

PAROXYSMAL ATRIAL FIBRILLATION CLASSIFICATION FROM SINGLE-
LEAD ECG USING COMBINED CNN AND LSTM



AN INDEPENDENT STUDY SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT
FOR THE DEGREE OF MASTER OF SCIENCE PROGRAM IN
DATA SCIENCE AND ANALYTICS
KMUTL-DIGITAL ANALYTICS AND INTELLIGENCE CENTER SCHOOL OF SCIENCE
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
2022

KMITL-2022-SC-M-017-106

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.



COPYRIGHT 2022

SCHOOL OF SCIENCE

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

หัวข้อการค้นคว้าอิสระ	การจำแนกภาวะการเต้นหัวใจห้องบนสั้นพรีจากคลื่นไฟฟ้าหัวใจแบบลีดเดี่ยวด้วยการผสมผสานระหว่าง CNN และ LSTM
ชื่อนักศึกษา	นายณพพร ทรัพย์เสริมส่ง
รหัสประจำตัว	63605076
ปริญญา	วิทยาศาสตร์มหาบัณฑิต (วิทยาการข้อมูลและการวิเคราะห์) ศูนย์วิเคราะห์ข้อมูลดิจิทัลอัจฉริยะพระจอมเกล้าลาดกระบัง
พ.ศ.	2565
อาจารย์ที่ปรึกษาการค้นคว้าอิสระ	รองศาสตราจารย์ ดร.ละออ บุญเกษม

บทคัดย่อ

ภาวะหัวใจห้องบนสั้นพรีวเป็นประเภทหนึ่งของภาวะหัวใจเต้นผิดปกติ ซึ่งมีความสัมพันธ์กับโรคหลอดเลือดหัวใจหลายชนิด การตรวจหาภาวะหัวใจห้องบนสั้นพรีวนั้นยากต่อการตรวจพบเนื่องจากในหลายกรณีมักจะไม่มีการแสดงอาการออกมา ส่งผลให้ไม่ได้รับการวินิจฉัย โดยเฉพาะภาวะหัวใจห้องบนสั้นพรีวแบบเกิดขึ้นชั่วคราวเป็นรูปแบบหนึ่งของภาวะหัวใจห้องบนสั้นพรีวและมีโอกาสถูกตรวจพบต่ำกว่าปกติ ในการศึกษาวิจัยใช้ข้อมูลจาก Dynamic ECG Recordings: The 4th China Physiological Signal Challenge 2021 โดยการนำหน้าต่างการบันทึกสัญญาณคลื่นไฟฟ้าหัวใจ 30 วินาที ที่ถูกแบ่งจากช่วงเวลา 15 นาที เข้าสู่รูปแบบการเรียนรู้เชิงลึก (Deep Learning) ด้วยเทคนิคการกรองแบนด์พาส (Bandpass Filtering) เพื่อลดสัญญาณรบกวนและการใช้วิธีสังเคราะห์ข้อมูลเพิ่ม (SMOTE) เพื่อแก้ปัญหาความไม่สมดุลของข้อมูล รูปแบบของการเรียนรู้เชิงลึก (Deep Learning) ประกอบไปด้วยโครงข่ายประสาทเทียมแบบคอนโวลูชัน (Convolution Neural Network) และหน่วยความจำยาวระยะสั้น (Long-Short Term Memory) สำหรับการจำแนกภาวะหัวใจห้องบนสั้นพรีวชั่วคราว ได้ประสิทธิภาพจากคลื่นไฟฟ้าหัวใจลีดที่ 1 สูงถึง 94.15% และประสิทธิภาพจากคลื่นไฟฟ้าหัวใจลีดที่ 2 สูงถึง 95.18%

คำสำคัญ: การกรองแบนด์พาส การเรียนรู้เชิงลึก คลื่นไฟฟ้าหัวใจ โครงข่ายประสาทเทียมแบบคอนโวลูชัน ภาวะหัวใจห้องบนสั้นพรีวชั่วคราว หน่วยความจำยาวระยะสั้น

Independent Study Title	Paroxysmal Atrial Fibrillation Classification from Single-lead ECG using combined CNN and LSTM
Student Name	Mr. Nopporn Subsermsong
Student ID	63605076
Degree	Master of Science (Data Science and Analytics) KMITL-Digital Analytics and Intelligence Center
Year	2022
Independent Study Advisor	Assoc. Prof. Dr. Laor Boongasame

Abstract

Atrial Fibrillation (AF) is a type of cardiac arrhythmia which has a close connection to many cardiovascular diseases. Detection of atrial fibrillation is difficult to detect since many cases are usually undiagnosed. In particular, paroxysmal atrial fibrillation (PAF) is a form of AF that occurs occasionally and has a lower probability of being detected. In this study, we used the data from Dynamic ECG Recordings: The 4th China Physiological Signal Challenge 2021. The single lead of ECG signals from a 30 second recording window, over a 15-minute data segment, are fed into a deep learning (DL) model with a bandpass filtering technique for denoising ECG signals and SMOTE for solving the imbalanced datasets. The DL model consists of Convolution Neural Network (CNN) and Long-Short Term Memory (LSTM) for classification of PAF which give the accuracy of lead I of the ECG signal is 94.15% and the accuracy of lead II of the ECG signal is 95.18%.

Keywords: ECG signals, Bandpass Filtering, Denoising, Convolution Neural Networks, Deep Learning Model, Long Short-Term Memory, Paroxysmal Atrial Fibrillation

Acknowledgements

Foremost, I would like to express my sincere gratitude to my advisor, Assoc. Prof. Dr. Laor Boongasame, for her continuous support of my independent study (IS), for advice, motivation, enthusiasm, and immense knowledge. Her guidance helped me in all the research and writing of this independent study (IS).

Besides my advisor, I would like to thank the rest of my independent study committee: Asst. Prof. Dr. Pornpimol Chaiwuttisak, and Dr. Jiraphat Yokrattanasak for their encouragement, insightful comments, and hard questions.

Last, I would like to thank my family: my parents for their continuous support and motivation whenever I have problems.

Mr. Nopporn Subsermsong



Table of contents

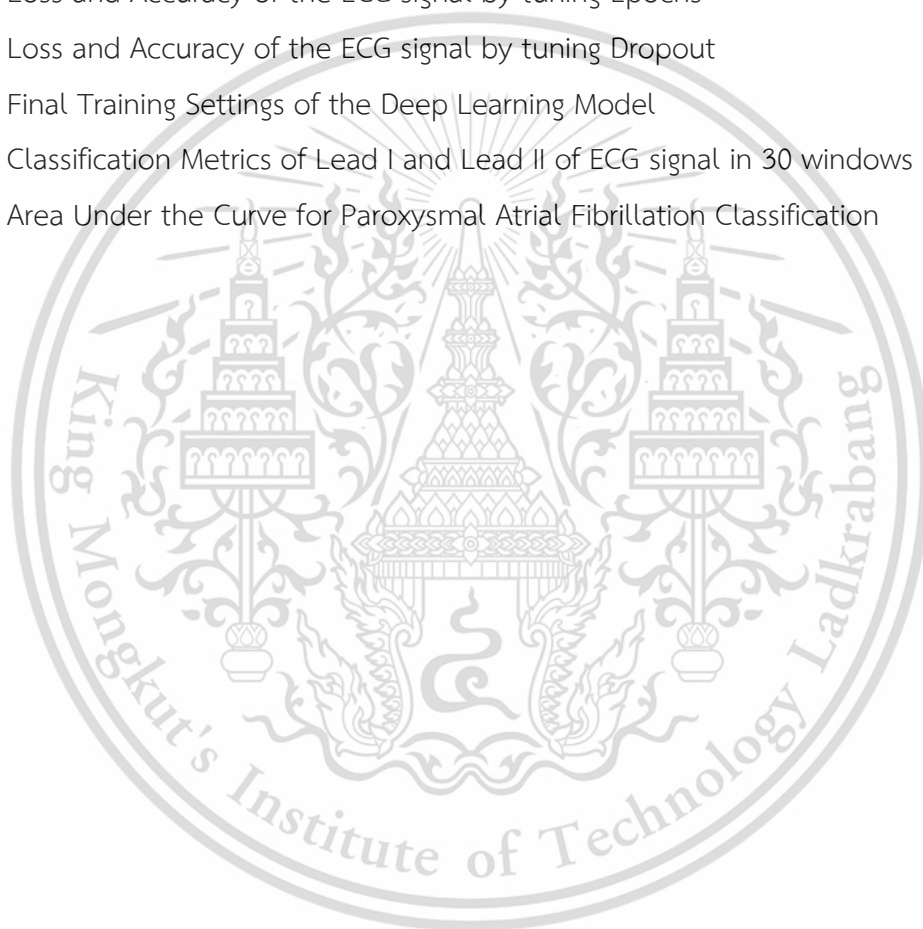
	Page
Abstract in Thai	i
Abstract in English	ii
Acknowledgements	iii
Table of contents	iv
List of tables	vii
List of figures	xi
Chapter 1 Introduction	1
1.1 Research motivation	1
1.2 Objectives of the study	2
1.3 Scope(s) of the study	2
1.4 Benefits of the study	2
Chapter 2 Theory and literature reviews	3
2.1 Background	3
2.1.1 Atrial Fibrillation	3
2.1.2 Electrocardiography	4
2.1.3 Convolution Neural Network	5
2.1.4 Recurrent Neural Network	14
2.2 Related Works	23
2.2.1 Convolution Neural Network	23
2.2.2 Recurrent Neural Network	24
2.2.3 Literature Reviews Summary	26
Chapter 3 Research methodology	
3.1 Introduction	30
3.2 Data Description	30
3.3 Data Pre-processing	32
3.3.1 Data Segmentation	32
3.3.2 Data De-noising	33
3.3.3 Data Normalization	34

Table of contents (cont.)

	Page
3.3.4 Over-Sampling	35
3.3.5 Data Modeling	37
Chapter 4 Main results and discussion	39
4.1 Experimental Setup	39
4.1.1 Tuning Model	39
4.1.2. Measure the performance of deep learning model (Without SMOTE)	42
4.1.3 Measure the performance of deep learning model (With SMOTE)	43
4.2 Summary and Discussion	47
4.2.1 Learning Curves	47
4.2.2 Classification Metrics	50
Chapter 5 Conclusions and suggestions	51
5.1 Conclusions	51
5.2 Suggestions	51
References	52
Author biography	55

List of tables

Table	Page
1.1 Architecture of the Deep Learning Model	38
1.2 Initial Training Settings of the Deep Learning Model	39
1.3 Loss and Accuracy of the ECG signal by tuning Learning Rate	40
1.4 Loss and Accuracy of the ECG signal by tuning Batch Size	41
1.5 Loss and Accuracy of the ECG signal by tuning Epochs	42
1.6 Loss and Accuracy of the ECG signal by tuning Dropout	42
1.7 Final Training Settings of the Deep Learning Model	43
1.8 Classification Metrics of Lead I and Lead II of ECG signal in 30 windows	44
1.9 Area Under the Curve for Paroxysmal Atrial Fibrillation Classification	46



List of figures

Figure	Page
2.1 Heart's Anatomy and Physiology	4
2.2 The typical of ECG signal	5
2.3 Convolutional Neural Networks Architectures	6
2.4 Convolution Operation Step in Convolution Layer	7
2.5 Max - Pooling in pooling layer	8
2.6 Average - Pooling in pooling layer	8
2.7 Fully-Connected Layer Diagram	10
2.8 Sigmoid Function Graph	11
2.9 Tanh Function Graph	11
2.10 ReLU Function Graph	12
2.11 LeakyReLU Function Graph	12
2.12 Optimization of Cost Function with Gradient Descent	13
2.13 An unrolled recurrent neural network	14
2.14 Architecture comparison of feedforward neural network and recurrent neural network	15
2.15 Types of Standard Recurrent Neural Network Diagram	15
2.16 (Left) Vanishing Gradients, (Right) Exploding gradients	16
2.17 (Left) Recurrent Neural Network, (Middle) Long-Short Term Memory Units, (Right) Gate Recurrent Units	17
2.18 Long-Short Term Memory (LSTM) Blocks	18
2.19 Forget gate diagram in Long-short Term Memory	19
2.20 Forget gate diagram in Long-short Term Memory	19
2.21 Cell state diagram in Loong-Short Term Memory	20
2.22 Output gate diagram in Long-Short Term Memory	21
2.23 Gated Recurrent Units (GRU) Block	21
2.24 Update gate of Gated Recurrent Units (GRU)	22
2.25 Reset gate of Gated Recurrent Units (GRU)	23
3.1 Flowchart of this Study	29

List of figures (cont.)

