in the representations of ECGs. Secondly, the researcher merely uses standard kernel size for the parallel multi-scale CNN and processes the training and testing on a single dataset. The robustness of the proposed approach has not been fully demonstrated due to the

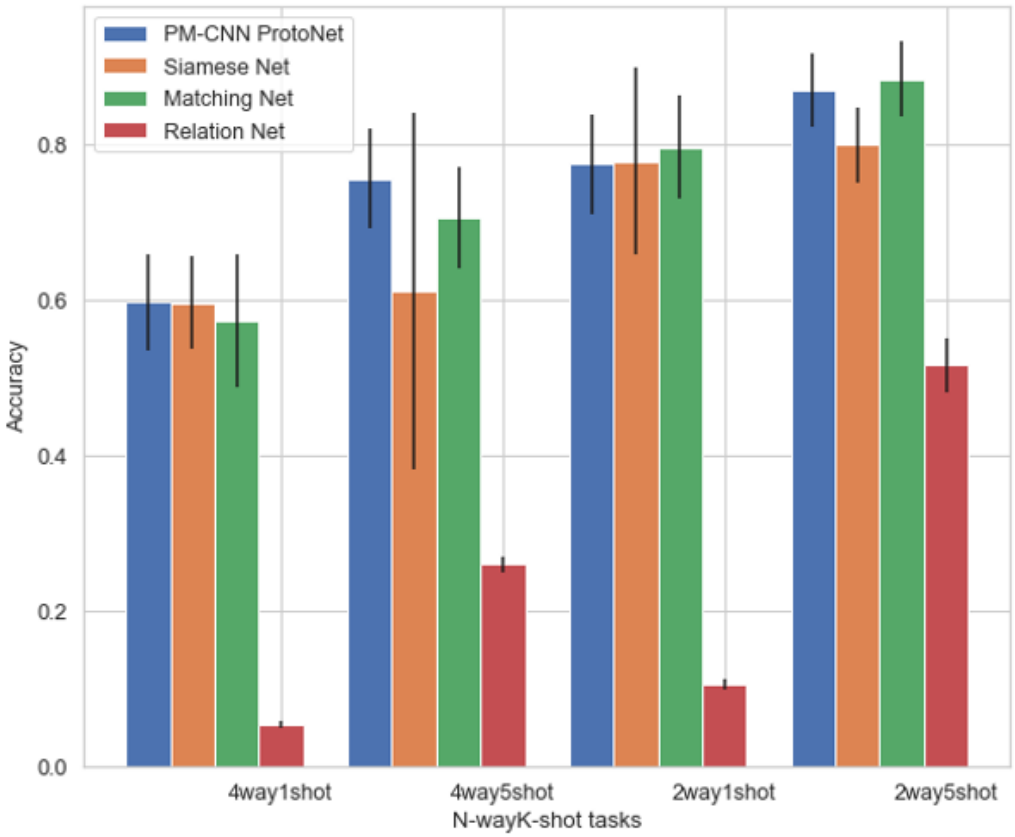

Figure 6.3: The experiment result of the proposed model and other three state-of-art models with the same dataset and experimental setting.

lacking of kernel-size tunning and cross-validation. To provide accurate treatment for cardiac patients, high precision in automatic detection is required. Although the proposed model has potential in arrhythmias detection, its accuracy still can be improved.

6.6 Conclusion

In this paper, researcher presents a parallel multi-scale CNN based prototypical network, as a few-shot learning model to screen different types of arrhythmias based on ECG beats. In pre-processing stage, researcher proposes to transform the ECG beat signals to time-frequency spectrograms for more comprehensive information. Then the parallel multi-scale CNN can simultaneously capture smooth and detailed feature information using different scaled kernel sizes, further providing robust feature embeddings for the prototypical network. Concerning the comparative results of experiments, the proposed model can accomplish most N-way K-shot tasks with satisfying accuracy and illustrate its effectiveness among other metric-based FSL models. In addition,

the proposed model based on relatively shallow CNN avails comprehensive feature extraction whilst contributing small calculation quantity. Consequently, the parallel multi-scale CNN based prototypical network has potential application in arrhythmia detection on a small number of samples.

Chapter 7

General Discussion and Conclusion

7.1 Introduction

The novelty and significance of the works from the thesis are asserted in this chapter, which serves as a general discussion and conclusion of the key research efforts. Additionally, the potential future work of this thesis is listed below.

7.2 Novelty and Methodology Discussion

The studies presented in this thesis are focused on reducing the expense of manual diagnosis through automatic classification of cardiac arrhythmias using advanced deep learning technologies. For contributing to higher classification accuracy and efficiency, these studies proposed some novel improvements based on the state-of-art models and technologies in the deep learning domain. Firstly in chapter 4, frame blocking is a novel pre-processing method for ECG signals, which is the first attempt to divide the raw ECG signals into uni-sized frames with the least loss of genuine signals. Dividing the multi-label classification into multiple binary classification tasks was easy to interpret and conduct. Additionally, the proposed classification model consists of ResNet-based CNN and attention-based Bi-LSTM, which combined the state-of-art deep neural network structures and analysed the 12-lead ECG automatically without handcrafted features.

Second, deep metric learning was employed for reducing the model complexity and memory usage, all while exploring some novel techniques in the computer vision domain. In chapter 5, the presented auto-detection method also adopted the frame-blocking method proposed in chapter 4 in preprocessing stage. Different from the training algorithm used in chapter 4, chapter 5 adopted a metric-learning training algorithm so that the model can minimize the inner-class distance efficiently. Because CNN merely focused on the morphology of the input data, chapter 5 employed the temporal features based on the RR intervals and combined them with morphological features to produce more distinctive features. Compared with the model complexity and classification performance in chapter 4, chapter 5 had a simpler CNN-based model(encoder) while slightly lower classification performance.

Previous chapters 4 and 5 concentrated on the auto-detection methods based on the large amount of training data, which was inapplicable to detect rare arrhythmias under limited ECG instances. Thus, the third project in the thesis adopted the few-shot learning strategy to propose a parallel multi-CNN-based prototypical network for arrhythmia detection. Different from Chapters 4 and 5, chapter 6 divided ECG signals beat by beat for obtaining more training samples of rare types of arrhythmias. Then, the researcher transformed ECG beats into RGB images for extracting features on both temporal and spectral dimensions of ECGs. Currently, few ECG detection approaches are implemented based on few-shot learning. The novel PM-CNN is composed of two parallel convolutional networks with different kernel sizes, which maps the model input into embedding space and contributes to robust feature embeddings. Moreover, chapter 6 employed a parallel structure of shallow CNN, contributed to less model complexity and prevented over-fitting.

7.3 Summary of Major Findings and Novel Contributions

The major findings and contributions are summarized below:

7.3.1 Chapter4: Automatic Detection for Multi-labeled Cardiac Arrhythmia Based on Frame Blocking Preprocessing and Residual Networks

The important findings of this chapter are:

1) The frame blocking technique can decompose the 12-lead ECG signals into overlapped frames during preprocessing stage. In the process of decomposing unequal length ECG into frame blocks with unified size, the frame blocking technique minimised the loss of valid ECG signals while maintaining the continuity of signals. The proposed preprocessing method exhibited its benefit and improved classification performance when compared to popular preprocessing methods (zero paddingdirect cutting).

2) The classification model utilized the structure of the residual network and attention-based BiLSTM. The ResNet-based network considerably outperformed other deep neural networks in terms of classification performance and alleviated the exploring/vanishing gradient issue that plagues deep neural networks. Additionally, the combination of attention mechanism and BiLSTM effectively focused on the essential part of the input. The attention-based BiLSTM caught the global and local connection simultaneously based on the weights and attention allocation for the input.

3) This work improved the interpretation of the feature extraction by dividing the multi-labelled classification into multiple binary classification jobs, where each cardiac state is recognised as an independent binary task.

7.3.2 Chapter5: Fusing Deep Metric Learning with KNN for 12-lead Multi-labeled ECG Classification

The crucial findings of this study are:

1) Similar to the work in chapter 4, the preprocessing also adopted the frame blocking, efficiently retained the genuine ECG signals and unified the length of different ECG signals.

2) The presented auto-detection method demonstrated its superiority in classification performance as a fresh trial of employing the combination of metric learning and deep neural networks for ECG auto-detection

3) While providing a more complete vision of heart activity, 12-lead ECG are also more complex than single-lead ECG. The presented method demonstrated its advantages in the classification of subtle morphological differences based on the 12-lead ECG.

4) Through experiments, the fusion of morphological features and temporal features significantly improved the classification performance. ResNet-based model merely focused on the morphological information, but combining morphological and temporal features can result in comprehensive features that are more efficient in discriminating heart abnormalities.

7.3.3 Chapter6: Parallel Multi-scale Convolution based Prototypical Network for Few-shot ECG beats Classification

The significant findings in this work are listed below:

1) As a quite novel algorithm, the few-shot learning can be used for the ECG automatic detection and has the potential to address the issue of insufficient training ECG samples due to the rareness of arrhythmia. Through the few-shot training strategy, the classification model was trained in an episodic approach and each episode contained an N-way k-shot classification task. This approach allowed the classification model to detect unseen classes of arrhythmia solely based on a small number of training samples (1,5,10).

2) This work implemented a PM-CNN prototypical network which consisted of two parallel CNN (convnet1, convnet2) with different kernel sizes. The features that were individually retrieved from two CNNs can be successfully combined using linear combination. The PM-CNN prototype network outperformed the conventional prototypical neural network[330] in most N-way k-shot tasks with higher accuracy. Additionally, the PM-CNN can accomplish extraction and fusion of different scaled features without expanding the depth of the neural network, partially preventing overfitting.

3) According to the comparison between the proposed model and other metric-based FSL models, the performance of the proposed model visibly surpassed that of Siamese Net and Relation Net. The proposed model displayed good stability when tested against various learning tasks and randomly chosen classes.

7.4 Potential Limitations

There are a few potential limitations in the works of this thesis:

1. In chapter 4, using random under-sampling to address the imbalanced datasets could lead to the loss of important information from data in the majority classes.

Random under-sampling randomly discards samples from majority classes, weakens the decision boundary between different classes and may introduce bias during model training.

2. In chapter 4, the presented approach transformed the multi-label classification problem into multiple binary-classification tasks, which required a long training time, large computation and memory usage. Furthermore, the training time and memory usage will be substantially increased since the types of arrhythmias are more.

3. As for the feature fusion stage in chapter 5, RR interval-based temporal features belong to handcrafted features, imposed additional time cost.

4. The proposed model in chapter 5 differs from existing state-of-art metric deep learning models [320, 321] in that it only considers the inner-class distance while neglecting the inter-class distance.

5. Both of the models proposed in chapters 45 required a large number of training samples and computation. However, these models are challenging to process the training based on some rare arrhythmias with insufficient clinical records.

6. Few-shot learning was involved in chapter 6 for providing a novel training method and addressing the problem of insufficient ECG samples. The one-dimensional signals were transformed as three-channel RGB images during preprocessing stage using CWT and RGB converter from MATLAB. CWT is computationally intensive and adds redundancy. Furthermore, RGB images provided more discriminated features, whilst also increased the computational cost during training.

7. The few-shot learning-based PM-CNN was not cross-validated on additional datasets because there are so few datasets that match the experimental condition. Thus, the robustness of the proposed approach has not been fully demonstrated.

Whilst it is crucial to enumerate the potential limitations of the approaches proposed in this thesis, they have no bearing on the conclusions.

7.5 Future Work

The works presented in this thesis pose some interesting research questions and limitations for future improvement, the most crucial of which will be discussed below. In chapter 4, the imbalanced dataset was addressed via random under-sampling, which could result in the loss of important data from the majority class. Besides simple data-level methods (i.e.under-sampling, over-sampling), a few algorithm-level methods and hybrid methods can be further explored. To lessen bias towards the negative class, auto-detection algorithms are usually modified to take a class penalty or weight into account or to change the prediction threshold. It is worthwhile to investigate more adaptive algorithms [42, 285, 286] to reduce time-demand for training and increase classification efficiency.

In chapter 5, the proposed model combined metric learning with a ResNet-based neural network, evaluated the mapping from ECG signals to feature embeddings and reflected the similarity between input ECG samples via the distances between feature embeddings. The inner-class distance can be minimized for the final classification of different types of arrhythmia by training the deep metric models. However, the inter-class distance should also be considered for a more precise classification. Studies[320–322, 324] provided inspiration of the improvement of deep metric-based models. Using the triplet network and triplet/proxy loss function can narrow the inner-class distance and simultaneously extend the inter-class distance, which may improve the classification accuracy and efficiency.

In chapter 6, the few-shot learning algorithm is quite novel in the ECG detection domain. The proposed PM-CNN prototypical network conducts the ECG beats classification based on a few-shot learning algorithm, which has the potential to address the problem of insufficient training samples of uncommon arrhythmias. However, the CWT-induced redundancy may limit the classification performance of the proposed model. As a result, other types of time-frequency analysis techniques can be further explored and other state-of-art few-shot-learning-based models can be kept tracking in the future.

On a more practical level, deep learning-based auto-detection techniques can improve the applicability of warning or auto-diagnosing systems of arrhythmias. The portable auto-diagnosing application is still difficult to spread since prophase training of deep learning models requires a large amount of computation. The use of cloud computing for deep learning models in the future is a solid approach, allowing for the simple storage of a large amount of training data and low cost of GPU during the training process of deep learning models. Additionally, it may contribute to realizing the real-time monitoring system which can provide early warning for the patients suffering from arrhythmias.

7.6 Closing Words

The projects presented in the thesis show an ongoing exploration of the deep-learning-based ECG auto-detection approaches. These projects provided improvements in pre-processing techniques, deep neural network structures and training strategy, which contributes to the satisfying performance of ECG automatic detection. The observations in this thesis have the potential to prevent stroke or sudden death caused by cardiac arrhythmia. Currently, an increasing number of clinical applications use the ECG automatic detection approaches for emergency medical services such as pre-hospital ECG analysis, allowing for effective and timely treatments. The prevention and treatment of cardiac arrhythmias will continue to be improved in the future, according to the author, by the use of more deep-learning-based ECG detection methods.

Bibliography

[1] Ayano Chiba, Haruko Watanabe-Takano, Takahiro Miyazaki, and Naoki Mochizuki. Cardiomyokines from the heart. *Cellular and Molecular Life Sciences*, 75(8):1349–1362, 2018.