Figure	Page
3.2 Example lead-I ECG of Non-Atrial Fibrillation	30
3.3 Example lead-I ECG of Persistent Atrial Fibrillation	30
3.4 Example lead-I ECG of Paroxysmal Atrial Fibrillation (PAF)	30
3.5 Schematic diagram of data segmentation	33
3.6 (a) The raw lead I ECG signal (b) The filtered lead I ECG signal	34
3.7 Example of Comparison between Raw lead I ECG and Normalized lead I ECG (both are filtered with bandpass filtering)	35
3.8 Imbalanced dataset of ECG data	36
3.9 Architecture of CNN and LSTM	37
4.1 Schematic of Measuring Performance of Model	44
4.2 (a) Learning Curve of Loss of Lead I (b) Learning Curve of Loss of Lead II	48
4.3 (a) Learning Curve of Accuracy of Lead I (b) Learning Curve of Accuracy of Lead II	49

Chapter 1

Introduction

1.1 Research motivation

Cardiovascular diseases (CVDs) are a group of disorders of the heart and blood vessels that are usually associated with a build-up of fatty deposits inside the arteries and an increased risk of blood clots that can damage arteries in organs such as the brain, heart, etc. [1]. Atrial fibrillation (AFib or AF) is one of the most common types of arrhythmias, which are irregular heart rhythms that can lead to blood clots, strokes, heart failure, and other heart-related complications. According to ICD-10-CM diagnosis code I48 [2], there are two main types of atrial fibrillation, consisting of paroxysmal atrial fibrillation (PAF) and persistent atrial fibrillation (AF). The PAF is an episode of uncoordinated movement of the atria that occurs occasionally and takes a few minutes to a few days to stop. But with a physical exam or an electrocardiogram (ECG or EKG), a doctor will be able to figure out what's wrong.

An ECG is a simple test that involves having sensors attached to the skin on the arms, legs, and chest. The sensors detect electrical signals each time the heart beats. Then, doctors will interpret the signals from components consisting of P-wave, QRS complexes, T-wave, and U-wave. Both AF [3] and PAF patients have tiny irregular fluctuations in the P-wave and unequal RR intervals which can be observed in ECG signals. But the PAF does not have any obvious symptoms for the patient and may not be detected during clinical monitoring, so patients should wear an ECG wearable device to monitor irregular heartbeats that might occur sometime during the day. Although using an ECG wearable device given by doctors helps to specify PAF easily, wearing it is not practical in daily life.

Nowadays, technology is growing very fast and making daily life easier, such as smartphones, smartwatches, and so on. Moreover, some devices add healthcare functions to get people healthy. For example, the Apple Watch has an ECG sensor to pick up on strange signals from users. However, the limitation of the Apple Watch is that it can only record an ECG signal but not classify or specify disease. Hence, if we can improve this function by combining algorithms for classification, it will decrease a doctor's workload and mistakes from manual interpretation. In this paper, the

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

researcher uses data from Paroxysmal Atrial Fibrillation Events Detection from Dynamic ECG Recordings: The 4th China Physiological Signal Challenge 2021 to make algorithms for classifying paroxysmal atrial fibrillation.

1.2 Objectives of the study

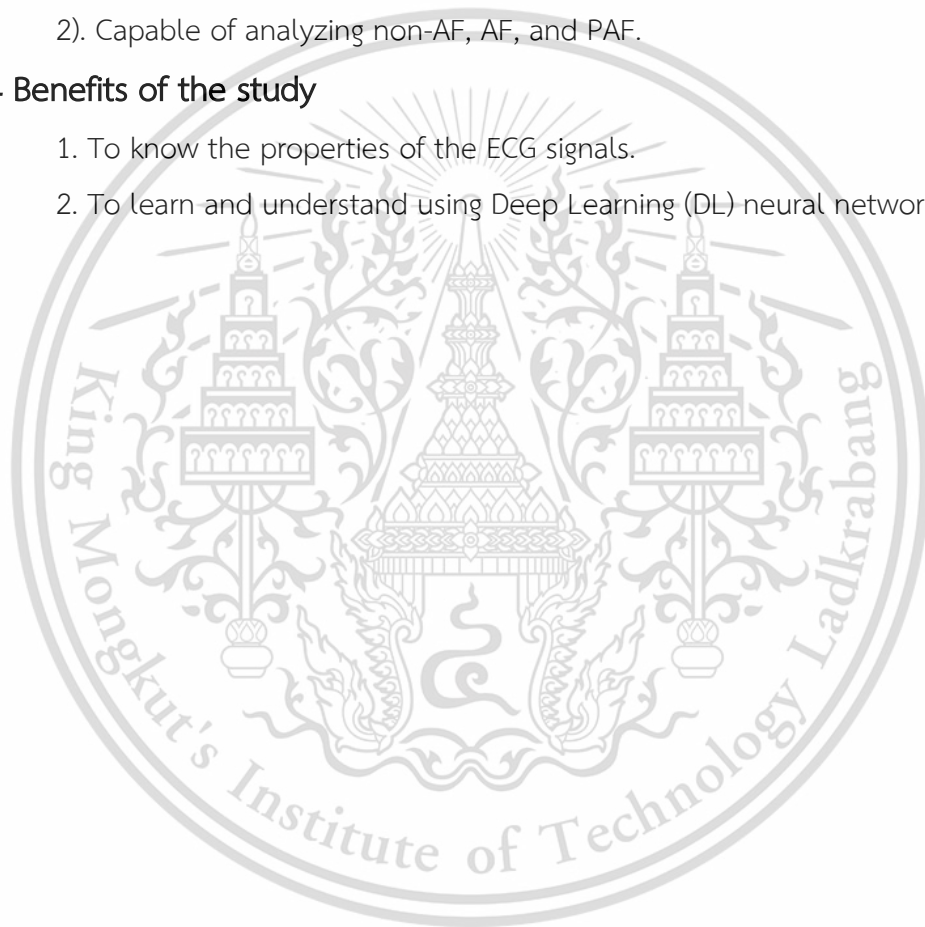
To build the model for detecting PAF from ECG signals.

1.3 Scope(s) of the study

- 1). Capable of classifying or predicting Paroxysmal Atrial Fibrillation (PAF) in everyday life.
- 2). Capable of analyzing non-AF, AF, and PAF.

1.4 Benefits of the study

1. To know the properties of the ECG signals.
2. To learn and understand using Deep Learning (DL) neural networks.



Chapter 2

Background and literature Reviews

2.1 Background

2.1.1 Atrial Fibrillation

Atrial Fibrillation (AFib or AF) is the most common cardiac rhythm disorder, which results in mortality and disability from stroke, heart failure, thromboembolism complications, and quality of life. In 2017, the number of deaths due to atrial fibrillation and flutter increased by 47.8% from 2007 [4]. Hence, AFib has become a critical issue which should be treated promptly to prevent risk factors for stroke.

AFib occurs when electrical signals in the right atrium impulses rapidly fire at once, causing rapid and intermittent rhythms. As a result, the atria are unable to contract and squeeze blood effectively, resulting in embolism, which can lead to stroke [5] and other heart-related problems.

There are many types of AF, but the American Heart Association (AHA) and American College of Cardiology (ACC) further classified AF into four types, including Paroxysmal AF, Persistent AF, Long-Lasting AF, and Permanent AF. Meanwhile, this study is only interested in three types, excluding long-lasting.

The simplified definitions of AF can be described by the duration of episodes. The first type of AF that terminates spontaneously or with intervention within 7 days of onset is called "Paroxysmal AF". The second type of AF is a continuous AF that is sustained for more than 7 days, which is called "Persistent AF". The last type of AF is called "Permanent AF" [5] when the heart rhythm cannot be restored as normal. So, the patient and the doctor should decide together to change a treatment for a symptom rather than try to fix the sinus rhythm.

2.1.2 Electrocardiography

Over the past two centuries, the electrocardiogram (ECG) has been a useful technology for clinical diagnosis.

ECG is a graph versus time of electrical activity of the heart using electrodes placed on the skin to detect the electrical signal on the sinoatrial node (SA) which causes contraction on the atria and blood pumping to the ventricles as shown in Figure 2.1. Therefore, the PR interval is shown on graph as a wave is called P-wave. Then, the signal passes from atria to the ventricles through the atrioventricular node (AV) causing the ventricle is filled with blood. The signal is detected as a flat line on graph called PR segment. After the signal leaves the AV node and travels along with a Bundle of His, the signal travels across the heart's ventricles causing it to contract, pumping blood to the lungs and the body. Next to the previous signal is called QRS complex. Finally, the ventricles recover to a normal state. The signal is detected as shown on graph is called T wave which is connected to a QRS complex. The flat line between QRS and T wave is called QT interval and ST interval as shown in Figure 2.2.

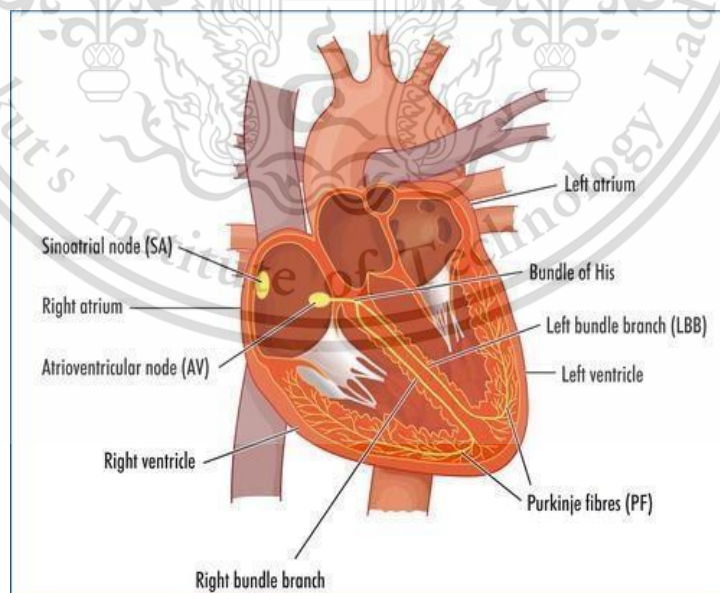


Figure 2. 1 Heart's Anatomy and Physiology

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

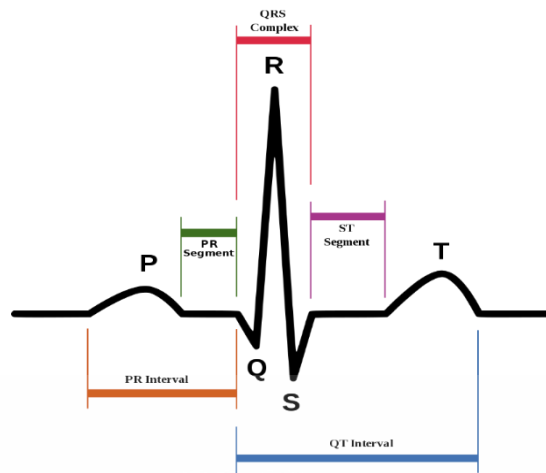


Figure 2. 2 The typical of ECG signal

In general, ECG signals are measured by surface electrodes which are placed on the patient's skin. Meanwhile, it also causes various types of noise to occur, including baseline drift, motion artifacts, electrode contact, muscle contraction, power-line interference, etc. Moreover, the noise in the ECG signal may mislead doctors to misdiagnose the disease. Hence, ECG noise removal is necessary to decrease misdiagnosis of diseases.