[2] Alain Karma. Physics of cardiac arrhythmogenesis. *Annu. Rev. Condens. Matter Phys.*, 4(1):313–337, 2013.

[3] Anthony H Kashou, Hajira Basit, and Lovely Chhabra. Physiology, sinoatrial node (sa node). 2017.

[4] Himanshu Gothwal, Silky Kedawat, Rajesh Kumar, et al. Cardiac arrhythmias detection in an ecg beat signal using fast fourier transform and artificial neural network. *Journal of Biomedical Science and Engineering*, 4(04):289, 2011.

[5] Edoardo Bertero and Christoph Maack. Metabolic remodelling in heart failure. *Nature Reviews Cardiology*, 15(8):457–470, 2018.

[6] Gregory A Roth, Degu Abate, Kalkidan Hassen Abate, Solomon M Abay, Cristiana Abbafati, Nooshin Abbasi, Hedayat Abbastabar, Foad Abd-Allah, Jemal Abdela, Ahmed Abdelalim, et al. Global, regional, and national age-sex-specific mortality for 282 causes of death in 195 countries and territories, 1980–2017: a systematic analysis for the global burden of disease study 2017. *The Lancet*, 392(10159):1736–1788, 2018.

[7] Adam Timmis, Nick Townsend, Chris P Gale, Aleksandra Torbica, Maddalena Lettino, Steffen E Petersen, Elias A Mossialos, Aldo P Maggioni, Dzianis Kazakiewicz, Heidi T May, et al. European society of cardiology: cardiovascular disease statistics 2019. *European heart journal*, 41(1):12–85, 2020.

[8] Santanu Sahoo, Asit Subudhi, Manasa Dash, and Sukanta Sabut. Automatic classification of cardiac arrhythmias based on hybrid features and decision tree algorithm. *International Journal of Automation and Computing*, 17(4):551–561, 2020.

[9] Sandeep Raj and Kailash Chandra Ray. Automated recognition of cardiac arrhythmias using sparse decomposition over composite dictionary. *Computer methods and programs in biomedicine*, 165:175–186, 2018.

[10] Taiyong Li and Min Zhou. Ecg classification using wavelet packet entropy and random forests. *Entropy*, 18(8):285, 2016.

[11] Shirin Shadmand and Behbood Mashoufi. A new personalized ecg signal classification algorithm using block-based neural network and particle swarm optimization. *Biomedical Signal Processing and Control*, 25:12–23, 2016.

[12] Jihong Yan and Lei Lu. Improved hilbert–huang transform based weak signal detection methodology and its application on incipient fault diagnosis and ecg signal analysis. *Signal Processing*, 98:74–87, 2014.

[13] R Rodríguez, A Mexicano, J Bila, S Cervantes, and R Ponce. Feature extraction of electrocardiogram signals by applying adaptive threshold and principal component analysis. *Journal of applied research and technology*, 13(2):261–269, 2015.

[14] William S Noble. What is a support vector machine? *Nature biotechnology*, 24(12):1565–1567, 2006.

[15] Santanu Sahoo, Bhupen Kanungo, Suresh Behera, and Sukanta Sabut. Multiresolution wavelet transform based feature extraction and ecg classification to detect cardiac abnormalities. *Measurement*, 108:55–66, 2017.

[16] Shameer Faziludeen and PV Sabiq. Ecg beat classification using wavelets and svm. In *2013 IEEE Conference on Information & Communication Technologies*, pages 815–818. IEEE, 2013.

[17] Argyro Kampouraki, George Manis, and Christophoros Nikou. Heartbeat time series classification with support vector machines. *IEEE transactions on information technology in biomedicine*, 13(4):512–518, 2008.

[18] Sandeep Raj and Kailash Chandra Ray. Ecg signal analysis using dct-based dost and pso optimized svm. *IEEE Transactions on instrumentation and measurement*, 66(3):470–478, 2017.

[19] Carolina Varon, Dries Testelmans, Bertien Buyse, Johan AK Suykens, and Sabine Van Huffel. Sleep apnea classification using least-squares support vector machines on single lead ecg. In *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 5029–5032. IEEE, 2013.

[20] Essam H Houssein, Ahmed A Ewees, and Mohamed Abd ElAziz. Improving twin support vector machine based on hybrid swarm optimizer for heartbeat classification. *Pattern Recognition and Image Analysis*, 28(2):243–253, 2018.

[21] Zhongheng Zhang. Introduction to machine learning: k-nearest neighbors. *Annals of translational medicine*, 4(11), 2016.

[22] Ridhi Saini, Namita Bindal, and Puneet Bansal. Classification of heart diseases from ecg signals using wavelet transform and knn classifier. In *International Conference on Computing, Communication & Automation*, pages 1208–1215. IEEE, 2015.

[23] C Venkatesan, P Karthigaikumar, and RJMT Varatharajan. A novel lms algorithm for ecg signal preprocessing and knn classifier based abnormality detection. *Multimedia Tools and Applications*, 77(8):10365–10374, 2018.

[24] S Rasoul Safavian and David Landgrebe. A survey of decision tree classifier methodology. *IEEE transactions on systems, man, and cybernetics*, 21(3):660–674, 1991.

[25] Emina Alickovic and Abdulhamit Subasi. Medical decision support system for diagnosis of heart arrhythmia using dwt and random forests classifier. *Journal of medical systems*, 40(4):1–12, 2016.

[26] Ahmet Mert, Niyazi Kilic, and Aydın Akan. Ecg signal classification using ensemble decision tree. *J Trends Dev Mach Assoc Technol*, 16(1):179–182, 2012.

[27] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

[28] Roshan Joy Martis, U Rajendra Acharya, Choo Min Lim, KM Mandana, Ajoy K Ray, and Chandan Chakraborty. Application of higher order cumulant features for cardiac health diagnosis using ecg signals. *International journal of neural systems*, 23(04):1350014, 2013.

[29] Roshan Joy Martis, U Rajendra Acharya, and Lim Choo Min. Ecg beat classification using pca, lda, ica and discrete wavelet transform. *Biomedical Signal Processing and Control*, 8(5):437–448, 2013.

[30] Vega Pradana Rachim, Gang Li, and Wan-Young Chung. Sleep apnea classification using ecg-signal wavelet-pca features. *Bio-medical materials and engineering*, 24(6):2875–2882, 2014.

[31] Mark J Shensa et al. The discrete wavelet transform: wedding the a trous and mallat algorithms. *IEEE Transactions on signal processing*, 40(10):2464–2482, 1992.

[32] VK Srivastava and Devendra Prasad. Dwt-based feature extraction from ecg signal. *American J. of Eng. Research (AJER)*, 2(3):44–50, 2013.

[33] Murugappan Murugappan, Subbulakshmi Murugappan, and Bong Siao Zheng. Frequency band analysis of electrocardiogram (ecg) signals for human emotional state classification using discrete wavelet transform (dwt). *Journal of physical therapy science*, 25(7):753–759, 2013.

[34] Gabriel Rilling, Patrick Flandrin, Paulo Goncalves, et al. On empirical mode decomposition and its algorithms. In *IEEE-EURASIP workshop on nonlinear signal and image processing*, volume 3, pages 8–11. Citeseer, 2003.

[35] Santanu Sahoo, Monalisa Mohanty, Suresh Behera, and Sukanta Kumar Sabut. Ecg beat classification using empirical mode decomposition and mixture of features. *Journal of medical engineering & technology*, 41(8):652–661, 2017.

[36] Elif Izci, Mehmet Akif Ozdemir, Reza Sadighzadeh, and Aydin Akan. Arrhythmia detection on ecg signals by using empirical mode decomposition. In *2018 Medical Technologies National Congress (TIPTEKNO)*, pages 1–4. IEEE, 2018.

[37] Yuksel Ozbay and Bekir Karlik. A recognition of ecg arrhytihemias using artificial neural networks. In *2001 Conference Proceedings of the 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, volume 2, pages 1680–1683. IEEE, 2001.

[38] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. Understanding of a convolutional neural network. In *2017 international conference on engineering and technology (ICET)*, pages 1–6. Ieee, 2017.

[39] U Rajendra Acharya, Shu Lih Oh, Yuki Hagiwara, Jen Hong Tan, Muhammad Adam, Arkadiusz Gertych, and Ru San Tan. A deep convolutional neural network model to classify heartbeats. *Computers in biology and medicine*, 89:389–396, 2017.

[40] Chen Chen, Zhengchun Hua, Ruiqi Zhang, Guangyuan Liu, and Wanhui Wen. Automated arrhythmia classification based on a combination network of cnn and lstm. *Biomedical Signal Processing and Control*, 57:101819, 2020.

[41] Chengsi Luo, Hongxiu Jiang, Quanchi Li, and Nini Rao. Multi-label classification of abnormalities in 12-lead ecg using 1d cnn and lstm. In *Machine Learning and Medical Engineering for Cardiovascular Health and Intravascular Imaging and Computer Assisted Stenting*, pages 55–63. Springer, 2019.

[42] Tomer Golany and Kira Radinsky. Pgans: Personalized generative adversarial networks for ecg synthesis to improve patient-specific deep ecg classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 557–564, 2019.

[43] Tomer Golany, Gal Lavee, Shai Tejman Yarden, and Kira Radinsky. Improving ecg classification using generative adversarial networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13280–13285, 2020.

[44] Pu Wang, Borui Hou, Siyu Shao, and Ruqiang Yan. Ecg arrhythmias detection using auxiliary classifier generative adversarial network and residual network. *Ieee Access*, 7:100910–100922, 2019.

[45] Sandeep Raj and Kailash Chandra Ray. Sparse representation of ecg signals for automated recognition of cardiac arrhythmias. *Expert systems with applications*, 105:49–64, 2018.

[46] MA Quiroz-Juárez, O Jiménez-Ramírez, R Vázquez-Medina, V Breña-Medina, JL Aragón, and RA Barrio. Generation of ecg signals from a reaction-diffusion model spatially discretized. *Scientific reports*, 9(1):1–10, 2019.

[47] Oliver Monfredi, Kenta Tsutsui, Bruce Ziman, Michael D Stern, Edward G Lakatta, and Victor A Maltsev. Electrophysiological heterogeneity of pacemaker cells in the rabbit intercaval region, including the sa node: insights from recording multiple ion currents in each cell. *American Journal of Physiology-Heart and Circulatory Physiology*, 314(3):H403–H414, 2018.

[48] Samadrita Bhattacharyya and Nikhil V Munshi. Development of the cardiac conduction system. *Cold Spring Harbor Perspectives in Biology*, 12(12):a037408, 2020.

[49] Alireza Fallahi, Hamidreza Ghanbari Khorram, and Alireza Kokabi. Electrocardiogram signal generation using electrical model of cardiac cell: application in cardiac ischemia. *Journal of medical engineering & technology*, 43(4):207–216, 2019.

[50] Texas Heart Institute. Conduction system. 2022. Accessed 28 Feb 2022. https://www.texasheart.org/heart-health/heart-information-center/topics/the-conduction-system/.

[51] Jonas B Nielsen, Jørgen T Kühl, Adrian Pietersen, Claus Graff, Bent Lind, Johannes J Struijk, Morten S Olesen, Moritz F Sinner, Troels N Bachmann, Stig Haunsø, et al. P-wave duration and the risk of atrial fibrillation: Results from the copenhagen ecg study. *Heart Rhythm*, 12(9):1887–1895, 2015.

[52] Niek Verweij, Irene Mateo Leach, Malou van den Boogaard, Dirk J van Veldhuisen, Vincent M Christoffels, LifeLines Cohort Study, Hans L Hillege, Wiek H van Gilst, Phil Barnett, Rudolf A de Boer, et al. Genetic determinants of p wave duration and pr segment. *Circulation: Cardiovascular Genetics*, 7(4):475–481, 2014.

[53] P Tirumala Rao, S Koteswarao Rao, G Manikanta, and S Ravi Kumar. Distinguishing normal and abnormal ecg signal. *Indian Journal of Science and Technology*, 9(10):1–5, 2016.

[54] A Peterkova and M Stremy. The raw ecg signal processing and the detection of qrs complex. In *IEEE European modelling symposium*, pages 80–85, 2015.

[55] Alex Sagie, Martin G Larson, Robert J Goldberg, James R Bengtson, and Daniel Levy. An improved method for adjusting the qt interval for heart rate (the framingham heart study). *The American journal of cardiology*, 70(7):797–801, 1992.

[56] M Riadh Arefin, Kouhyar Tavakolian, and Reza Fazel-Rezai. Qrs complex detection in ecg signal for wearable devices. In *2015 37th annual international conference of the ieee engineering in medicine and biology society (EMBC)*, pages 5940–5943. IEEE, 2015.

[57] Bong Gun Song. Electrocardiographic differential diagnosis of narrow qrs and wide qrs complex tachycardias. 2022.

[58] Merck Sharp Dohme Corp. Electrocardiography (ecg) waves. 2022. Accessed 28 Feb 2022. https://www.msdmanuals.com/en-gb/professional/multimedia/figure/electrocardiography-ecg-waves/.

[59] Nejib Zemzemi, Miguel O Bernabeu, Javier Saiz, Jonathan Cooper, Pras Pathmanathan, Gary R Mirams, Joe Pitt-Francis, and Blanca Rodriguez. Computational assessment of drug-induced effects on the electrocardiogram: from ion channel to body surface potentials. *British journal of pharmacology*, 168(3):718–733, 2013.

[60] Gerald F Fletcher, Gary J Balady, Ezra A Amsterdam, Bernard Chaitman, Robert Eckel, Jerome Fleg, Victor F Froelicher, Arthur S Leon, Ileana L Pina, Roxanne Rodney, et al. Exercise standards for testing and training: a statement for healthcare professionals from the american heart association. *Circulation*, 104(14):1694–1740, 2001.

[61] Majd AlGhatrif and Joseph Lindsay. A brief review: history to understand fundamentals of electrocardiography. *Journal of community hospital internal medicine perspectives*, 2(1):14383, 2012.

[62] Gaetano D Gargiulo. True unipolar ecg machine for wilson central terminal measurements. *BioMed research international*, 2015, 2015.