2.1.3. Convolutional Neural Networks

Convolutional neural networks [7] (CNN or ConvNet) are types of Artificial Neural Networks (ANN) that have a deep feed-forward architecture and have an amazing ability to generalize compared to other networks with FC layers. They can learn abstract features of objects, especially spatial data, and identify objects quickly.

Generally, the CNN architecture is formed from the stacking of three types of layers, which are the convolutional layer, the pooling layer, and the fully-connected layer. As shown in Figure 2.9, in part of the convolution layer and pooling layer, these two layers are known as the "feature extraction" layers because they are mostly used to extract features from data. Another part is the fully-connected layer, which is used for classification.

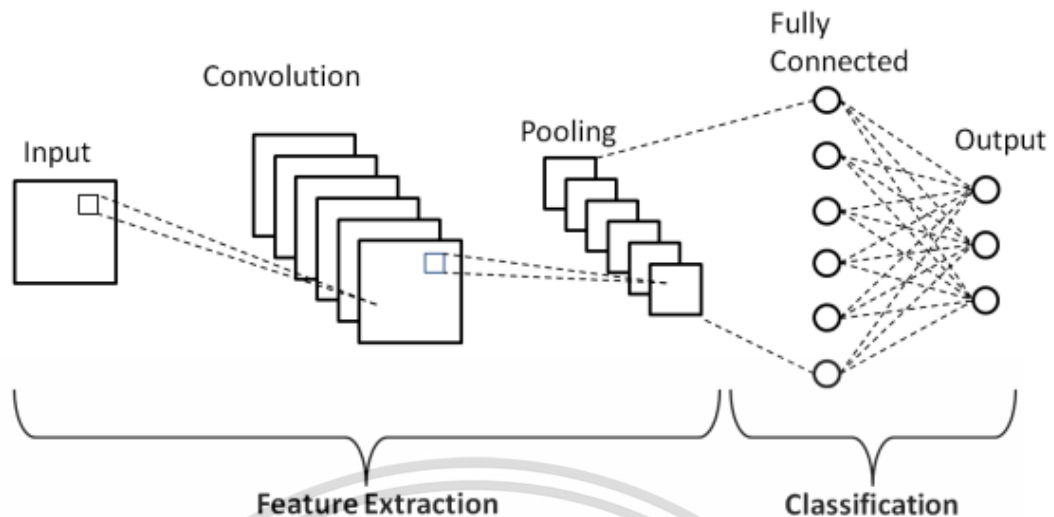


Figure 2. 3 Convolutional Neural Networks Architectures

I. Convolution layer

The convolution layer is the most important component of any CNN architecture. It contains a set of convolutional kernels (also called filters). These kernels can be described as a grid of discrete values or numbers, where each value is known as the weight of this kernel. At the beginning of the training process of a CNN model, all the weights are assigned with random numbers and will be tuned at each training epoch. But when the data reaches a convolutional layer, the layer combines each filter across the input's spatial dimensions to make an output feature map.

In the convolution layer, the convolution operation is used to calculate the difference between input data and filters. For example, a grayscale image of 4x4 dimension is convolved by a 2 x 2 kernel with randomly initialized weights. Then, we slide each filter across the width and height of the input image and compute dot products between the entries of the filter and the input at any position to generate an output feature map. As shown in Figure 2.10, we can understand the convolution operation more clearly in each computation step.

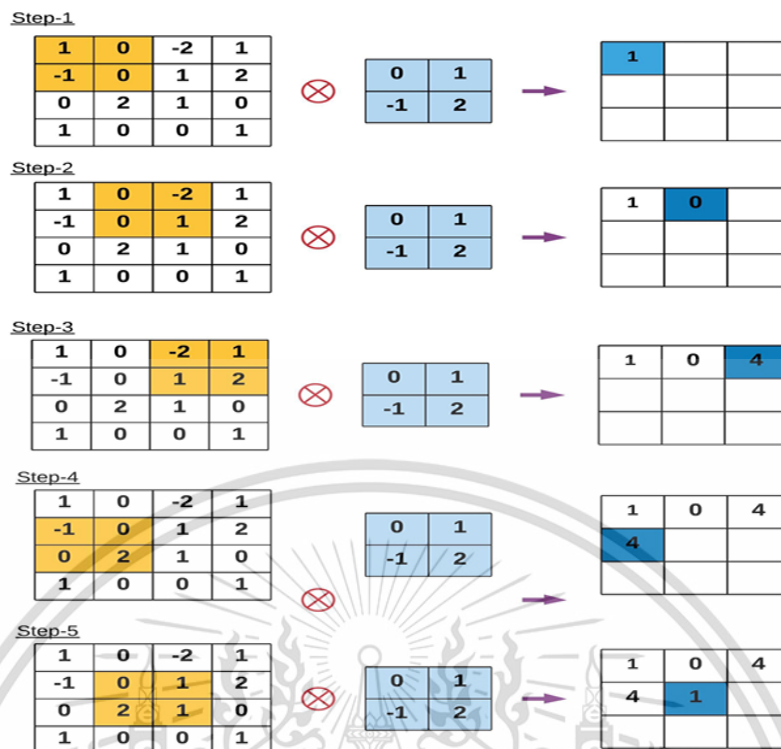


Figure 2. 4 Convolution Operation Step in Convolution Layer

In the above example, the convolution operation is applied to the input image with no padding and with a stride of 1 to the kernel. But we can use other stride values in the convolution operation. However, optimization of these parameters also significantly reduces the complexity of the model. Besides both padding and stride, there is another parameter, which is the number of filters. Hence, there are three hyperparameters consisting of the number of filters, the stride, and setting zero-padding.

A. The number of filters affects the depth of the output that is produced by the convolutional layers.

B. Stride is the distance or number of pixels that the kernel moves over the input matrix, which can be calculated in the equation as shown below.

$$\lfloor \frac{n + 2p - f}{s} + 1 \rfloor, \lfloor \frac{n + 2p - f}{s} + 1 \rfloor$$

where n denotes input dimension, f denotes filter dimension, and p denotes pixel dimension on each side of padding

C. Zero-padding is usually used when the filters do not fit the input data. However, all elements are set to zero at the outside of the input matrix that produces a larger or equally sized output. There are two types of padding, which include:

a. Valid padding: This is also known as no padding. In this case, the last convolution is dropped if the dimensions do not align. Dimensions of output are shown below.

$$(n - f + 1), (n - f + 1)$$

where n denotes the dimension of input and f the dimension of the filter,

b. Same padding: This padding ensures that the output layer has the same size as the input layer. Dimensions of output are shown below.

$$(n + 2p - f + 1), (n + 2p - f + 1)$$

$$p = \frac{f - 1}{2}$$

where p denotes the dimension of a pixel on each side of padding.

II. Pooling Layer

Pooling layers, also known as downsampling, conducts the number of parameters in the input. The pooling operation sweeps a filter across the entire output that is similar to the convolution layer. However, the difference between these two layers is that this filter does not have any weights. Instead, the kernel

applies an aggregation function to the values within the receptive field by populating the output array. There are two main types of pooling:

A. Max pooling

In max pooling, the filter simply selects the maximum pixel value in the receptive field. For example, given 4 pixels with the values 2, 3, 5, and -1 as shown in Figure 2.11, the max pooling layer will send an output of 5.

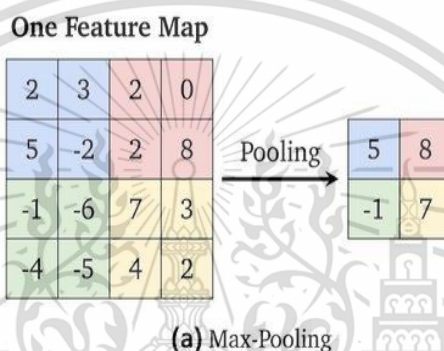


Figure 2. 5 Max - Pooling in pooling layer

B. Average pooling

In average pooling, it works by calculating the average value of the pixel values in the receptive field. For example, given 4 pixels with the values 2, 3, 5, and -1 as shown in Figure 2.12, the average pooling layer will produce an output of 2.

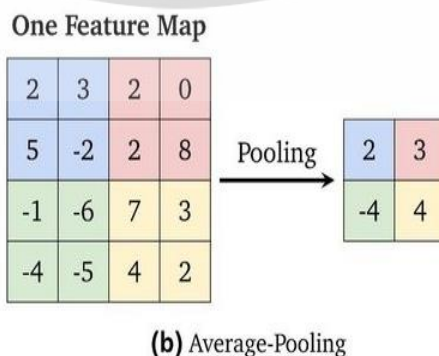


Figure 2. 6 Average- Pooling in pooling layer

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

Although there is a lot of information lost in the pooling layer, the pooling operations help to reduce complexity, improve efficiency, and limit the risk of overfitting.

III. Fully-Connected Layer

This layer is the last part of every CNN architecture, consisting of fully-connected layers. The FC layers take input from the final convolutional or pooling layer, which is in the form of a set of metrics (feature maps), and those metrics are flattened to create a vector, and this vector is then fed into the last layer of FC layers to generate the final output of CNN, as shown in Figure 2.13.

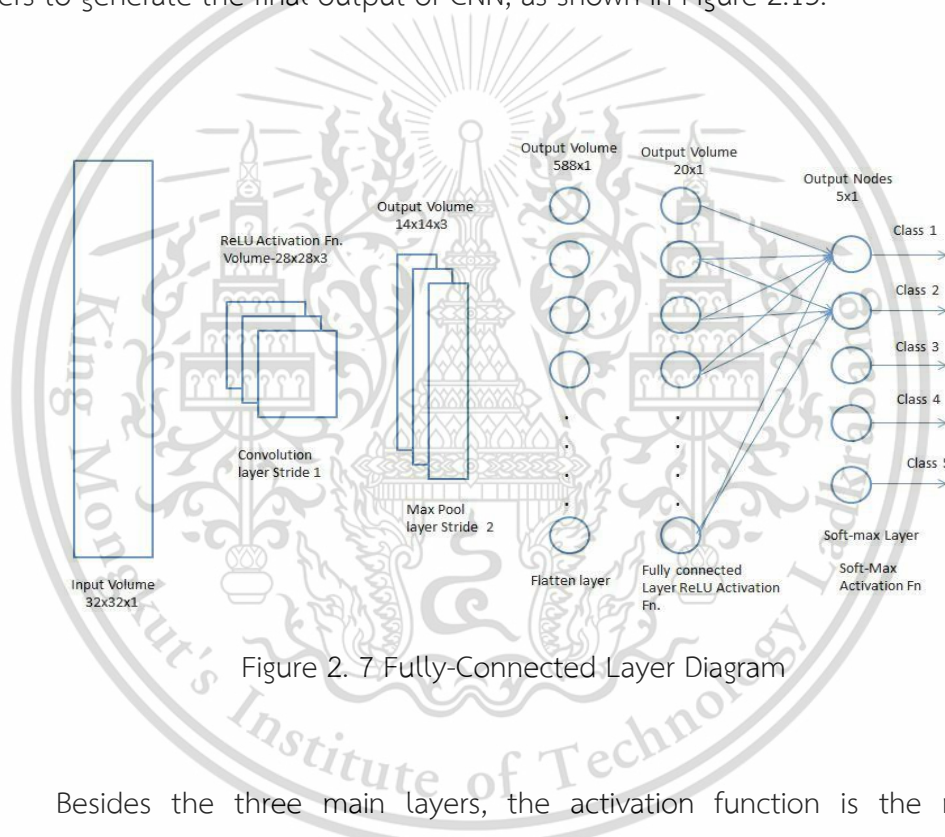


Figure 2. 7 Fully-Connected Layer Diagram

Besides the three main layers, the activation function is the necessary component that is used to map the input to the output. The most used activation functions are in deep neural networks described below.