[63] Georges H Mairesse, Patrick Moran, Isabelle C Van Gelder, Christian Elsner, Marten Rosenqvist, Jonathan Mant, Amitava Banerjee, Bulent Gorenek, Johannes Brachmann, Niraj Varma, et al. Screening for atrial fibrillation: a european heart rhythm association (ehra) consensus document endorsed by the heart rhythm society (hrs), asia pacific heart rhythm society (aphrs), and sociedad latinoamericana de estimulacion cardiaca y electrofisiologia (solaece). *Ep Europace*, 19(10):1589–1623, 2017.

[64] Jason Andrade, Paul Khairy, Dobromir Dobrev, and Stanley Nattel. The clinical profile and pathophysiology of atrial fibrillation: relationships among clinical features, epidemiology, and mechanisms. *Circulation research*, 114(9):1453–1468, 2014.

[65] Syed Khairul Bashar, Eric Ding, Daniella Albuquerque, Michael Winter, Sophia Binici, Allan J Walkey, David D McManus, and Ki H Chon. Atrial fibrillation detection in icu patients: A pilot study on mimic iii data. In *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 298–301. IEEE, 2019.

[66] Nekane Larburu, T Lopetegi, and I Romero. Comparative study of algorithms for atrial fibrillation detection. In *2011 Computing in Cardiology*, pages 265–268. IEEE, 2011.

[67] Xiao Zhao, Chaofeng Sun, Miaomiao Cao, and Hao Li. Atrioventricular block can be used as a risk predictor of clinical atrial fibrillation. *Clinical Cardiology*, 42(4):452–458, 2019.

[68] Borys Surawicz, Rory Childers, Barbara J Deal, and Leonard S Gettes. Aha/accf/hrs recommendations for the standardization and interpretation of the electrocardiogram: part iii: intraventricular conduction disturbances a scientific statement from the american heart association electrocardiography and arrhythmias committee, council on clinical cardiology; the american college of cardiology foundation; and the heart rhythm society endorsed by the international society for computerized electrocardiology. *Journal of the American College of Cardiology*, 53(11):976–981, 2009.

[69] Pasquale Crea, Giuseppe Picciolo, Francesco Luzza, and Giuseppe Oreto. A three-dimensional computed model of st segment abnormality in type 1 brugada pattern: A key role of right ventricular outflow tract orientation? *Journal of Electrocardiology*, 53:31–35, 2019.

[70] Amy West Pollak. Confounders of st-elevation myocardial infarction. *Electrocardiogram in Clinical Medicine*, pages 69–74, 2020.

[71] Gaurang Nandkishor Vaidya, Steve Antoine, Syed Haider Imam, Hani Kozman, Harold Smulyan, and Daniel Villarreal. Reciprocal st-segment changes in myocardial infarction: ischemia at distance versus mirror reflection of st-elevation. *The American journal of the medical sciences*, 355(2):162–167, 2018.

[72] Jeyson Marcus Miranda, William Santos de Oliveira, Velasquez Pereira de Sá, Isabella Ferezini de Sá, and Nestor Oliveira Neto. Transient triangular qrs-st-t waveform with good outcome in a patient with left main coronary artery stenosis: a case report. *Journal of Electrocardiology*, 54:87–89, 2019.

[73] Hidekatsu Fukuta and William C Little. The cardiac cycle and the physiologic basis of left ventricular contraction, ejection, relaxation, and filling. *Heart failure clinics*, 4(1):1–11, 2008.

[74] DLRH Durrer, L Schoo, RM Schuilenburg, and HJJ Wellens. The role of premature beats in the initiation and the termination of supraventricular tachycardia in the wolff-parkinson-white syndrome. In *Professor Hein JJ Wellens*, pages 1–20. Springer, 2000.

[75] Antonio Vincenti, Roberta Brambilla, Maria Grazia Fumagalli, Rita Merola, and Stefano Pedretti. Onset mechanism of paroxysmal atrial fibrillation detected by ambulatory holter monitoring. *Europace*, 8(3):204–210, 2006.

[76] Vessela T Krasteva, Irena I Jekova, and Ivaylo I Christov. Automatic detection of premature atrial contractions in the electrocardiogram. *Electrotechniques Electronics E & E*, 9(10), 2006.

[77] Md Billal Hossain, Syed Khairul Bashar, Allan J Walkey, David D McManus, and Ki H Chon. An accurate qrs complex and p wave detection in ecg signals using complete ensemble empirical mode decomposition with adaptive noise approach. *IEEE Access*, 7:128869–128880, 2019.

[78] David G Strauss, Ronald H Selvester, and Galen S Wagner. Defining left bundle branch block in the era of cardiac resynchronization therapy. *The American journal of cardiology*, 107(6):927–934, 2011.

[79] Ramon Brugada, Josep Brugada, Charles Antzelevitch, Glenn E Kirsch, Domenico Potenza, Jeffrey A Towbin, and Pedro Brugada. Sodium channel blockers identify risk for sudden death in patients with st-segment elevation and right bundle branch block but structurally normal hearts. *Circulation*, 101(5):510–515, 2000.

[80] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[81] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[82] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[83] Charu C Aggarwal et al. Neural networks and deep learning. *Springer*, 10:978–3, 2018.

[84] Max Kuhn, Kjell Johnson, et al. *Applied predictive modeling*, volume 26. Springer, 2013.

[85] Brian D Ripley. *Pattern recognition and neural networks*. Cambridge university press, 2007.

[86] Subana Shanmuganathan. Artificial neural network modelling: An introduction. In *Artificial neural network modelling*, pages 1–14. Springer, 2016.

[87] Anders Krogh. What are artificial neural networks? *Nature biotechnology*, 26(2):195–197, 2008.

[88] Ajith Abraham. Artificial neural networks. *Handbook of measuring system design*, 2005.

[89] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[90] Heung-Il Suk. An introduction to neural networks and deep learning. In *Deep Learning for Medical Image Analysis*, pages 3–24. Elsevier, 2017.

[91] Neha Gupta et al. Artificial neural network. *Network and Complex Systems*, 3(1):24–28, 2013.

[92] Saurabh Karsoliya. Approximating number of hidden layer neurons in multiple hidden layer bpnn architecture. *International Journal of Engineering Trends and Technology*, 3(6):714–717, 2012.

[93] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

[94] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[95] Mian Mian Lau and King Hann Lim. Review of adaptive activation function in deep neural network. In *2018 IEEE-EMBS Conference on Biomedical Engineering and Sciences (IECBES)*, pages 686–690. IEEE, 2018.

[96] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323. JMLR Workshop and Conference Proceedings, 2011.

[97] Andrew L Maas, Awni Y Hannun, Andrew Y Ng, et al. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3. Citeseer, 2013.

[98] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.

[99] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[100] Christopher M Bishop et al. *Neural networks for pattern recognition*. Oxford university press, 1995.

[101] Sebastian Raschka and Vahid Mirjalili. *Python machine learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow 2*. Packt Publishing Ltd, 2019.

[102] Yi Shen. *Loss functions for binary classification and class probability estimation*. University of Pennsylvania, 2005.

[103] Elliott Gordon-Rodriguez, Gabriel Loaiza-Ganem, Geoff Pleiss, and John Patrick Cunningham. Uses and abuses of the cross-entropy loss: case studies in modern deep learning. 2020.

[104] Shie Mannor, Dori Peleg, and Reuven Rubinstein. The cross entropy method for classification. In *Proceedings of the 22nd international conference on Machine learning*, pages 561–568, 2005.

[105] Tong Wu, Qingqiu Huang, Ziwei Liu, Yu Wang, and Dahua Lin. Distribution-balanced loss for multi-label classification in long-tailed datasets. In *European Conference on Computer Vision*, pages 162–178. Springer, 2020.

[106] Jinseok Nam, Jungi Kim, Eneldo Loza Mencía, Iryna Gurevych, and Johannes Fürnkranz. Large-scale multi-label text classification—revisiting neural networks. In *Joint european conference on machine learning and knowledge discovery in databases*, pages 437–452. Springer, 2014.

[107] Pushparaja Murugan. Implementation of deep convolutional neural network in multi-class categorical image classification. *arXiv preprint arXiv:1801.01397*, 2018.

[108] Simon Du, Jason Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. In *International conference on machine learning*, pages 1675–1685. PMLR, 2019.

[109] Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on*, 14(8):2, 2012.

[110] Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[111] Yoshua Bengio. Practical recommendations for gradient-based training of deep architectures. In *Neural networks: Tricks of the trade*, pages 437–478. Springer, 2012.

[112] X Fang, H Luo, and J Tang. Structural damage detection using neural network with learning rate improvement. *Computers & structures*, 83(25-26):2150–2161, 2005.

[113] Karim Ahmed and Lorenzo Torresani. Maskconnect: Connectivity learning by gradient descent. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 349–365, 2018.

[114] Murat H Sazli. A brief review of feed-forward neural networks. *Communications Faculty of Sciences University of Ankara Series A2-A3 Physical Sciences and Engineering*, 50(01), 2006.

[115] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 6.5 back-propagation and other differentiation algorithms. *Deep learning*, pages 200–220, 2016.

[116] Zhen-Guo Che, Tzu-An Chiang, Zhen-Hua Che, et al. Feed-forward neural networks training: a comparison between genetic algorithm and back-propagation learning algorithm. *International journal of innovative computing, information and control*, 7(10):5839–5850, 2011.

[117] Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *Siam Review*, 60(2):223–311, 2018.

[118] Dominic Masters and Carlo Luschi. Revisiting small batch training for deep neural networks. *arXiv preprint arXiv:1804.07612*, 2018.

[119] Mu Li, Tong Zhang, Yuqiang Chen, and Alexander J Smola. Efficient mini-batch training for stochastic optimization. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 661–670, 2014.

[120] Jason Brownlee. What is the difference between a batch and an epoch in a neural network. *Machine Learning Mastery*, 20, 2018.

[121] Michel Jose Anzanello and Flavio Sanson Fogliatto. Learning curve models and applications: Literature review and research directions. *International Journal of Industrial Ergonomics*, 41(5):573–583, 2011.

[122] Leslie N Smith. A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay. *arXiv preprint arXiv:1803.09820*, 2018.

[123] Russell Reed and Robert J MarksII. *Neural smithing: supervised learning in feedforward artificial neural networks*. Mit Press, 1999.

[124] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016.

[125] Marcin Andrychowicz, Misha Denil, Sergio Gomez, Matthew W Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. Learning to learn by gradient descent by gradient descent. *Advances in neural information processing systems*, 29, 2016.

[126] Léon Bottou. Stochastic gradient descent tricks. In *Neural networks: Tricks of the trade*, pages 421–436. Springer, 2012.

[127] Ersan Yazan and M Fatih Talu. Comparison of the stochastic gradient descent based optimization techniques. In *2017 International Artificial Intelligence and Data Processing Symposium (IDAP)*, pages 1–5. IEEE, 2017.

[128] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147. PMLR, 2013.

[129] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

[130] T Tieleman and G Hinton. Divide the gradient by a running average of its recent magnitude. coursera: Neural networks for machine learning. *Technical Report*, 2017.

[131] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[132] Imran Khan Mohd Jais, Amelia Ritahani Ismail, and Syed Qamrun Nisa. Adam optimization algorithm for wide and deep neural network. *Knowledge Engineering and Data Science*, 2(1):41–46, 2019.

[133] Xue Ying. An overview of overfitting and its solutions. In *Journal of Physics: Conference Series*, volume 1168, page 022022. IOP Publishing, 2019.

[134] Guodong Zhang, Chaoqi Wang, Bowen Xu, and Roger Grosse. Three mechanisms of weight decay regularization. *arXiv preprint arXiv:1810.12281*, 2018.

[135] Andrew Y Ng. Feature selection, l 1 vs. l 2 regularization, and rotational invariance. In *Proceedings of the twenty-first international conference on Machine learning*, page 78, 2004.

[136] Yuchen Zhang, Jason D Lee, and Michael I Jordan. l1-regularized neural networks are improperly learnable in polynomial time. In *International Conference on Machine Learning*, pages 993–1001. PMLR, 2016.

[137] Guang Shi, Jiangshe Zhang, Huirong Li, and Changpeng Wang. Enhance the performance of deep neural networks via l2 regularization on the input of activations. *Neural Processing Letters*, 50(1):57–75, 2019.

[138] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[139] George E Dahl, Tara N Sainath, and Geoffrey E Hinton. Improving deep neural networks for lvcsr using rectified linear units and dropout. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 8609–8613. IEEE, 2013.

[140] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[141] Nikhil Ketkar and Eder Santana. *Deep learning with Python*, volume 1. Springer, 2017.

[142] Kenji Suzuki. Overview of deep learning in medical imaging. *Radiological physics and technology*, 10(3):257–273, 2017.

[143] Suresh Dara, Priyanka Tumma, Nageswara Rao Eluri, and Gangadhara Rao Kancharla. Feature extraction in medical images by using deep learning approach. *Int. J. Pure Appl. Math*, 120(6):305–312, 2018.

[144] Hoo-Chang Shin, Holger R Roth, Mingchen Gao, Le Lu, Ziyue Xu, Isabella Nogues, Jianhua Yao, Daniel Mollura, and Ronald M Summers. Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. *IEEE transactions on medical imaging*, 35(5):1285–1298, 2016.

[145] Vincent Dumoulin and Francesco Visin. A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*, 2016.

[146] Rikiya Yamashita, Mizuho Nishio, Richard Kinh Gian Do, and Kaori Togashi. Convolutional neural networks: an overview and application in radiology. *Insights into imaging*, 9(4):611–629, 2018.

[147] Hieu Pham, Melody Guan, Barret Zoph, Quoc Le, and Jeff Dean. Efficient neural architecture search via parameters sharing. In *International conference on machine learning*, pages 4095–4104. PMLR, 2018.

[148] Coenraad Mouton, Johannes C Myburgh, and Marelie H Davel. Stride and translation invariance in cnns. In *Southern African Conference for Artificial Intelligence Research*, pages 267–281. Springer, 2021.

[149] Dominik Scherer, Andreas Müller, and Sven Behnke. Evaluation of pooling operations in convolutional architectures for object recognition. In *International conference on artificial neural networks*, pages 92–101. Springer, 2010.

[150] Fred H Hamker. Predictions of a model of spatial attention using sum-and max-pooling functions. *Neurocomputing*, 56:329–343, 2004.

[151] Chen-Yu Lee, Patrick W Gallagher, and Zhuowen Tu. Generalizing pooling functions in convolutional neural networks: Mixed, gated, and tree. *arXiv preprint arXiv:1509.08985*, 2015.

[152] SH Shabbeer Basha, Shiv Ram Dubey, Viswanath Pulabaigari, and Snehasis Mukherjee. Impact of fully connected layers on performance of convolutional neural networks for image classification. *Neurocomputing*, 378:112–119, 2020.

[153] Larry R Medsker and LC Jain. Recurrent neural networks. *Design and Applications*, 5:64–67, 2001.

[154] Tehseen Zia and Usman Zahid. Long short-term memory recurrent neural network architectures for urdu acoustic modeling. *International Journal of Speech Technology*, 22(1):21–30, 2019.

[155] Imon Banerjee, Yuan Ling, Matthew C Chen, Sadid A Hasan, Curtis P Langlotz, Nathaniel Moradzadeh, Brian Chapman, Timothy Amrhein, David Mong, Daniel L Rubin, et al. Comparative effectiveness of convolutional neural network (cnn) and recurrent neural network (rnn) architectures for radiology text report classification. *Artificial intelligence in medicine*, 97:79–88, 2019.

[156] Navin Kumar Manaswi. Rnn and lstm. In *Deep Learning with Applications Using Python*, pages 115–126. Springer, 2018.

[157] Apeksha Shewalkar. Performance evaluation of deep neural networks applied to speech recognition: Rnn, lstm and gru. *Journal of Artificial Intelligence and Soft Computing Research*, 9(4):235–245, 2019.

[158] Jiang Guo. Backpropagation through time. *Unpubl. ms., Harbin Institute of Technology*, 40:1–6, 2013.

[159] Jan Koutnik, Klaus Greff, Faustino Gomez, and Juergen Schmidhuber. A clockwork rnn. In *International Conference on Machine Learning*, pages 1863–1871. PMLR, 2014.

[160] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[161] Alex Graves. Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, pages 37–45, 2012.

[162] Kamilya Smagulova and Alex Pappachen James. A survey on lstm memristive neural network architectures and applications. *The European Physical Journal Special Topics*, 228(10):2313–2324, 2019.

[163] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681, 1997.

[164] Alex Graves, Navdeep Jaitly, and Abdel-rahman Mohamed. Hybrid speech recognition with deep bidirectional lstm. In *2013 IEEE workshop on automatic speech recognition and understanding*, pages 273–278. IEEE, 2013.

[165] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*, 2015.

[166] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

[167] Yuan Gao and Dorota Glowacka. Deep gate recurrent neural network. In *Asian conference on machine learning*, pages 350–365. PMLR, 2016.

[168] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[169] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[170] Andrei Vladimirovich Sozykin. An overview of methods for deep learning in neural networks. *Vestnik Yuzhno-Ural'skogo Gosudarstvennogo Universiteta. Seriya" Vychislitelnaya Matematika i Informatika"*, 6(3):28–59, 2017.

[171] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[172] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.

[173] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2017.

[174] Worku Jifara, Feng Jiang, Seungmin Rho, Maowei Cheng, and Shaohui Liu. Medical image denoising using convolutional neural network: a residual learning approach. *The Journal of Supercomputing*, 75(2):704–718, 2019.

[175] Muhammad Shehzad Hanif and Muhammad Bilal. Competitive residual neural network for image classification. *ICT Express*, 6(1):28–37, 2020.

[176] Francois Chollet. *Deep learning with Python*. Simon and Schuster, 2021.

[177] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. {TensorFlow}: A system for {Large-Scale} machine learning. In *12th USENIX symposium on operating systems design and implementation (OSDI 16)*, pages 265–283, 2016.

[178] Aurélien Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems.* " O'Reilly Media, Inc.", 2019.

[179] Google. Colaboratory: Frequently asked questions.

[180] N.V. Thakor and Y.-S. Zhu. Applications of adaptive filtering to ecg analysis: noise cancellation and arrhythmia detection. *IEEE Transactions on Biomedical Engineering*, 38(8):785–794, 1991.

[181] A. Gotchev, N. Nikolaev, and K. Egiazarian. Improving the transform domain ecg denoising performance by applying interbeat and intra-beat decorrelating transforms. In *ISCAS 2001. The 2001 IEEE International Symposium on Circuits and Systems (Cat. No.01CH37196)*, volume 2, pages 17–20 vol. 2, 2001.

[182] B.-U. Kohler, C. Hennig, and R. Orglmeister. The principles of software qrs detection. *IEEE Engineering in Medicine and Biology Magazine*, 21(1):42–57, 2002.

[183] Association for the Advancement of Medical Instrumentation. American national standard for ambulatory electrocardiographs, publication ansi. *AAMI*, EC38, 1994.

[184] Norman J. Holter. New method for heart studies. *Science*, 134(3486):1214–1220, 1961.

[185] ES.A. El-Dahshan. Genetic algorithm and wavelet hybrid scheme for ecg signal denoising. *Telecommun Syst 46*, page 209–215, 2011.

[186] A. Illanes-Manriquez and Q. Zhang. An algorithm for robust detection of qrs onset and offset in ecg signals. In *2008 Computers in Cardiology*, pages 857–860, 2008.

[187] Jiapu Pan and Willis J. Tompkins. A real-time qrs detection algorithm. *IEEE Transactions on Biomedical Engineering*, BME-32(3):230–236, 1985.

[188] Cuiwei Li, Chongxun Zheng, and Changfeng Tai. Detection of ecg characteristic points using wavelet transforms. *IEEE Transactions on Biomedical Engineering*, 42(1):21–28, 1995.

[189] Boudreaux-Bartels GF. Kadambe S, Murray R. Applications of adaptive filtering to ecg analysis: noise cancellation and arrhythmia detection. *IEEE Transactions on Biomedical Engineering*, 46(7):838–48, 1999.

[190] Detection of ecg characteristic points using multiresolution wavelet analysis based selective coefficient method. *Measurement*, 43(2):255–261, 2010.

[191] Amit J. Nimunkar and Willis J. Tompkins. R-peak detection and signal averaging for simulated stress ecg using emd. In *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 1261–1264, 2007.

[192] Zine-Eddine Hadj Slimane and Amine Naït-Ali. Qrs complex detection using empirical mode decomposition. *Digital Signal Processing*, 20(4):1221–1228, 2010.

[193] Yu Hen Hu, S. Palreddy, and W.J. Tompkins. A patient-adaptable ecg beat classifier using a mixture of experts approach. *IEEE Transactions on Biomedical Engineering*, 44(9):891–900, 1997.

[194] Xu-L. Jordan, M. I. Convergence properties of the em approach to learning in mixture-of-experts architectures. *Dept. Brain and Cognitive Sci., MIT, Cambridge,MA, Tech*, 1993.

[195] Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. Adaptive mixtures of local experts. *Neural Computation*, 3(1):79–87, 1991.

[196] Robert A. Jacobs. Methods for combining experts' probability assessments. *Neural Computation*, 7(5):867–888, 1995.

[197] Sheng wei Fei. Diagnostic study on arrhythmia cordis based on particle swarm optimization-based support vector machine. *Expert Systems with Applications*, 37(10):6748–6752, 2010.

[198] Juyoung Park and Kyungtae Kang. Pchd: Personalized classification of heartbeat types using a decision tree. *Computers in Biology and Medicine*, 54:79–88, 2014.

[199] Mariano Llamedo and Juan Pablo Martinez. An automatic patient-adapted ecg heartbeat classifier allowing expert assistance. *IEEE Transactions on Biomedical Engineering*, 59(8):2312–2320, 2012.

[200] Sandeep Raj, Kailash Chandra Ray, and Om Shankar. Cardiac arrhythmia beat classification using dost and pso tuned svm. *Computer Methods and Programs in Biomedicine*, 136:163–177, 2016.

[201] Babak Mohammadzadeh Asl, Seyed Kamaledin Setarehdan, and Maryam Mohebbi. Support vector machine-based arrhythmia classification using reduced features of heart rate variability signal. *Artificial Intelligence in Medicine*, 44(1):51–64, 2008.

[202] G. Baudat and F. Anouar. Generalized Discriminant Analysis Using a Kernel Approach. *Neural Computation*, 12(10):2385–2404, 10 2000.

[203] Rajni R. Marwaha A Kaur, I. Ecg signal analysis and arrhythmia detection using wavelet transform. *J. Inst. Eng. India Ser*, B(97):499–507, 2016.

[204] Turker Ince*, Serkan Kiranyaz, and Moncef Gabbouj. A generic and robust system for automated patient-specific classification of ecg signals. *IEEE Transactions on Biomedical Engineering*, 56(5):1415–1426, 2009.

[205] Cardiac decision making using higher order spectra. *Biomedical Signal Processing and Control*, 8(2):193–203, 2013.

[206] Omar Behadada and Mohammed Amine Chikh. An interpretable classifier for detection of cardiac arrhythmias by using the fuzzy decision tree. *Artif. Intell. Res.*, 2(3):45–58, 2013.

[207] Wai Kei Lei, Bing Nan Li, Ming Chui Dong, and Mang I Vai. Afc-ecg: An adaptive fuzzy ecg classifier. In *Soft computing in industrial applications*, pages 189–199. Springer, 2007.

[208] Mohammad Reza Homaeinezhad, Ehsan Tavakkoli, and Ali Ghaffari. Discrete wavelet-based fuzzy network architecture for ecg rhythm-type recognition: feature extraction and clustering-oriented tuning of fuzzy inference system. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, 4(3):107–130, 2011.

[209] Muhammad Arif, Muhammad Usman Akram, et al. Pruned fuzzy k-nearest neighbor classifier for beat classification. *Journal of Biomedical Science and Engineering*, 3(04):380, 2010.

[210] Roshan Joy Martis, U Rajendra Acharya, KM Mandana, Ajoy Kumar Ray, and Chandan Chakraborty. Application of principal component analysis to ecg signals for automated diagnosis of cardiac health. *Expert Systems with Applications*, 39(14):11792–11800, 2012.

[211] Siao Zheng Bong, M Murugappan, and Sazali Yaacob. Analysis of electrocardiogram (ecg) signals for human emotional stress classification. In *International Conference on Intelligent Robotics, Automation, and Manufacturing*, pages 198–205. Springer, 2012.

[212] Siti Agrippina Alodia Yusuf and Risanuri Hidayat. Mfcc feature extraction and

knn classification in ecg signals. In *2019 6th International Conference on Information Technology, Computer and Electrical Engineering (ICITACEE)*, pages 1–5. IEEE, 2019.

[213] Mohammad Reza Homaeinezhad, Seyyed Abbas Atyabi, E Tavakkoli, Hamid Najjaran Toosi, Ali Ghaffari, and Reza Ebrahimpour. Ecg arrhythmia recognition via a neuro-svm–knn hybrid classifier with virtual qrs image-based geometrical features. *Expert Systems with Applications*, 39(2):2047–2058, 2012.

[214] Leigang Zhang, Hu Peng, and Chenglong Yu. An approach for ecg classification based on wavelet feature extraction and decision tree. In *2010 International Conference on Wireless Communications & Signal Processing (WCSP)*, pages 1–4. IEEE, 2010.

[215] LV Kumari, Y Padma Sai, et al. Classification of ecg beats using optimized decision tree and adaptive boosted optimized decision tree. *Signal, Image and Video Processing*, 16(3):695–703, 2022.

[216] Smita L Kasar and Madhuri S Joshi. Analysis of multi-lead ecg signals using decision tree algorithms. *International Journal of Computer Applications*, 134(16), 2016.

[217] Zhanquan Sun, Chaoli Wang, Yangyang Zhao, and Chao Yan. Multi-label ecg signal classification based on ensemble classifier. *IEEE Access*, 8:117986–117996, 2020.

[218] Yuwen Li, Zhimin Zhang, Fan Zhou, Yantao Xing, Jianqing Li, and Chengyu Liu. Multi-label feature selection for long-term electrocardiogram signals. In *2020 International Conference on Sensing, Measurement & Data Analytics in the era of Artificial Intelligence (ICSMD)*, pages 335–340. IEEE, 2020.

[219] Giovanna Sannino and Giuseppe De Pietro. A deep learning approach for ecg-based heartbeat classification for arrhythmia detection. *Future Generation Computer Systems*, 86:446–455, 2018.

[220] Ranjana D Raut and Sanjay V Dudul. Arrhythmias classification with mlp

neural network and statistical analysis. In *2008 First International Conference on Emerging Trends in Engineering and Technology*, pages 553–558. IEEE, 2008.

[221] Bahareh Pourbabaee, Mehrsan Javan Roshtkhari, and Khashayar Khorasani. Deep convolutional neural networks and learning ecg features for screening paroxysmal atrial fibrillation patients. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 48(12):2095–2104, 2018.

[222] Arlene John, Barry Cardiff, and Deepu John. A 1d-cnn based deep learning technique for sleep apnea detection in iot sensors. In *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5. IEEE, 2021.

[223] U Rajendra Acharya, Hamido Fujita, Shu Lih Oh, Yuki Hagiwara, Jen Hong Tan, Muhammad Adam, and Ru San Tan. Deep convolutional neural network for the automated diagnosis of congestive heart failure using ecg signals. *Applied Intelligence*, 49(1):16–27, 2019.

[224] U Rajendra Acharya, Hamido Fujita, Shu Lih Oh, Yuki Hagiwara, Jen Hong Tan, and Muhammad Adam. Application of deep convolutional neural network for automated detection of myocardial infarction using ecg signals. *Information Sciences*, 415:190–198, 2017.

[225] Awni Y Hannun, Pranav Rajpurkar, Masoumeh Haghpanahi, Geoffrey H Tison, Codie Bourn, Mintu P Turakhia, and Andrew Y Ng. Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network. *Nature medicine*, 25(1):65–69, 2019.

[226] Yong Xia, Naren Wulan, Kuanquan Wang, and Henggui Zhang. Detecting atrial fibrillation by deep convolutional neural networks. *Computers in biology and medicine*, 93:84–92, 2018.

[227] Md Rashed-Al-Mahfuz, Mohammad Ali Moni, Pietro Lio, Sheikh Mohammed Shariful Islam, Shlomo Berkovsky, Matloob Khushi, and Julian MW Quinn. Deep convolutional neural networks based ecg beats classification to diagnose cardiovascular conditions. *Biomedical engineering letters*, 11(2):147–162, 2021.