A. Sigmoid (or Logistic)

The sigmoid activation function takes any real values as input and outputs values in the range of $[0,1]$. The curve of the sigmoid function is of 'S' shaped. The mathematical representation of a sigmoid is:

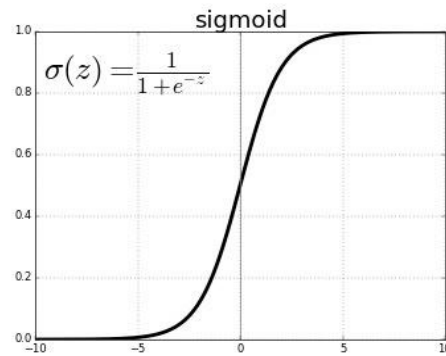


Figure 2. 8 Sigmoid Function Graph

$$f(x)_{\text{sigm}} = \frac{1}{1 + e^{-x}}$$

B. Tanh

The Tanh activation function is used to bind the input values (real numbers) within the range of $[-1, 1]$. The mathematical representation of Tanh is:

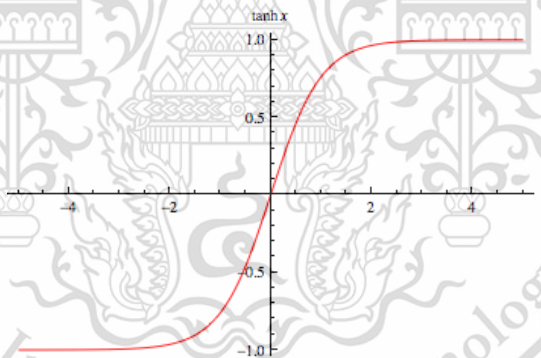


Figure 2. 9 Tanh Function Graph

$$f(x)_{\text{tanh}} = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

C. ReLU

The Rectifier Linear Unit (ReLU) is the most commonly used activation function in convolution neural networks. It is used to convert all the input values to positive numbers. The mathematical representation of ReLU is:

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

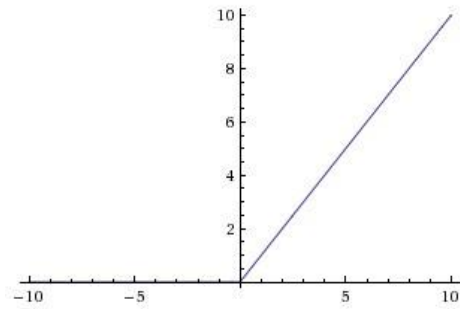


Figure 2. 10 ReLU Function Graph

$$f(x)_{ReLU} = \max(0, x)$$

D. Leaky ReLU

Leaky ReLU activation function does not ignore the negative inputs completely, rather it down-scales those negative inputs. Leaky ReLU is used to solve the Dying ReLU problem. The mathematical representation of Leaky ReLU is:

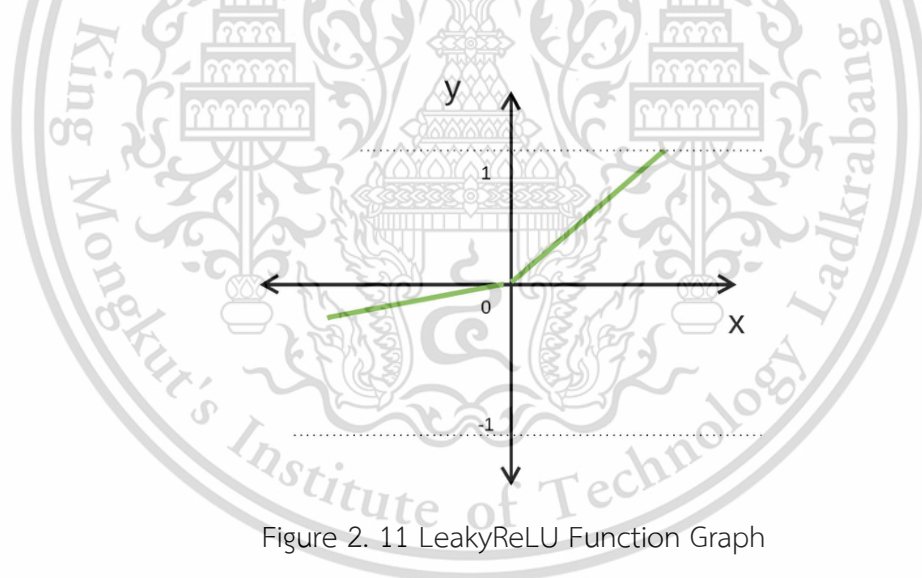


Figure 2. 11 LeakyReLU Function Graph

$$f(x)_{LeakyReLU} = \begin{cases} x, & \text{if } x > 0 \\ mx, & \text{if } x \leq 0 \end{cases}$$

In this output layer, the prediction error or classification error is calculated by using some loss function to tell the network how far off their prediction is from the actual output. Then, this error will be optimized during the learning process of the model. The loss function uses two parameters to calculate the error. The first parameter is the estimated output, and the second parameter is the actual output. However, there are several types of loss functions used in different situations or problems, such as the Cross-Entropy or Soft-Max Loss Function, the Euclidean Loss Function, and so on.

After the end of forward propagation, that is, evaluation of prediction or classification, we can use back propagation for updating and finding the optimal values of weights by using optimizers, which helps the model to minimize the error. Optimizers are the methods or mathematical formulations that change the attributes of neural networks, such as Gradient Descent.

For example, using gradient descent helps in calculating the new weights. To understand how Gradient Descent optimizes the cost function as shown in the Figure 2.8 and step of back propagation as shown below.

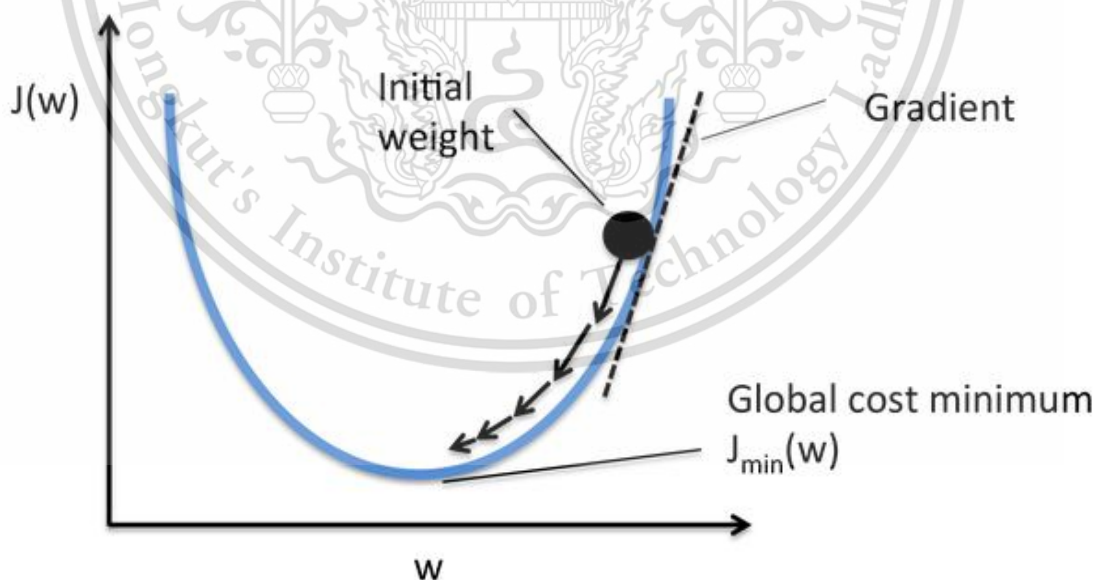


Figure 2. 12 Optimization of Cost Function with Gradient Descent

Step 1: The weights are initialized randomly, and intercepts are assigned to the model while forward propagation and the errors are calculated after all computation.

Step 2: The gradient is calculated

Step 3: The new weights are calculated using the below formula

$$W_x^* = W_x - a \left(\frac{\partial Error}{\partial W_x} \right)$$

where W_x^* is new weight, W_x is old weight, a is learning rate, and $\frac{\partial Error}{\partial W_x}$ is

derivative of error with respect to weight.

Step 4: This process will calculate the new weight, then calculate the errors from the new weights, then update the weights, and then continue till the model reaches global minima and loss is minimized.

2.1.4 Recurrent Neural Network

A recurrent neural network (RNN) [8] is a special type of artificial neural network adapted to work with time series data or data that involves sequences. RNN is a kind of deep learning that is commonly used for ordinary and temporal problems, such as language translation, natural language processing (NLP), speech recognition, and image captioning. Therefore, there are many applications that incorporate these algorithms, such as Siri and Google Translate. The architecture of RNNs is shown in Figure 2.14.

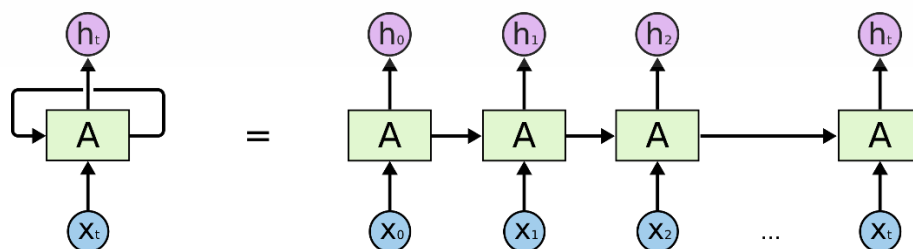


Figure 2. 13 An unrolled recurrent neural network

In a feed-forward neural network, information only moves in one direction from the input layer, through the hidden layers, to the output layer. The information moves straight through the network without remembering where it came from. This makes it hard to predict what will happen in the future. A feed-forward network only considers the current input. Therefore, it simply cannot remember anything that occurred in the past except its training. Conversely, the RNN, which has an architecture like a feed-forward neural network as shown in Figure 2.15, has a memory to take information from prior inputs to influence the current input and output.

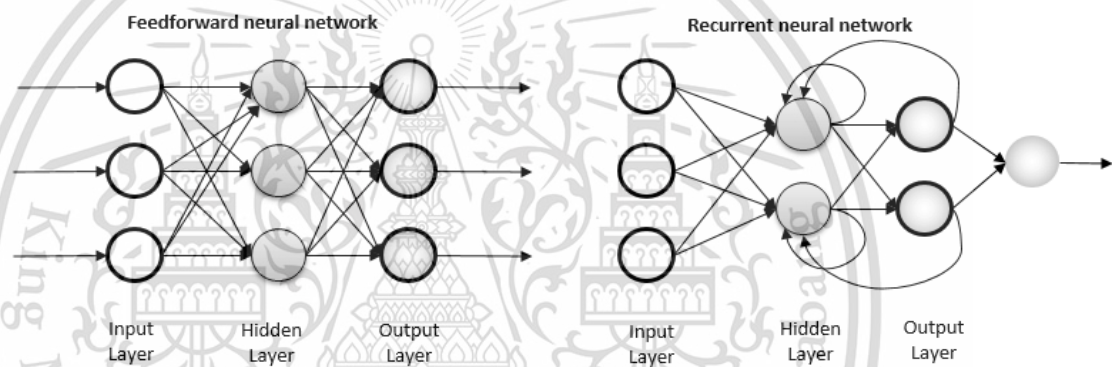


Figure 2. 14 Architecture comparison of feedforward neural network and recurrent neural network

Since the RNNs have four main types of RNNs consisting of one-to-one, one-to-many, many-to-one, and many-to-many as shown in Figure 2.16, they can be used for different use cases such as music generation, sentiment classification, and machine translation.

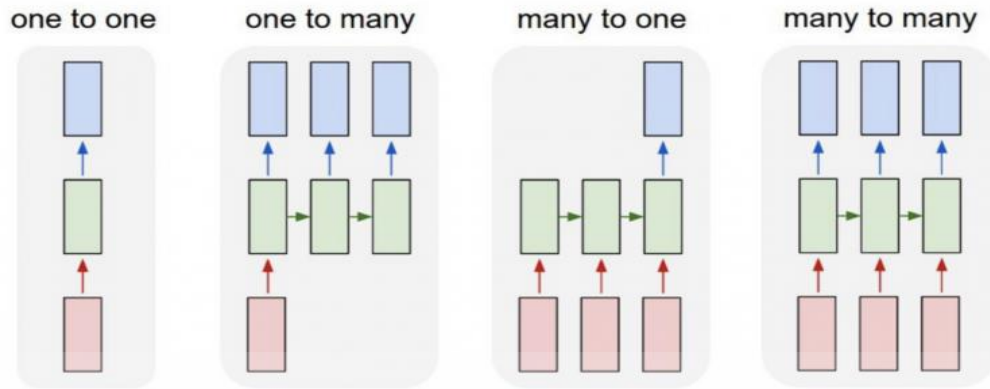


Figure 2. 15 Types of Standard Recurrent Neural Network Diagram

I. Problem of Recurrent Neural Networks

There are two major obstacles that RNNs have had to deal with that are about gradient descent. A gradient is a partial derivative with respect to its inputs. That means a gradient measure how much the output of a function changes if the inputs are changed a little bit.

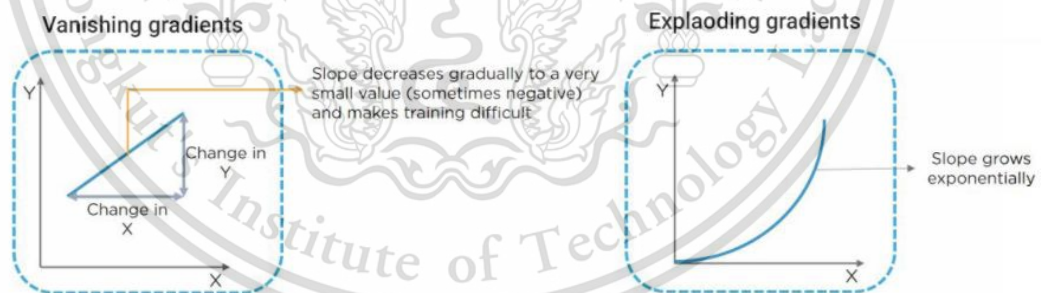


Figure 2. 16 (Left) Vanishing Gradients, (Right) Exploding gradients

A. Exploding Gradients Problem

Exploding gradients is a problem in which the gradient values become large during an update in recurrent neural networks. The gradient value will overflow and result in NaN values if taken to an extreme.

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

As a result, exploding gradients in recurrent neural networks can lead to an unstable network that cannot learn from the training data at best and cannot learn over long input sequences of data.

B. Vanishing Gradients Problem

Vanishing gradients is a problem in which the gradient values frequently become smaller and smaller until they approach zero, leaving the weights of the beginning or lower layers essentially unchanged. As a result, gradient descent never converges to the best solution.

According to the RNN problem, both exploding and vanish gradients will occur often in recurrent neural networks because the RNNs cannot deal with long-term dependencies. For example, predicting the last word of "The fat boy didn't play basketball because he was sick." In this sentence, the word "he" is mentioned to the flat boy, but the networks don't remember the previous words. Hence, the gap between two words is too large, which causes the RNNs to not learn to connect the information.

Hence, LSTMs and GRUs were created to solve the problem of long-term dependencies. Both have internal mechanisms called gates that can regulate the flow of information.

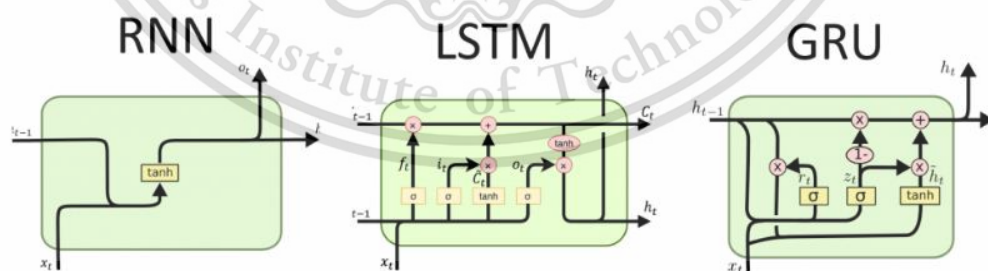


Figure 2. 17 (Left) Recurrent Neural Network, (Middle) Long-Short Term Memory Units,(Right) Gate Recurrent Units

These gates can learn which data in a sequence is important to keep or throw away. For this reason, it can pass relevant information down the long chain of sequences to make predictions. Almost all state-of-the-art results based on recurrent neural networks are achieved on these two networks.

II. Long-Short Term Memory

In 1997, Long Short-Term Memory Networks (mostly called LSTMs) were introduced by Hochreiter & Schmidhuber (1997) [9]. LSTMs are a special kind of RNN that is used to deal with the problem of long-term dependencies.

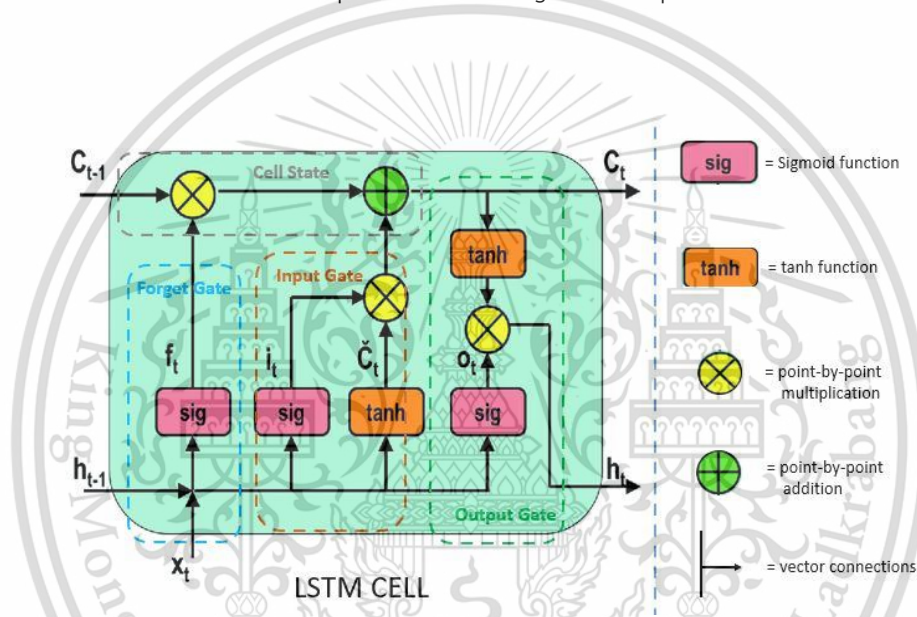


Figure 2. 18 Long-Short Term Memory (LSTM) Blocks

As per the problem of long-term dependencies, LSTMs enable RNNs to remember inputs over a long period of time. For this reason, the LSTM contains information in memory much like the memory of a computer. Hence, the LSTM can read, write, and delete information from its memory. In an LSTM, there are three gates consisting of input, forget, and output gates and one cell state, which are:

A. Forget gate

This gate decides what information should be deleted or kept. First, the information from the current input and hidden state is passed through the sigmoid function. Then, sigmoid generates values between 0 and 1. The closer to 0 means to

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

forget, and the closer to 1 means to keep. This value of the sigmoid function will later be used by the cell for pointwise multiplication.

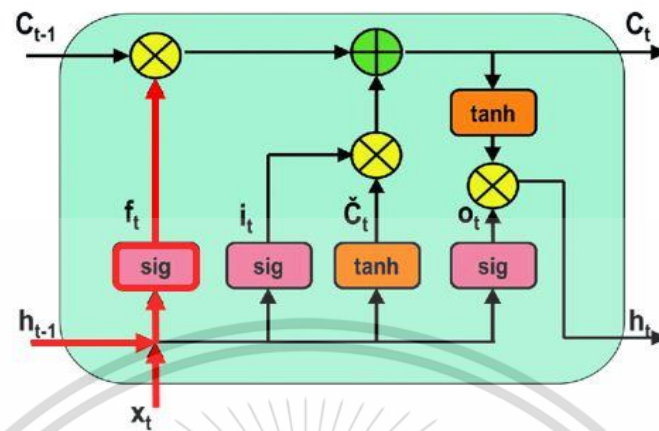


Figure 2. 19 Forget gate diagram in Long-short Term Memory

B. Input gate

First, the current state and previously hidden state are passed into the second sigmoid function that decides which values will be updated by transforming the values to be between 0 and 1. 0 means not important, and 1 means important. Next, the same information of the hidden state and current state will be passed through the tanh function to e values between -1 and 1 to regulate the network. Then, the output values generated form the activation functions are ready to pointwise multiplication.

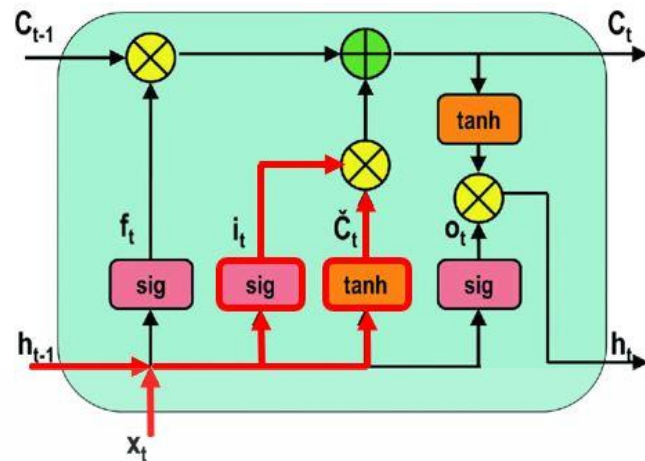


Figure 2. 20 Forget gate diagram in Long-short Term Memory

C. Cell state

Whenever there is enough information, it can be calculated in the cell state. First, the cell state gets pointwise multiplied by the forget vector. This has the possibility of dropping values in the cell state if it gets multiplied by values near 0. Then, the network takes the output value of the input vector and performs pointwise addition, which updates the cell state, giving the network a new cell state.

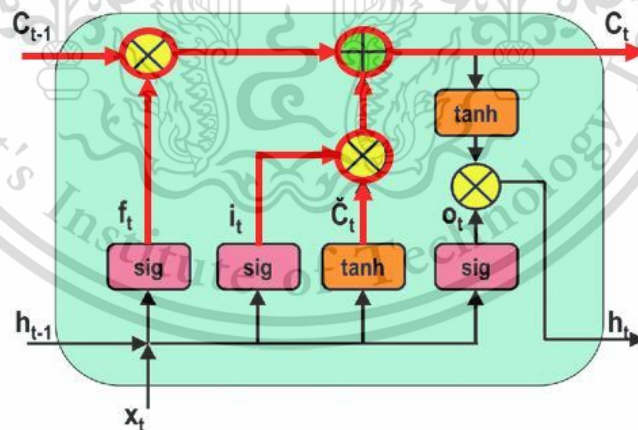


Figure 2. 21 Cell state diagram in Loong-Short Term Memory

D. Output gate

The output gate decides what the next hidden state should be. Remember that the hidden state contains information on previous inputs that is also used for

predictions. First, the values of the current state and previous hidden state are passed into the third sigmoid function. Then, the new cell state generated from the cell state is passed through the tanh function. Both outputs are multiplied by pointwise. Based upon the final value, the network decides which information the hidden state should carry. The hidden state is used for prediction. Finally, the new cell state and new hidden state are carried over to the next time step.