[228] Aykut Diker, Zafer Cömert, Engin Avcı, Mesut Toğaçar, and Burhan Ergen. A novel application based on spectrogram and convolutional neural network for ecg classification. In *2019 1st International Informatics and Software Engineering Conference (UBMYK)*, pages 1–6. IEEE, 2019.

[229] Jingshan Huang, Binqiang Chen, Bin Yao, and Wangpeng He. Ecg arrhythmia classification using stft-based spectrogram and convolutional neural network. *IEEE access*, 7:92871–92880, 2019.

[230] Awais M Lodhi, Adnan N Qureshi, Usman Sharif, and Zahid Ashiq. A novel approach using voting from ecg leads to detect myocardial infarction. In *Proceedings of SAI Intelligent Systems Conference*, pages 337–352. Springer, 2018.

[231] Ulas Baran Baloglu, Muhammed Talo, Ozal Yildirim, Ru San Tan, and U Rajendra Acharya. Classification of myocardial infarction with multi-lead ecg signals and deep cnn. *Pattern Recognition Letters*, 122:23–30, 2019.

[232] Yeonghyeon Park, Il Dong Yun, and Si-Hyuck Kang. Preprocessing method for performance enhancement in cnn-based stemi detection from 12-lead ecg. *IEEE Access*, 7:99964–99977, 2019.

[233] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[234] D Pawade, A Sakhapara, Mansi Jain, Neha Jain, and Krushi Gada. Story scrambler-automatic text generation using word level rnn-lstm. *International Journal of Information Technology and Computer Science (IJITCS)*, 10(6):44–53, 2018.

[235] Yajie Miao, Mohammad Gowayyed, and Florian Metze. Eesen: End-to-end speech recognition using deep rnn models and wfst-based decoding. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 167–174. IEEE, 2015.

[236] Chenshuang Zhang, Guijin Wang, Jingwei Zhao, Pengfei Gao, Jianping Lin, and Huazhong Yang. Patient-specific ecg classification based on recurrent neural networks and clustering technique. In *2017 13th IASTED International Conference on Biomedical Engineering (BioMed)*, pages 63–67. IEEE, 2017.

[237] Özal Yildirim. A novel wavelet sequence based on deep bidirectional lstm network model for ecg signal classification. *Computers in biology and medicine*, 96:189–202, 2018.

[238] Oliver Faust, Alex Shenfield, Murtadha Kareem, Tan Ru San, Hamido Fujita, and U Rajendra Acharya. Automated detection of atrial fibrillation using long short-term memory network with rr interval signals. *Computers in biology and medicine*, 102:327–335, 2018.

[239] Shraddha Singh, Saroj Kumar Pandey, Urja Pawar, and Rekh Ram Janghel. Classification of ecg arrhythmia using recurrent neural networks. *Procedia computer science*, 132:1290–1297, 2018.

[240] VG Sujadevi, KP Soman, and R Vinayakumar. Real-time detection of atrial fibrillation from short time single lead ecg traces using recurrent neural networks. In *The International Symposium on Intelligent Systems Technologies and Applications*, pages 212–221. Springer, 2017.

[241] Rasmus S Andersen, Abdolrahman Peimankar, and Sadasivan Puthusserypady. A deep learning approach for real-time detection of atrial fibrillation. *Expert Systems with Applications*, 115:465–473, 2019.

[242] Georgios Petmezas, Kostas Haris, Leandros Stefanopoulos, Vassilis Kilintzis, Andreas Tzavelis, John A Rogers, Aggelos K Katsaggelos, and Nicos Maglaveras. Automated atrial fibrillation detection using a hybrid cnn-lstm network on imbalanced ecg datasets. *Biomedical Signal Processing and Control*, 63:102194, 2021.

[243] Aiyun Chen, Fei Wang, Wenhan Liu, Sheng Chang, Hao Wang, Jin He, and Qijun Huang. Multi-information fusion neural networks for arrhythmia automatic detection. *Computer methods and programs in biomedicine*, 193:105479, 2020.

[244] Qihang Yao, Xiaomao Fan, Yunpeng Cai, Ruxin Wang, Liyan Yin, and Ye Li. Time-incremental convolutional neural network for arrhythmia detection in varied-length electrocardiogram. In *2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech*, pages 754–761. IEEE, 2018.

[245] Tsai-Min Chen, Chih-Han Huang, Edward SC Shih, Yu-Feng Hu, and Ming-Jing Hwang. Detection and classification of cardiac arrhythmias by a challenge-best deep learning neural network model. *Iscience*, 23(3):100886, 2020.

[246] Runnan He, Yang Liu, Kuanquan Wang, Na Zhao, Yongfeng Yuan, Qince Li, and Henggui Zhang. Automatic cardiac arrhythmia classification using combination of deep residual network and bidirectional lstm. *IEEE Access*, 7:102119–102135, 2019.

[247] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[248] Adyasha Rath, Debahuti Mishra, Ganapati Panda, and Suresh Chandra Satapathy. Heart disease detection using deep learning methods from imbalanced ecg samples. *Biomedical Signal Processing and Control*, 68:102820, 2021.

[249] Fei Zhu, Fei Ye, Yuchen Fu, Quan Liu, and Bairong Shen. Electrocardiogram generation with a bidirectional lstm-cnn generative adversarial network. *Scientific reports*, 9(1):1–11, 2019.

[250] Constantinos H Papadopoulos, Dimitrios Oikonomidis, Efstathios Lazaris, and Petros Nihoyannopoulos. Echocardiography and cardiac arrhythmias. *Hellenic Journal of Cardiology*, 59(3):140–149, 2018.

[251] David O Arnar, Georges H Mairesse, Giuseppe Boriani, Hugh Calkins, Ashley Chin, Andrew Coats, Jean-Claude Deharo, Jesper Hastrup Svendsen, Hein Heidbüchel, Rodrigo Isa, et al. Management of asymptomatic arrhythmias: a

european heart rhythm association (ehra) consensus document, endorsed by the heart failure association (hfa), heart rhythm society (hrs), asia pacific heart rhythm society (aphrs), cardiac arrhythmia society of southern africa (cassa), and latin america heart rhythm society (lahrs). *EP Europace*, 21(6):844–845, 2019.

[252] Allam Jaya Prakash and Samit Ari. A system for automatic cardiac arrhythmia recognition using electrocardiogram signal. In *Bioelectronics and Medical Devices*, pages 891–911. Elsevier, 2019.

[253] Mihaela Porumb, Saverio Stranges, Antonio Pescapè, and Leandro Pecchia. Precision medicine and artificial intelligence: a pilot study on deep learning for hypoglycemic events detection based on ecg. *Scientific reports*, 10(1):1–16, 2020.

[254] Saeed Saadatnejad, Mohammadhosein Oveisi, and Matin Hashemi. Lstm-based ecg classification for continuous monitoring on personal wearable devices. *IEEE journal of biomedical and health informatics*, 24(2):515–523, 2019.

[255] Shrimanti Ghosh, Ankur Banerjee, Nilanjan Ray, Peter W Wood, Pierre Boulanger, and Raj Padwal. Continuous blood pressure prediction from pulse transit time using ecg and ppg signals. In *2016 IEEE Healthcare Innovation Point-Of-Care Technologies Conference (HI-POCT)*, pages 188–191. IEEE, 2016.

[256] Ö Yildirim and Pławiak Paweł and Tan. R.-s. & acharya, ur arrhythmia detection using deep convolutional neural network with long duration ecg signals. *Comput. Biol. Med*, 102:411–420, 2018.

[257] Szu-Hsien Chou, Kuan-Yu Lin, Zhen-Ye Chen, Chun-Jung Juan, Chien-Yi Ho, and Tzu-Ching Shih. Integrating patient-specific electrocardiogram signals and image-based computational fluid dynamics method to analyze coronary blood flow in patients during cardiac arrhythmias. *Journal of Medical and Biological Engineering*, 40(2):264–272, 2020.

[258] Antônio H Ribeiro, Manoel Horta Ribeiro, Gabriela MM Paixão, Derick M Oliveira, Paulo R Gomes, Jéssica A Canazart, Milton PS Ferreira, Carl R Andersson, Peter W Macfarlane, Wagner Meira Jr, et al. Automatic diagnosis of

the 12-lead ecg using a deep neural network. *Nature communications*, 11(1):1–9, 2020.

[259] Hongling Zhu, Cheng Cheng, Hang Yin, Xingyi Li, Ping Zuo, Jia Ding, Fan Lin, Jingyi Wang, Beitong Zhou, Yonge Li, et al. Automatic multilabel electrocardiogram diagnosis of heart rhythm or conduction abnormalities with deep learning: a cohort study. *The Lancet Digital Health*, 2(7):e348–e357, 2020.

[260] Stanislaw Osowski, Krzysztof Siwek, and Tomasz Markiewicz. Mlp and svm networks-a comparative study. In *Proceedings of the 6th Nordic Signal Processing Symposium, 2004. NORSIG 2004.*, pages 37–40. IEEE, 2004.

[261] Noman Naseer and Hammad Nazeer. Classification of normal and abnormal ecg signals based on their pqrst intervals. In *2017 International Conference on Mechanical, System and Control Engineering (ICMSC)*, pages 388–391. IEEE, 2017.

[262] Zachary C Lipton, David C Kale, Charles Elkan, and Randall Wetzel. Learning to diagnose with lstm recurrent neural networks. *arXiv preprint arXiv:1511.03677*, 2015.

[263] R Bousseljot, D Kreiseler, and A Schnabel. Nutzung der ekg-signaldatenbank cardiodat der ptb über das internet. 1995.

[264] Udit Satija, Barathram Ramkumar, and M Sabarimalai Manikandan. Noise-aware dictionary-learning-based sparse representation framework for detection and removal of single and combined noises from ecg signal. *Healthcare technology letters*, 4(1):2–12, 2017.

[265] Harshita Gupta and Divya Gupta. Lpc and lpcc method of feature extraction in speech recognition system. In *2016 6th International Conference-Cloud System and Big Data Engineering (Confluence)*, pages 498–502. IEEE, 2016.

[266] Tao Chen, Ruifeng Xu, Yulan He, and Xuan Wang. Improving sentiment analysis via sentence type classification using bilstm-crf and cnn. *Expert Systems with Applications*, 72:221–230, 2017.

[267] George E Dahl, Dong Yu, Li Deng, and Alex Acero. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on audio, speech, and language processing*, 20(1):30–42, 2011.

[268] Haibo He and Yunqian Ma. Imbalanced learning: foundations, algorithms, and applications. 2013.

[269] Rhys Heffernan, Yuedong Yang, Kuldip Paliwal, and Yaoqi Zhou. Capturing non-local interactions by long short-term memory bidirectional recurrent neural networks for improving prediction of protein secondary structure, backbone angles, contact numbers and solvent accessibility. *Bioinformatics*, 33(18):2842–2849, 2017.

[270] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.

[271] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

[272] Julià Camps, Blanca Rodríguez, and Ana Mincholé. Deep learning based qrs multilead delineator in electrocardiogram signals. In *2018 Computing in Cardiology Conference (CinC)*, volume 45, pages 1–4. IEEE, 2018.

[273] M Hashemi. Enlarging smaller images before inputting into convolutional neural network: zero-padding vs. interpolation. j big data 6, 98 (2019), 2019.

[274] AV Klimov, VG Glavnyi, GV Bakakin, and VG Meledin. Spectral method for processing signals of a high-accuracy laser radar. *Optoelectronics, Instrumentation and Data Processing*, 52(6):563–569, 2016.

[275] Huaqing Wang, Shi Li, Liuyang Song, and Lingli Cui. A novel convolutional neural network based fault recognition method via image fusion of multi-vibration-signals. *Computers in Industry*, 105:182–190, 2019.

[276] Pranav Rajpurkar, Awni Y Hannun, Masoumeh Haghpanahi, Codie Bourn, and Andrew Y Ng. Cardiologist-level arrhythmia detection with convolutional neural networks. *arXiv preprint arXiv:1707.01836*, 2017.

[277] Feifei Liu, Chengyu Liu, Lina Zhao, Xiangyu Zhang, Xiaoling Wu, Xiaoyan Xu, Yulin Liu, Caiyun Ma, Shoushui Wei, Zhiqiang He, et al. An open access database for evaluating the algorithms of electrocardiogram rhythm and morphology abnormality detection. *Journal of Medical Imaging and Health Informatics*, 8(7):1368–1373, 2018.

[278] Patrick Wagner, Nils Strodthoff, Ralf-Dieter Bousseljot, Dieter Kreiseler, Fatima I Lunze, Wojciech Samek, and Tobias Schaeffter. Ptb-xl, a large publicly available electrocardiography dataset. *Scientific data*, 7(1):1–15, 2020.

[279] George B Moody and Roger G Mark. The impact of the mit-bih arrhythmia database. *IEEE Engineering in Medicine and Biology Magazine*, 20(3):45–50, 2001.

[280] Min-Ling Zhang and Zhi-Hua Zhou. A review on multi-label learning algorithms. *IEEE transactions on knowledge and data engineering*, 26(8):1819–1837, 2013.

[281] Jiang Wang, Yi Yang, Junhua Mao, Zhiheng Huang, Chang Huang, and Wei Xu. Cnn-rnn: A unified framework for multi-label image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2285–2294, 2016.

[282] Georgia Sarquella-Brugada, Sergi Cesar, Maria D Zambrano, Anna Fernandez-Falgueras, Victoria Fiol, Anna Iglesias, Francesc Torres, Oscar Garcia-Algar, Elena Arbelo, Josep Brugada, et al. Electrocardiographic assessment and genetic analysis in neonates: a current topic of discussion. *Current Cardiology Reviews*, 15(1):30–37, 2019.

[283] Ziliang Song, Kai Xu, Xiaofeng Hu, Weifeng Jiang, Shaohui Wu, Mu Qin, and Xu Liu. A study of cardiogenic stroke risk in non-valvular atrial fibrillation patients. *Frontiers in Cardiovascular Medicine*, page 246, 2020.

[284] Gary Tse, Sharen Lee, Andrew Li, Dong Chang, Guangping Li, Jiandong Zhou, Tong Liu, and Qingpeng Zhang. Automated electrocardiogram analysis identifies novel predictors of ventricular arrhythmias in brugada syndrome. *Frontiers in cardiovascular medicine*, page 399, 2021.