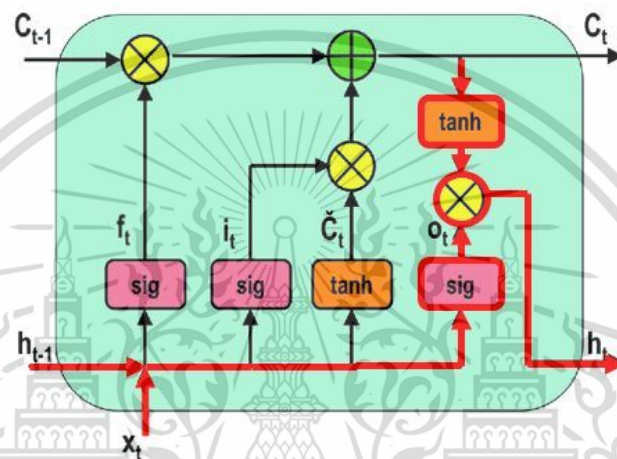


Figure 2. 22 Output gate diagram in Long-Short Term Memory

III. Gated Recurrent Units

A recurrent unit (GRU) was introduced by Cho, et al. in 2014 [10], which supports gating and a hidden state to control the flow of information, to solve the vanishing gradient problem faced by standard recurrent neural networks (RNN). GRU uses two gates, consisting of the update gate and the reset gate. Unlike LSTM, GRU does not have an output gate and combines the input and the forget gate into a single update gate.

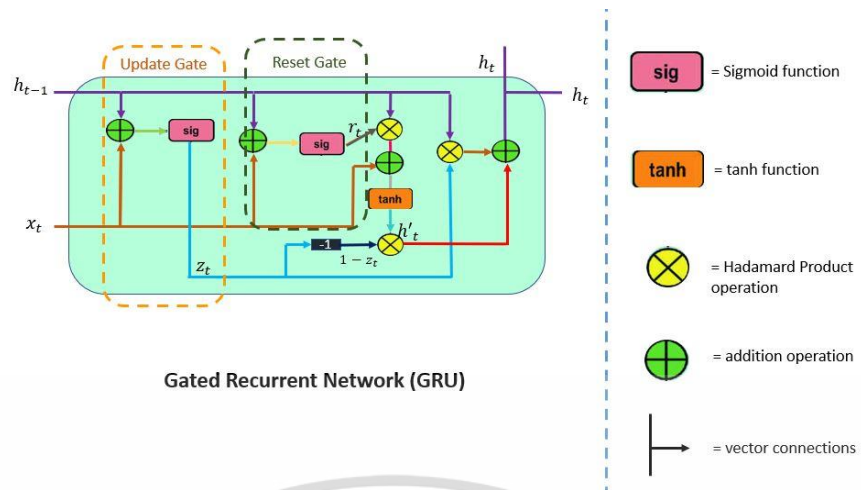


Figure 2. 23 Gated Recurrent Units (GRU) Block.

The update gate acts similar to the forget and input gate of an LSTM. It decides what information to delete and what new information to add.

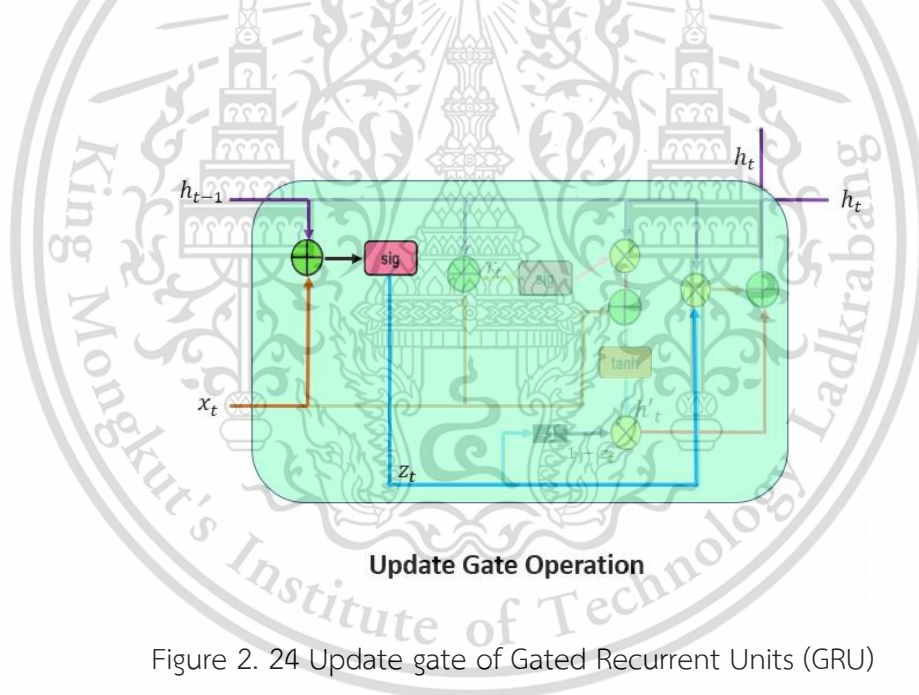


Figure 2. 24 Update gate of Gated Recurrent Units (GRU)

B. Reset Gate

The reset gate is another gate that is used to decide how much past information to forget.

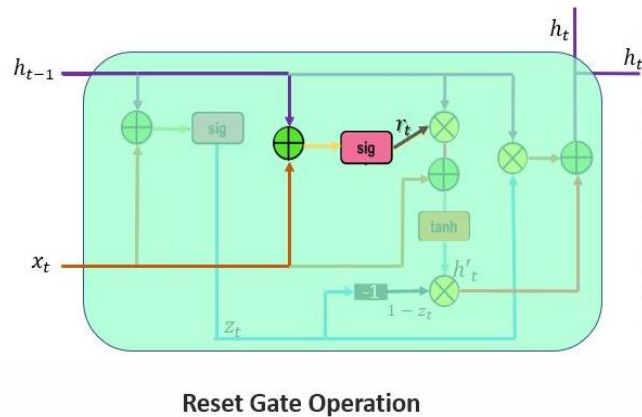


Figure 2. 25 Reset gate of Gated Recurrent Units (GRU)

2.2. Related Works

2.2.1. Convolution Neural Networks

Convolution neural networks (CNN) are a deep learning technique that is widely used in the healthcare system. For example, CNN could be used to extract features or classify cardiac arrhythmia from ECG signals. However, the healthcare system has a major problem, which is data imbalance. Because of collecting data and manipulating data, that can cause over-fitting on training datasets. The model cannot be generalized to new data despite fitting perfectly on training data. As a result, many researchers have attempted to solve this problem through data sampling and other techniques. Nowadays, the over-sampling technique is widely used to solve imbalanced datasets such as SMOTE. Verma D. et al. [11] used SMOTE to balance the dataset by making minority classes equal to the majority before training datasets in CNN.

In the last few years, CNN has been widely used to detect arrhythmia in ECG signals. Xiong Z. et al. [12] designed a sixteen-layered 1D CNN to detect AF that consisted of sixteen sequential skip connections by using single-lead ECG recordings from the Physionet Challenge 2017 dataset. Mostly, CNN is implemented to extract features from the ECG signals. Limam M. et al [13] designed a ten-layered 1D CNN to extract features for detecting AF consisting of an input layer, three convolution

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

layers, six downsampling layers, and one fully connected layer by using short single-lead ECG recordings from the Physionet Challenge 2017 dataset. Verma D. et al. developed using the same database to achieve cardiac arrhythmia classification by using 1D CNN consisting of the input layer, three convolution layers, six downsampling layers, and one fully connected layer by optimizing parameters in CNN architectures. Moreover, CNN can be implemented to detect QRS complexes from ECG signals. Yuen B. et al [14] designed a two-layered 2D CNN to extract features for detecting the QRS complexes on two channels of ECG signals, one of which is normal ECG, and the other is the gradient of channel 1, by using the ECG recordings from the MIT-BIH arrhythmia database and the European ST-T database.

In contrast, only a few studies have classified Paroxysmal Atrial Fibrillation (PAF) on ECG signals. However, Shashikumar SP. et al [15] designed a five-layered 1D CNN to extract features for detecting PAF, consisting of the input layer, two convolution layers, two downsampling layers, and one fully-connected layer by using the 24-hour Holter ECG recordings collected at the University of Virginia (UVA) Heart Station.

2.2.2 Recurrent Neural Networks

Recurrent neural networks (RNN) are deep learning techniques that are suitable for sequential or time-series data, such as ECG signals. For example, LSTM could be used to classify cardiac arrhythmia, or CNN and LSTM could be used together.

An RNN is a type of feed-forward artificial neural network consisting of an input layer, a hidden layer, and an output layer. However, the limitation of RNN is that it is capable of learning long-term dependencies. Hence, there is a subcategory of RNN to solve the problem that is called "Long-short term memory networks (LSTM)". Also, the LSTM can solve the vanishing gradient [16].

In general, the ECG signals are time-series data characterized by long-term dependencies. There are several pieces of research implementing RNN on ECG signals to classify arrhythmia. Klosowski G. et al [17]. designed a five-layered LSTM to

classify cardiac arrhythmia consisting of the input layer, an LSTM layer, and two fully connected layers by using the ECG recordings generated by FLUKE "ProSim 4 Vital Sign and ECG Simulator."

On the other hand, there is little research implementing RNN on the ECG signals to detect PAF. But Shashikumar SP. et al. used the ECG recordings from the 24-hour Holter ECG recordings made at the University of Virginia (UVA) Heart Station to create bi-directional recurrent neural networks (BRNN). These networks have a forward layer, a backward layer, and a fully connected layer.