[285] Junxian Cai, Weiwei Sun, Jianfeng Guan, and Ilsun You. Multi-ecgnet for ecg arrythmia multi-label classification. *IEEE Access*, 8:110848–110858, 2020.

[286] Eedara Prabhakararao and Samarendra Dandapat. Myocardial infarction severity stages classification from ecg signals using attentional recurrent neural network. *IEEE Sensors Journal*, 20(15):8711–8720, 2020.

[287] Glenn Van Steenkiste, Gunther van Loon, and Guillaume Crevecoeur. Transfer learning in ecg classification from human to horse using a novel parallel neural network architecture. *Scientific Reports*, 10(1):1–12, 2020.

[288] Manuel Franco, Richard S Cooper, Usama Bilal, and Valentín Fuster. Challenges and opportunities for cardiovascular disease prevention. *The American journal of medicine*, 124(2):95–102, 2011.

[289] Zafar M Yuldashev, Anatoli P Nemirko, Evgeny N Mikhaylov, Dmitry S Lebedev, Aleksei A Anisimov, Alena I Skorobogatova, and Darina S Ripka. Prediction of Local Abnormal Ventricular Myocardial Electrical Activation on Surface ECG in Patients with Structural Heart Disease. In *BIODEVICES*, pages 395–401, 2020.

[290] V. Jahmunah, Shu Lih Oh, Joel Koh En Wei, Edward J Ciaccio, Kuang Chua, Tan Ru San, and U. Rajendra Acharya. Computer-aided diagnosis of congestive heart failure using ecg signals – a review. *Physica Medica*, 62:95–104, 2019.

[291] Gregory Y H Lip, Tina D Hunter, Maria E Quiroz, Paul D Ziegler, and Mintu P Turakhia. Atrial fibrillation diagnosis timing, ambulatory ECG monitoring utilization, and risk of recurrent stroke. *Circulation: Cardiovascular Quality and Outcomes*, 10(1):e002864, 2017.

[292] Saeed Mian Qaisar and Abdulhamit Subasi. An adaptive rate ECG acquisition and analysis for efficient diagnosis of the cardiovascular diseases. In *2018 IEEE 3rd International Conference on Signal and Image Processing (ICSIP)*, pages 177–181. IEEE, 2018.

[293] Zahra Ebrahimi, Mohammad Loni, Masoud Daneshtalab, and Arash Gharehbaghi. A review on deep learning methods for ecg arrhythmia classification. *Expert Systems with Applications: X*, 7:100033, 2020.

[294] Fatma Murat, Ozal Yildirim, Muhammed Talo, Ulas Baran Baloglu, Yakup Demir, and U Rajendra Acharya. Application of deep learning techniques for heartbeats detection using ECG signals-analysis and review. *Computers in biology and medicine*, page 103726, 2020.

[295] Aidan N Gomez, Mengye Ren, Raquel Urtasun, and Roger B Grosse. The reversible residual network: Backpropagation without storing activations. *Advances in neural information processing systems*, 30, 2017.

[296] Kazım Hanbay. Deep neural network based approach for ecg classification using hybrid differential features and active learning. *IET Signal Processing*, 13(2):165–175, 2019.

[297] Zhe Sun, Raymond Chiong, and Zheng-ping Hu. Self-adaptive feature learning based on a priori knowledge for facial expression recognition. *Knowledge-Based Systems*, 204:106124, 2020.

[298] Jordan Ubbens, Mikolaj Cieslak, Przemyslaw Prusinkiewicz, and Ian Stavness. The use of plant models in deep learning: an application to leaf counting in rosette plants. *Plant methods*, 14(1):1–10, 2018.

[299] Jacob Goldberger, Geoffrey E Hinton, Sam Roweis, and Russ R Salakhutdinov. Neighbourhood components analysis. *Advances in neural information processing systems*, 17, 2004.

[300] Paul Rubel, Danilo Pani, Alois Schloegl, Jocelyne Fayn, Fabio Badilini, Peter W Macfarlane, and Alpo Varri. SCP-ECG V3. 0: An enhanced standard communication protocol for computer-assisted electrocardiography. In *2016 Computing in Cardiology Conference (CinC)*, pages 309–312. IEEE, 2016.

[301] Manas Rakshit and Susmita Das. An efficient ECG denoising methodology using empirical mode decomposition and adaptive switching mean filter. *Biomedical signal processing and control*, 40:140–148, 2018.

[302] Saeed Mian Qaisar and Dominique Dallet. ECG Noise Removal and Efficient Arrhythmia Identification Based on Effective Signal-Piloted Processing and Machine Learning. In *2021 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, pages 1–6. IEEE, 2021.

[303] Achmad Fanany Onnilita Gaffar, Rheo Malani, Agusma Wajiansyah, Arief Bramanto Wicaksono Putra, et al. A multi-frame blocking for signal segmentation in voice command recognition. In *2020 International Seminar on Intelligent Technology and Its Applications (ISITIA)*, pages 299–304. IEEE, 2020.

[304] Rajeev Ranjan and Abhishek Thakur. Analysis of feature extraction techniques for speech recognition system. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, 8(7C2), 2019.

[305] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[306] Rushi Lan and Yicong Zhou. An extended probabilistic collaborative representation based classifier for image classification. In *2017 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1392–1397, 2017.

[307] Bharat Richhariya and Muhammad Tanveer. EEG signal classification using universum support vector machine. *Expert Systems with Applications*, 106:169–182, 2018.

[308] Ivona Milanova, Ksenija Sarvanoska, Viktor Srbinoski, and Hristijan Gjoreski. Automatic text generation in macedonian using recurrent neural networks. In *International Conference on ICT Innovations*, pages 1–12. Springer, 2019.

[309] François Chollet et al. Keras https://keras. io. *Go to reference in article*, 2015.

[310] P Nejedly, A Ivora, I Viscor, J Halamek, P Jurak, and F Plesinger. Utilization of Residual CNN-GRU With Attention Mechanism for Classification of 12-lead ECG. In *2020 Computing in Cardiology*, pages 1–4, 2020.

[311] Laurens van der Maaten and Geoffrey Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.

[312] Leonid Karlinsky, Joseph Shtok, Sivan Harary, Eli Schwartz, Amit Aides, Rogerio Feris, Raja Giryes, and Alex M. Bronstein. Repmet: Representative-based metric learning for classification and few-shot object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[313] J Lu, J Hu, and J Zhou. Deep Metric Learning for Visual Understanding: An Overview of Recent Advances. *IEEE Signal Processing Magazine*, 34(6):76–84, 2017.

[314] Andrew Zhai and Hao-Yu Wu. Classification is a strong baseline for deep metric learning. *arXiv preprint arXiv:1811.12649*, 2018.

[315] Chuang Han and Li Shi. Ml–resnet: A novel network to detect and locate myocardial infarction using 12 leads ecg. *Computer methods and programs in biomedicine*, 185:105138, 2020.

[316] Pengyi Hao, Xiang Gao, Zhihe Li, Jinglin Zhang, Fuli Wu, and Cong Bai. Multi-branch fusion network for myocardial infarction screening from 12-lead ecg images. *Computer methods and programs in biomedicine*, 184:105286, 2020.

[317] Jia Li, Yujuan Si, Tao Xu, and Saibiao Jiang. Deep convolutional neural network based ECG classification system using information fusion and one-hot encoding techniques. *Mathematical Problems in Engineering*, 2018, 2018.

[318] Genshen Yan, Shen Liang, Yanchun Zhang, and Fan Liu. Fusing transformer model with temporal features for ECG heartbeat classification. In *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 898–905. IEEE, 2019.

[319] Bin Deng, Sen Jia, and Daming Shi. Deep metric learning-based feature embedding for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 58(2):1422–1435, 2019.

[320] Hoffer E and Ailon N. Deep metric learning using triplet network. *International Workshop on Similarity-Based Pattern Recognition*, pages 84–92, 2015.

[321] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[322] Gong Cheng, Ceyuan Yang, Xiwen Yao, Lei Guo, and Junwei Han. When deep learning meets metric learning: Remote sensing image scene classification via learning discriminative cnns. *IEEE transactions on geoscience and remote sensing*, 56(5):2811–2821, 2018.

[323] Alexander Hermans, Lucas Beyer, and Bastian Leibe. In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*, 2017.

[324] Yair Movshovitz-Attias, Alexander Toshev, Thomas K Leung, Sergey Ioffe, and Saurabh Singh. No fuss distance metric learning using proxies. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 360–368, 2017.

[325] Surat Tongyoo, Chairat Permpikul, Ratchada Haemin, and Nantawan Epichath. Predicting factors, incidence and prognosis of cardiac arrhythmia in medical, non-acute coronary syndrome, critically ill patients. *Journal of the Medical Association of Thailand= Chotmaihet thangphaet*, 96:S238–45, 2013.

[326] Andrew T Reisner, Gari D Clifford, and Roger G Mark. The physiological basis of the electrocardiogram. *Advanced methods and tools for ECG data analysis*, 1:25, 2006.

[327] Abdelrahman M Shaker, Manal Tantawi, Howida A Shedeed, and Mohamed F Tolba. Generalization of convolutional neural networks for ecg classification using generative adversarial networks. *IEEE Access*, 8:35592–35605, 2020.

[328] Peyman Bateni, Raghav Goyal, Vaden Masrani, Frank Wood, and Leonid Sigal. Improved few-shot visual classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14493–14502, 2020.

[329] Gregory Koch, Richard Zemel, Ruslan Salakhutdinov, et al. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2, page 0. Lille, 2015.

[330] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.

[331] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in neural information processing systems*, 29, 2016.

[332] Sion An, Soopil Kim, Philip Chikontwe, and Sang Hyun Park. Few-shot relation learning with attention for eeg-based motor imagery classification. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10933–10938. IEEE, 2020.

[333] Jyoti Narwariya, Pankaj Malhotra, Lovekesh Vig, Gautam Shroff, and TV Vishnu. Meta-learning for few-shot time series classification. In *Proceedings of the 7th ACM IKDD CoDS and 25th COMAD*, pages 28–36. 2020.

[334] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*, pages 1842–1850. PMLR, 2016.

[335] Tsendsuren Munkhdalai and Hong Yu. Meta networks. In *International Conference on Machine Learning*, pages 2554–2563. PMLR, 2017.

[336] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. 2016.

[337] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.

[338] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1199–1208, 2018.

[339] Tianshui Chen, Liang Lin, Xiaolu Hui, Riquan Chen, and Hefeng Wu. Knowledge-guided multi-label few-shot learning for general image recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[340] Cen Chen, Kenli Li, Wei Wei, Joey Tianyi Zhou, and Zeng Zeng. Hierarchical graph neural networks for few-shot learning. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(1):240–252, 2021.

[341] Sai Kumar Dwivedi, Vikram Gupta, Rahul Mitra, Shuaib Ahmed, and Arjun Jain. Protogan: Towards few shot learning for action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[342] Spyros Gidaris and Nikos Komodakis. Generating classification weights with gnn denoising autoencoders for few-shot learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 21–30, 2019.

[343] Zhu Zhan, Jinfeng Zhou, and Bugao Xu. Fabric defect classification using prototypical network of few-shot learning algorithm. *Computers in Industry*, 138:103628, 2022.

[344] Jiguang Dai, Yang Du, Tingting Zhu, Yang Wang, and Lin Gao. Multiscale residual convolution neural network and sector descriptor-based road detection method. *IEEE Access*, 7:173377–173392, 2019.

[345] Xiaomao Fan, Qihang Yao, Yunpeng Cai, Fen Miao, Fangmin Sun, and Ye Li. Multiscaled fusion of deep convolutional neural networks for screening atrial fibrillation from single lead short ecg recordings. *IEEE journal of biomedical and health informatics*, 22(6):1744–1753, 2018.

[346] Zhongke Gao, Xinlin Sun, Mingxu Liu, Weidong Dang, Chao Ma, and Guanrong Chen. Attention-based parallel multiscale convolutional neural network for visual evoked potentials eeg classification. *IEEE Journal of Biomedical and Health Informatics*, 25(8):2887–2894, 2021.

Appendix A

Supplementary figures

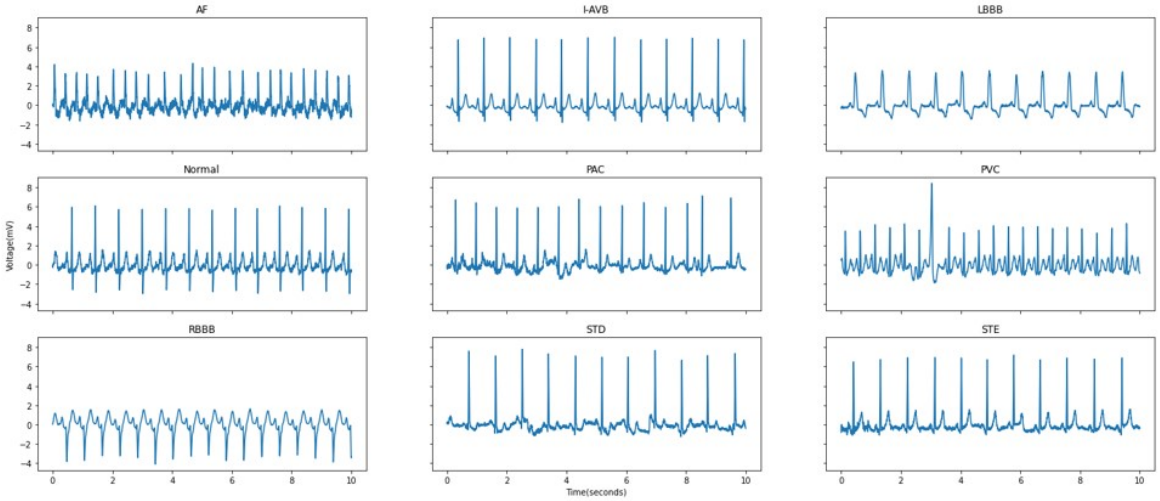

Figure A.1: Visualization of the ECG lead II waveform of 9 types of cardiac states in CPSC 2018