2.2.3 Literatures Review Summary

Authors	Year	Titles	Purpose	Models	Evaluation
Mohamed Liman, Frederic Precioso	2017	Atrial Fibrillation Detection and ECG Classification based on Convolution Recurrent Neural Network	To detect Atrial Fibrillation (AF)	CNN + LSTM	F1 Score: 0.77
Jen Hong Tan, et al. [18]	2018	Application of Stacked Convolution and Long Short-Term Memory Network for Accurate Identification of CAD ECG Signals	To detect coronary artery disease (CAD)	CNN + LSTM	Accuracy: 0.95 Recall: 0.95 Precision: 0.95 F1 Score: 0.95
Shu Lih Oh, et al. [19]	2018	Automated diagnosis of arrhythmia using combination of CNN and LSTM techniques with variable length Heart beats	To classify types of arrhythmias	CNN + LSTM	Best Performances is CNN + LSTM without Dropout Accuracy: 0.98 Specificity: 0.98

Authors	Year	Titles	Purpose	Models	Evaluation
Brsnan Yuen, Xiadai Dong, Tao Lu	2019	Inter-Patient CNN-LSTM for QRS Complex Detection in Noisy ECG signals	To detect QRS Complex	CNN + LSTM	Recall: 0.97 Precision: 0.95 F1 Score: 0.96
Supreeth P. Shashikumar, et al.	2019	Detection of Paroxysmal Atrial Fibrillation using Attention-based Bidirectional Recurrent Neural Networks	To detect Paroxysmal Atrial Fibrillation (PAF)	CNN + LSTM	Accuracy: 0.94 Specificity: 0.95 AUC precision: 0.84
Chen Chen, et al. [20]	2019	Automated arrhythmia classification based on a combination network of CNN and LSTM	To classify types of arrhythmias	CNN + LSTM	Accuracy: 0.99 Recall: 0.97 Precision: 0.97 Specificity: 0.99

Authors	Year	Titles	Purpose	Models	Evaluation
Hao Dang, et al. [21]	2019	A Novel Deep Arrhythmia-Diagnosis Network for Atrial Fibrillation Classification using Electrocardiogram Signals	To detect Atrial Fibrillation (AF)	CNN + LSTM	Accuracy: 0.96 Recall: 0.99 Specificity: 0.97
Geogios Petmezas, et al. [22]	2020	Automated Atrial Fibrillation Detection using a Hybrid CNN-LSTM Network on Imbalanced ECG Datasets	To classify types of arrhythmias	CNN + LSTM	Best Performance is using gamma (Focal Loss's Coefficient) equal to 2 Precision: 0.97 Specificity: 0.99
Fengying Ma, et al. [23]	2020	An Automatic System for Atrial Fibrillation by using a CNN-LSTM Model	To detect Atrial Fibrillation (AF)	CNN + LSTM	Accuracy: 0.97 Recall: 0.97 Specificity: 0.970 F1 Score: 0.95

Authors	Year	Titles	Purpose	Models	Evaluation
Yongjie Ping, Chao Chen, et al. [24]	2020	Automatic Detection of Atrial Fibrillation Based on CNN-LSTM and Shortcut Connection	To detect atrial fibrillation (AF)	CNN + LSTM	Best Performances is 8SL segment every 10s and 8 shortcuts Accuracy: 0.85 Recall: 0.87 Precision: 0.92 Specificity: 0.91 F1 Score: 0.89
Aboli N. Londhe, Mithilesh Atulkar [25]	2020	Semantic segmentation of ECG waves using hybrid channel-mix convolutional and bidirectional LSTM	To segment the ECG components i.e., P-wave, QRS Complex, T-wave	CNN + LSTM	Accuracy: 0.95 Recall: 0.93 Precision: 0.93 Specificity: 0.95 F1 Score: 0.93

Chapter 3

Methodology

3.1 Introduction

In this study, an algorithm for deep learning has been proposed. It consists of the steps shown in Figure 3.1: segmenting, removing noise, normalizing, oversampling, and training the data.

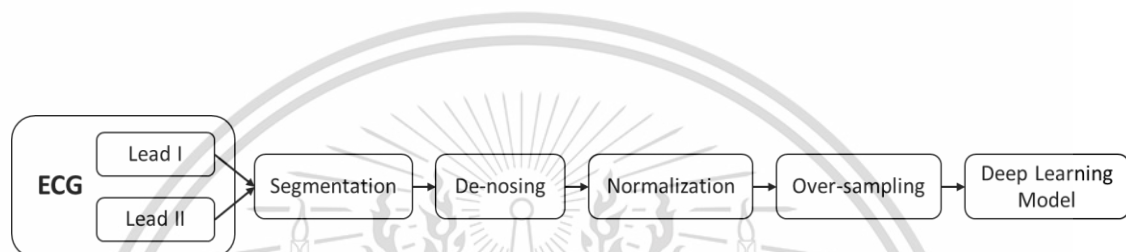


Figure 3. 1 Flowchart of this Study

3.2 Data Description

In this study, the ECG datasets and annotation files were downloaded from the PhysioNet Challenge 2021 [26] dataset consisting of 1416 samples of two-lead ECG (lead I and lead II), which is sampled at 200 Hz and each sample contains a variable range in duration. The dataset contains three types of ECG signal with different numbers of samples, consisting of 720 samples of non-atrial fibrillation, 467 samples of persistent fibrillation, and 227 samples of paroxysmal atrial fibrillation.

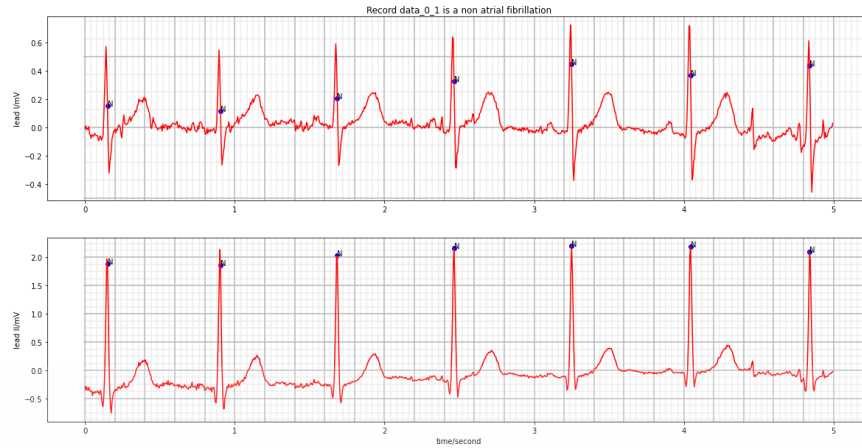


Figure 3. 2 Example lead-I ECG of Non-Atrial Fibrillation

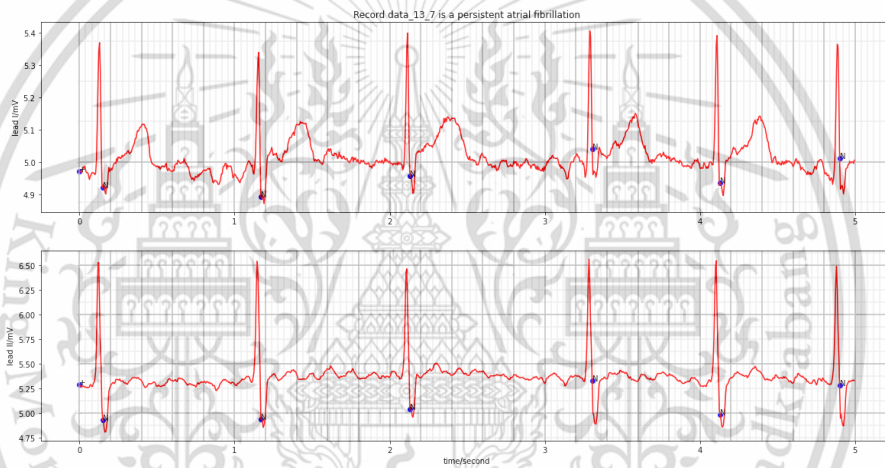


Figure 3. 3 Example lead-I ECG of Persistent Atrial Fibrillation (AF)

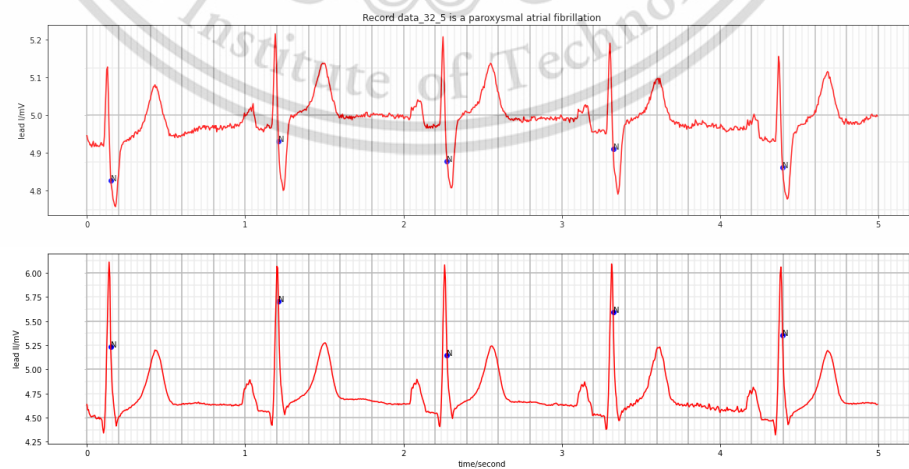


Figure 3. 4 Example lead-I ECG of Paroxysmal Atrial Fibrillation (PAF)

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

3.3 Data Pre-processing

The preprocessing of data is the most important step in the healthcare system, such as ECG recordings. For this study, the used data is ECG signal with noise and a problem of imbalance. Normally, noises always occur on the ECG signal by collecting the data from the device, such as electrode contact noise, baseline wander, and so on. In another case, data imbalance is a major problem in the healthcare system, which is mostly found from the data of patients much less than normal people. So, it needs to fix the problem of unbalanced datasets before sending the data to the next process.

However, the ECG data needs other preprocessing based on the specific data and the limitations of the data. First, the ECG data is sampled by trimming or segmenting the data due to the limitations of the data. Last, standardization is used to scale the ECG data before it is used to train a deep learning model.

3.3.1 Data Segmentation

There is a problem with the ECG signal, which varies in length in each record. Some records have a length that will lead to an additional cost of training. Hence, it is necessary to split the data to train the deep learning model.

In this study, the researcher will split the 15-minute segment into consecutive 30-second windows. First, the deep learning model will train and tune hyperparameters using the first 30 seconds of ECG data. Then, the tuned model will train and test for consecutive 30 second windows. The schematic diagram of data segmentation is shown in Figure 3.5.

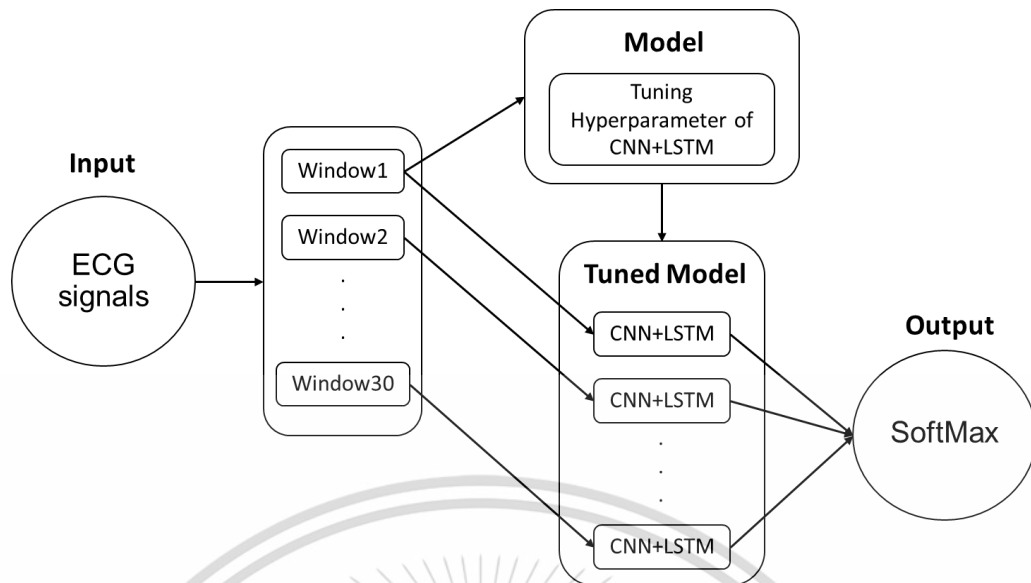


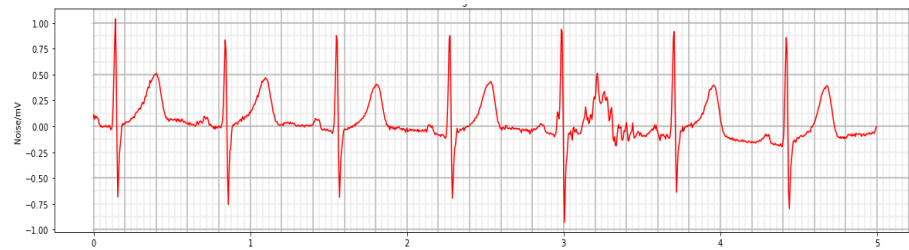
Figure 3. 5 Schematic diagram of data segmentation

3.3.2 Data De-noising

The ECG recordings are usually contaminated by different types of noise and artifacts such as power line interference, baseline wander, electrode contact noise, electrode motion artifacts, muscle contraction, electrosurgical noise, instrumentation noise, and so on [27]. Because this may decrease model performance, the ECG data must be de-noised before training on the deep learning model. There are many techniques for de-noising, such as bandpass filters, low-pass filters, high-pass filters, and so on. However, using each technique depends on what frequency of the signal will be removed.