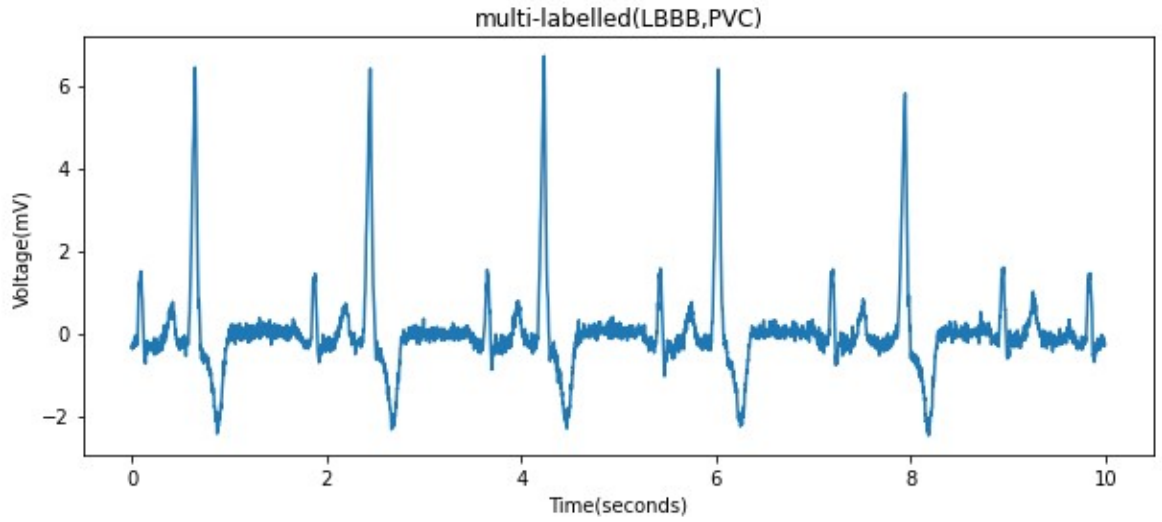

Figure A.2: Visualization of the ECG Lead II waveform of a multi-labelled ECG record (A2013)

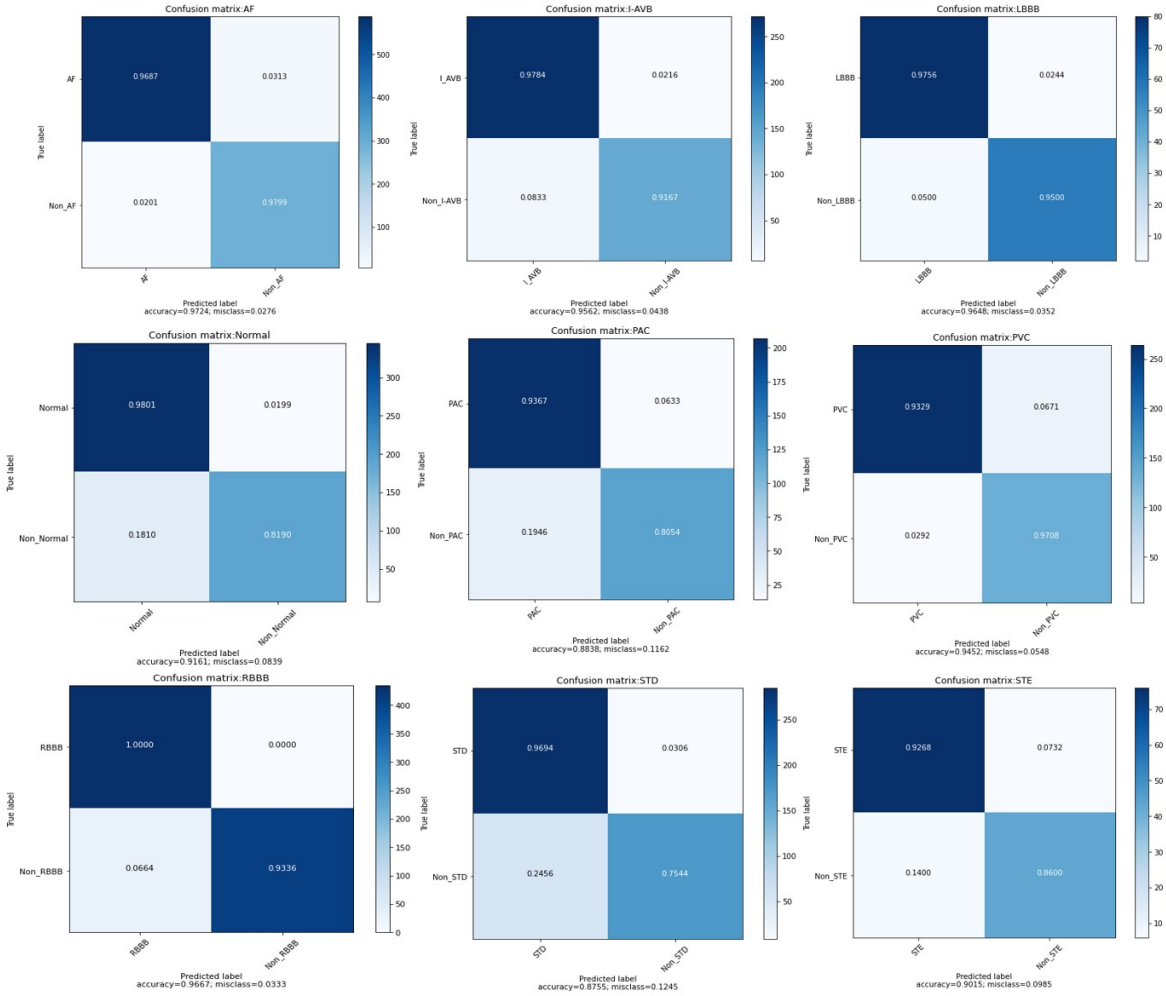

Figure A.3: Confusion matrix of 9 types of cardiac states by our proposed model.

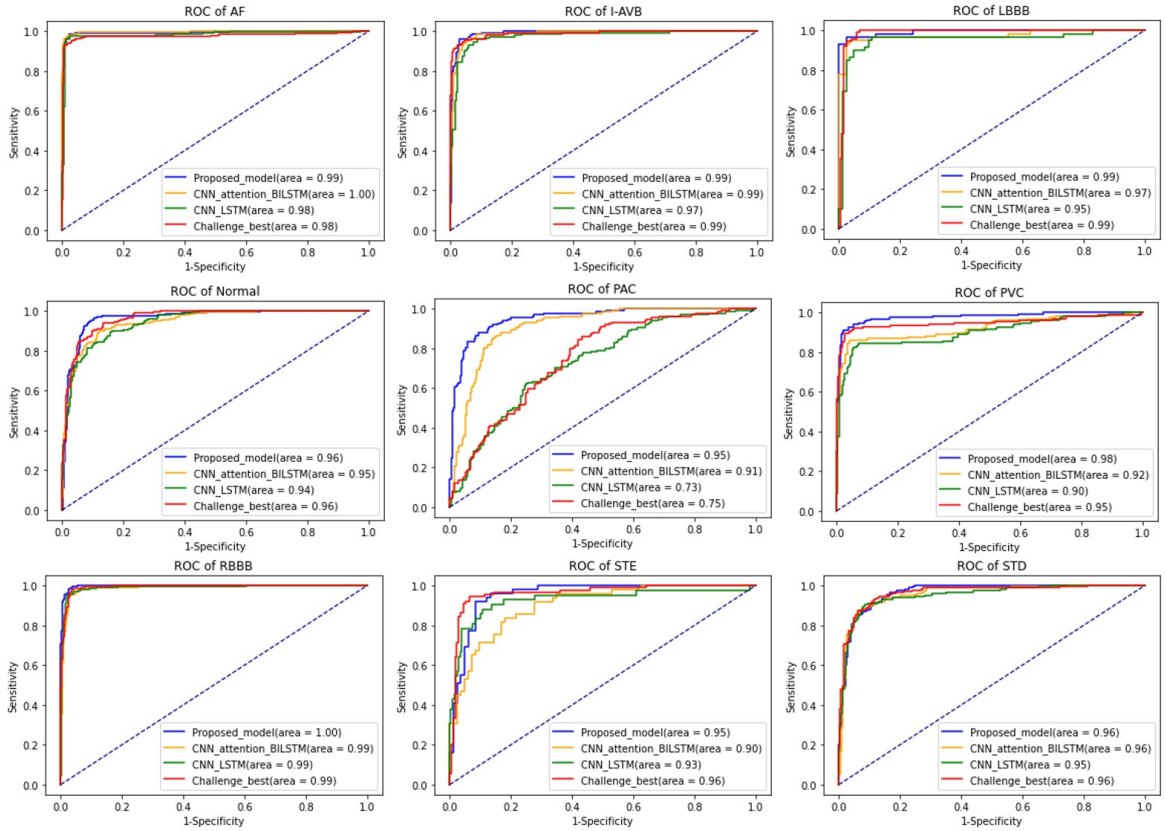

Figure A.4: Compared the receiver operator characteristic (ROC) curves for 9 types of abnormalities between 4 different models.

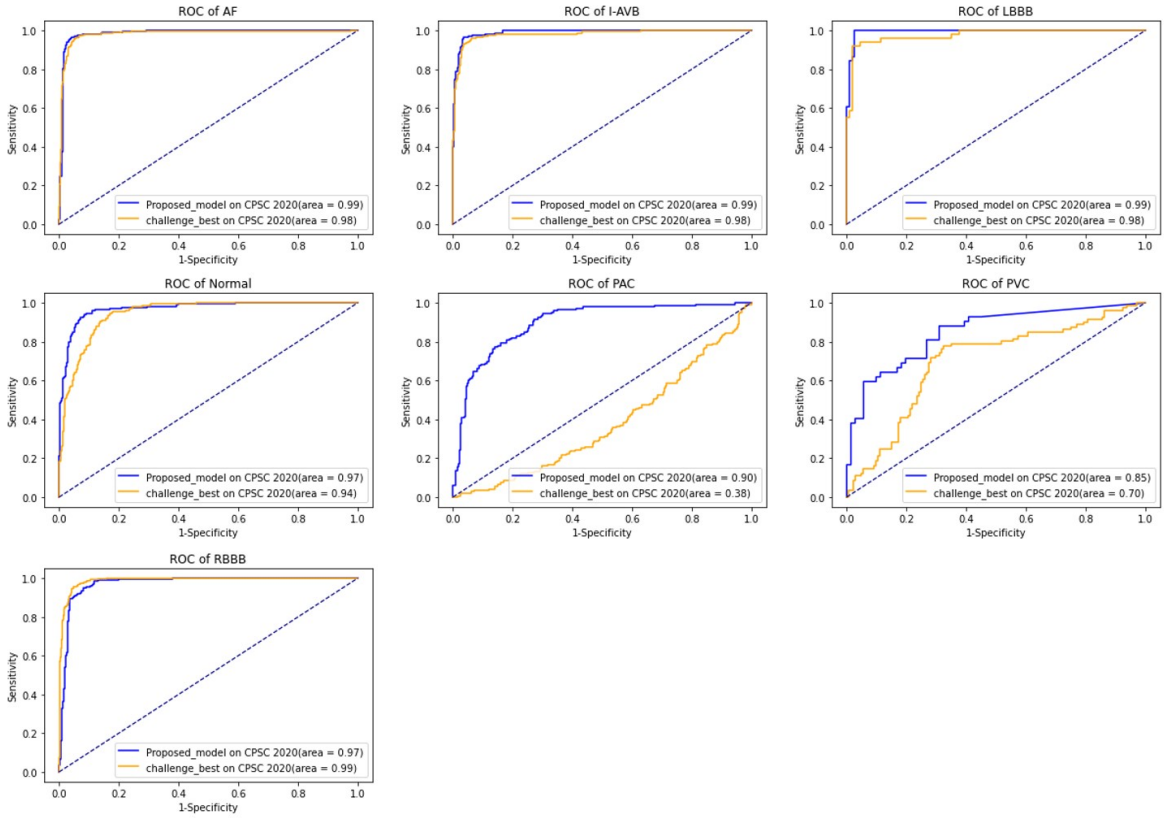

Figure A.5: Compared the Receiver operator characteristic (ROC) curves between the proposed model and challenge best model for 7 types of abnormalities in CPSC 2020.

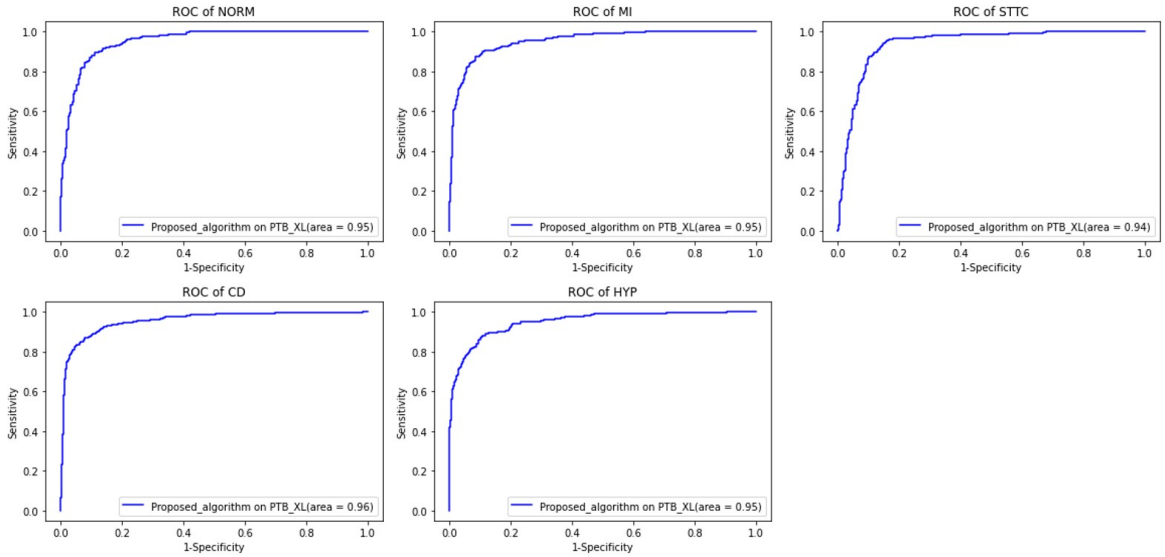

Figure A.6: Receiver operator characteristic (ROC) curves and AUC of 5 diagnosis labels in PTB XL dataset.

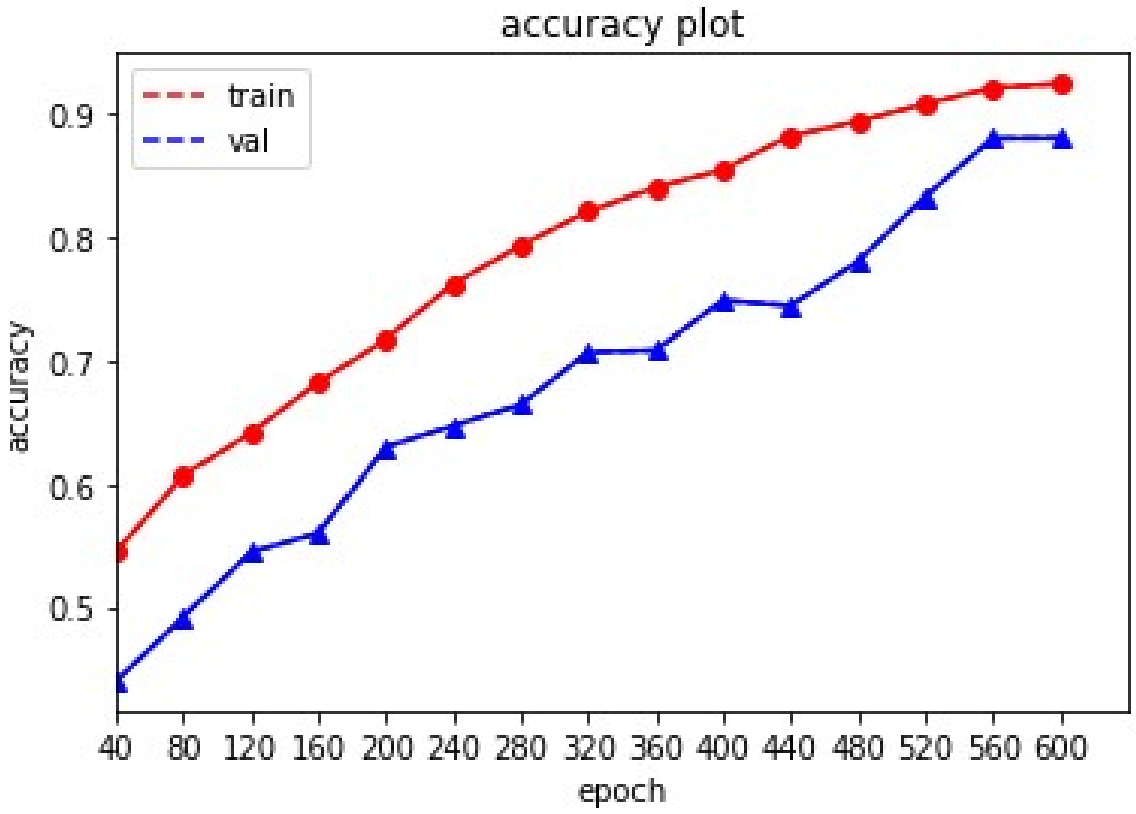

Figure A.7: Accuracy plot of the proposed model which evaluated by using 3-repeats 5-fold cross validation on training and validation set. Each point represents the training or validation accuracy of each fold.

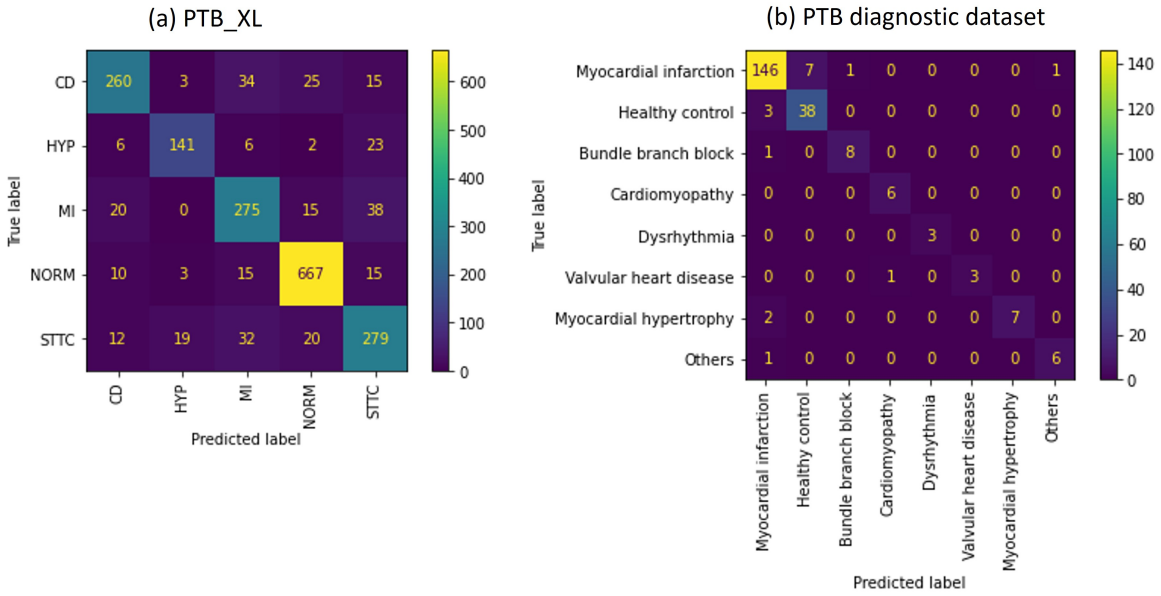

Figure A.8: Confusion matrix of the proposed model for cross validation. (a) confusion matrix of PTBXL dataset; (b) confusion matrix of PTB diagnostic dataset.

Appendix B

Supplementary Tables

Table B.1: Numbers and distribution of ECG recordings with multiple labels for six different types of abnormalities in CPSC2020.

	AF	**I-AVB**	**LBBB**	**RBBB**	**PAC**	**PVC**
AF	0	0	29	172	4	0
I-AVB		0	8	11	4	9
LBBB			0	0	10	0
RBBB				0	57	0
PAC					0	1
PVC						0

Table B.2: List of optimal parameters for each layer and residual block in the proposed model.

Layers/Blocks	**Kernel number**	**Kernel size**	**Pool size**
Conv1D	32	3	–
Pooling 1	–	–	3
DenseBlock 1	1^{st} : [32, 64] 2^{nd} : [64, 128]	1,1	–
DenseBlock 2	1^{st} : [32, 64, 64] 2^{nd} : [64, 128, 128]	3,3,7	–
Pooling 1	–	–	2
BiLSTM	128	–	–
Attention	256	–	–

Table B.3: List of optimal parameters for each layer and residual block in the proposed model.

No. of kernels Residual block 1	**No. of kernels Residual block 2**	**Kernel size Residual block 1**	**kernel size Residual block 2**	**Bilstm units**	Dropout	**Attention units**	Mean squared error
1^{st} : [8, 16] 2^{nd} : [16, 32]	1^{st} : [8, 16, 16] 2^{nd} : [16, 32, 32]	3,3	5,5,11	32	0.3	64	0.050
1^{st} : [1632] 2^{nd} : [32, 64]	1^{st} : [16, 32, 32] 2^{nd} : [32, 64, 64]	3,3	5,5,11	64	0.3	128	0.062
1^{st} : [32, 64] 2^{nd} : [64, 128]	1^{st} : [32, 64, 64] 2^{nd} : [64, 128, 128]	3,3	5,5,11	128	0.3	256	0.048
1^{st} : [8, 16] 2^{nd} : [16, 32]	1^{st} : [8, 16, 16] 2^{nd} : [16, 32, 32]	1,1	3,3,7	32	0.5	64	0.063
1^{st} : [16, 32] 2^{nd} : [32, 64]	1^{st} : [16, 32, 32] 2^{nd} : [32, 64, 64]	1,1	3,3,7	64	0.5	128	0.048
$\mathbf{1}^{st}$: [32, 64] $\mathbf{2}^{nd}$: [64, 128]	$\mathbf{1}^{st}$: [32, 64, 64] $\mathbf{2}^{nd}$: [64, 128, 128]	**1,1**	**3,3,7**	**128**	**0.5**	**256**	**0.044**

Table B.4: Structure and hyperparameters of plain CNN + attention based BiLSTM

layer	Kernel number/ size
Con1D+BN+ReLu	32 / 7
Dropout	0.5
Con1D+BN+ReLu	32 / 3
Con1D+BN+ReLu	32 / 3
Dropout	0.5
Con1D+BN+ReLu	32 / 3
Dropout	0.5
Con1D+BN+ReLu	64 / 3
Dropout	0.5
Con1D+BN+ReLu	64 / 3
Con1D+BN+ReLu	64 / 3
Dropout	0.5
Con1D+BN+ReLu	64 / 3
Dropout	0.5
Con1D+BN+ReLu	128 / 3
Con1D+BN+ReLu	128 / 3
Dropout	0.5
Flatten	-
Bidirectional LSTM	128
Dropout	0.5
Attention	256
Flatten	-
LSTM	256
sigmoid	1

Table B.5: Structure and modified hyperparameters based on the challenge-best model in CPSC 2018[245]

CNN blocks and layers	CNN layer	Kernel number/size
1	1	32/3
	2	32/3
	3 (pooling)	3
2	4	32/3
	5	32/3
	6 (pooling)	3
3	7	64/3
	8	64/3
	9 (pooling)	3
4	10	64/3
	11	64/3
	12 (pooling)	3
5	13	128/3
	14	128/3
	15 (pooling)	6
Bi-GRU	–	128
Attention	–	256
Batch Normalization	–	–
Dense(sigmoid)	–	–

Table B.6: Comparison of F1 score between different models based on test samples

Label	**Proposed Model**	**Plain CNN+Attention BiLSTM**	**Plain CNN+LSTM**	**Challenge-best**
	F1score	**F1score**	**F1score**	**F1score**
AF	0.959	0.961	0.957	**0.962**
I-AVB	**0.937**	0.878	0.883	0.846
LBBB	**0.958**	0.900	0.792	0.879
Normal	**0.885**	0.817	0.819	0.701
PAC	**0.848**	0.754	0.700	0.649
PVC	**0.920**	0.828	0.838	0.902
RBBB	**0.965**	0.954	0.940	0.904
STD	0.841	0.842	**0.852**	0.637
STE	**0.868**	0.683	0.667	0.601

Table B.7: The performance of models trained with CPSC2018 and CPSC2020 dataset.

	CPSC2018		CPSC2020	
label	Proposed F1score	Challenge-best F1score	Proposed F1score	Challenge-best F1score
AF	0.959	0.962	0.940	0.880
I-AVB	0.937	0.846	0.856	0.731
LBBB	0.958	0.879	0.898	**0.355**
Normal	0.885	0.701	0.870	**0.585**
PAC	0.848	0.649	0.743	**0.581**
PVC	0.920	0.902	0.798	**0.247**
RBBB	0.965	0.904	0.922	0.707

Table B.8: Layers and Parameters of TI-CNN model

Label	**Parameters**
Conv3-64	12×64×3
Conv3-64	64×64×3
Conv3-128	64×128×3
Conv3-128	128×128×3
Conv3-256	128×256×3
Conv3-256	256×256×3
Conv3-256	256×256×3
Conv3-512	256×256×3
Conv3-512	256×512×3
Conv3-512	512×512×3
Conv3-512	512×512×3
Conv3-512	512×512×3
Conv3-512	512×512×3
LSTM128	(512+128) ×128×4
LSTM32	(128+32) ×32×4
LSTM9	(32+9) ×9×4

Table B.9: Layers and Hyperparameters of CNN-GRU with Attention Mechanism

Label	**Kernel,stride**
Conv1-32	32*1*3, 1*2
Conv1-32	32*1*3, 1*2
Conv1-32	32*1*3, 1*2
Leaky ReLu	-
Dropout	0.5
Conv1-64	64*1*3, 1*2
Conv1-64	64*1*3, 1*2
Conv1-64	64*1*3, 1*2
Leaky ReLu	-
Dropout	0.5
Conv1-128	128*1*3, 1*2
Conv1-128	128*1*3, 1*2
Conv1-128	128*1*3, 1*2
Leaky ReLu	-
Dropout	0.5
Conv1-128	128*1*3, 1*2
Conv1-128	128*1*3, 1*2
Conv1-128	128*1*3, 1*2
Leaky ReLu	-
Dropout	0.5
Conv1-256	256*1*3, 1*2
Conv1-256	256*1*3, 1*2
Conv1-256	256*1*3, 1*2
Leaky ReLu	-
Dropout	0.5
BiGRU	-
BiGRU	-
Attention	-
Linear	24
sigmoid	

Table B.10: Layers and Hyperparameters of the network that is built based on the Challenge best model in CPSC 2018

Label	**Kernel,stride**
Conv3-12	12*3*3, 1
Conv3-12	12*3*3, 1
Maxpool	24*24, 2
Conv3-12	12*3*3, 1
Conv3-12	12*3*3, 1
Maxpool	24*42, 2
Conv3-12	12*3*3, 1
Conv3-12	12*3*3, 1
Maxpool	24*24, 2
Conv3-12	12*3*3, 1
Conv3-12	12*3*3, 1
Maxpool	24*24, 2
Conv3-12	12*3*3, 1
Conv3-12	12*3*3, 1
Max$_p$*ool*	48*48, 2
BiGRU	-
Attention	-
BN	-
Dense	9

Table B.11: Classification report of the proposed method on the test set.

Class	**Precision**	**Recall**	**F1 score**
AF	0.90	0.91	0.904
I-AVB	0.87	0.89	0.879
LBBB	0.98	0.94	0.959
Normal	0.84	0.90	0.868
PAC	0.78	0.81	0.794
PVC	0.94	0.92	0.929
RBBB	0.91	0.89	0.899
STD	0.88	0.86	0.869
STE	0.85	0.71	0.773

Table B.12: Classification report of the proposed method on the test set.

Class	**F1 scores**	
	without temporal RR features	with temporal features
AF	0.854	0.904
I-AVB	0.828	0.879
LBBB	0.943	0.959
Normal	0.859	0.868
PAC	0.625	0.794
PVC	0.849	0.929
RBBB	0.904	0.899
STD	0.860	0.869
STE	0.705	0.773