In this study, the unsampled ECG data (both lead I and lead II) were unsampled to a frequency of 200 Hz. The segmented ECG data (both lead I and lead II) were then denoised using bandpass filtering, as illustrated in Figure 3.6, which is widely used to remove muscle noise, baseline wander, power line interference, and low- and high-frequency noise components, as well as to limit ADC saturation and address antialiasing. The frequency range of 1–40 Hz is set for this filter.

(a)



(b)

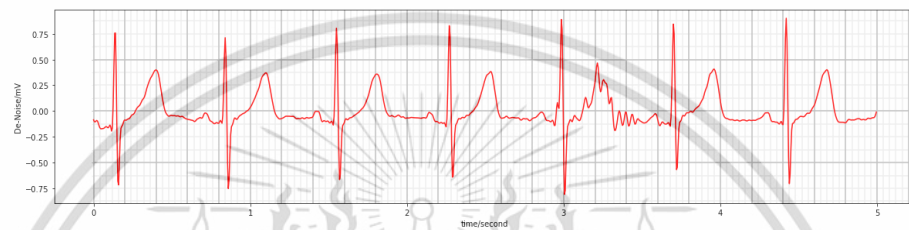


Figure 3. 6 (a) The raw lead I ECG signal (b) The filtered lead I ECG signal

3.3.3. Data Normalization

Normalization is a data preparation technique that is mostly used for scaling the data before feeding it into machine learning and deep learning models. The most widely used types of normalization are:

A. Min-Max scaling

This scaling will subtract the minimum value from each column's highest value and divide it by the range. After scaling, the value in each column will be updated to a minimum value of 0 and a maximum value of 1.

B. Standardization Scaling

This scaling will subtract the mean of each observation and then divide it by the standard deviation.

In this study, after the ECG data (both lead I and lead II) were de-noised and segmented, they were then normalized using Z-score normalization before being fed into deep learning as shown in Figure 3.6.

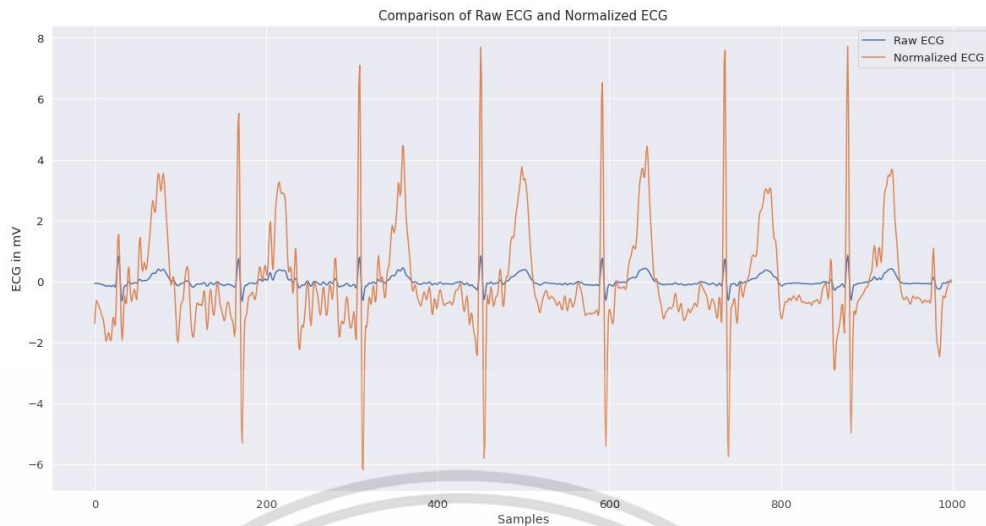


Figure 3. 7 Example of Comparison between Raw lead I ECG and Normalized lead I ECG (both are filtered with bandpass filtering)

3.3.4 Over-Sampling

Imbalanced datasets are a main problem in the healthcare system. This problem may cause overfitting in the training process. Thus, this problem should be fixed before feeding the data into the model with some techniques. Nowadays, there are three main techniques consisting of over-sampling, under-sampling, and combinations thereof, which are:

A. Over-sampling techniques.

This technique will duplicate data in the minority class or synthesize new data from the data in the minority class. Some of the more widely used and implemented over-sampling methods, such as Random Over-sampling, Synthetic Minority Oversampling Technique (SMOTE) [28], and so on.

B. Under-sampling techniques.

This technique will delete or select a subset of data from the majority class. Some of the more widely used and implemented under-sampling methods, such as random under-sampling, condensed nearest neighbour rule (CNN), and so on.

C. Combination techniques.

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

This technique will apply both over-sampling and under-sampling techniques together. Some of the more widely used and implemented combinations of sampling methods, such as SMOTE and Random Under-sampling, SMOTE and Tomek Links, and so on.

As shown in Figure 3.7, the ECG data is highly imbalanced in this study. The dataset has 1416 samples where the non-atrial fibrillation class (NON-AF) has 720 samples, persistent atrial fibrillation class (AF) has 467 samples, and paroxysmal atrial fibrillation class (PAF) has 229 samples. Since a highly imbalanced dataset may lead to bad results in a deep learning model, balancing the dataset for this study is the most necessary.

Hence, an over-sampling technique called SMOTE (Synthetic Minority Over-Sampling Technique) is used for balancing the dataset by increasing the class size of minority classes.

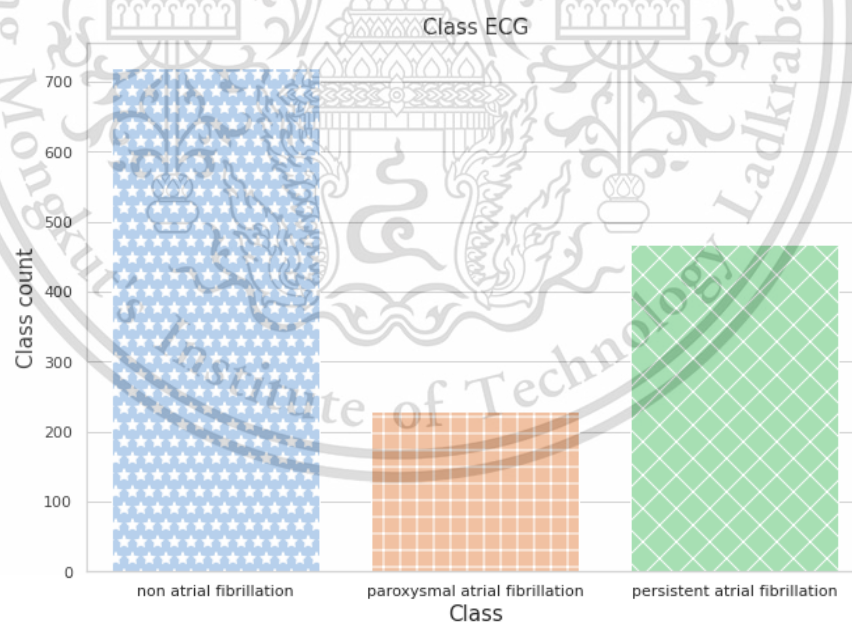


Figure 3. 8 Imbalanced dataset of ECG data

SMOTE will balance the dataset by making minority classes equal to the majority. In this case, the samples from the paroxysmal atrial fibrillation and

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

persistent atrial fibrillation classes are the same as those from the non-atrial fibrillation class. Hence, paroxysmal atrial fibrillation and persistent atrial fibrillation classes will be over-sampled to 720 samples.

3.3.5 Data Modeling

In this study, the model contains two main architectures, which are CNN and LSTM. First, the CNNs are used as a front-end to process the non-linear characteristics of the preprocessed input, including dimensionality reduction and extracting time information. Second, the LSTMs are used as a back-end to receive the abstracted data from the CNN and classify the outputs through the fully connected network.

The model has twenty layers, which are separated into two parts, consisting of Convolution Neural Networks (CNN) and Long-Short Term Memory (LSTM). First, the CNNs consist of four 1D convolution layers, four 1D Max-pooling layers, four batch normalization layers, and four dropout layers. Second, the LSTMs consist of two LSTM layers: one dropout layer and one dense layer. However, architectural details will be shown below in Table 1.1 and Figure 3.9.

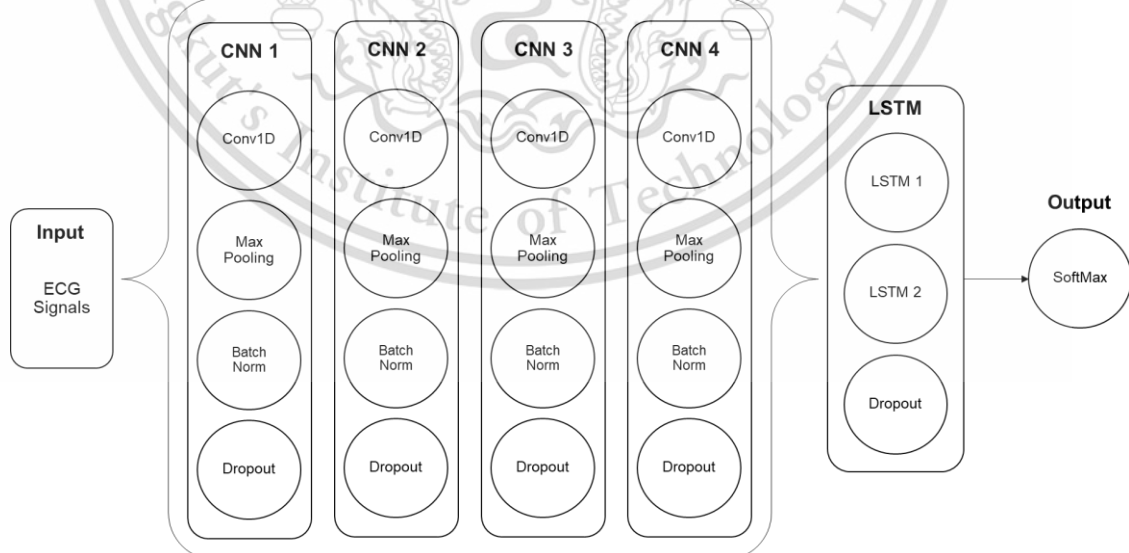


Figure 3. 9 Architecture of CNN and LSTM

Table 1. 1 Architecture of the deep learning model

Layers	Type	Parameters	Activation Function
0	ECG Input	-	-
1	1D Convolution	128	ReLU
2	1D Maxpooling	2	-
3	Dropout	0.1	-
4	BatchNormalization	-	-
5	1D Convolution	128	ReLU
6	1D Maxpooling	2	-
7	Dropout	0.1	-
8	BatchNormalization	-	-
9	1D Convolution	128	ReLU
10	1D Maxpooling	2	-
11	Dropout	0.1	-
12	BatchNormalization	-	-
13	1D Convolution	128	ReLU
14	1D Maxpooling	2	-
15	Dropout	0.1	-
16	BatchNormalization	-	-
17	LSTM	128	ReLU
18	LSTM	128	ReLU
19	Dense	3	SoftMax

Chapter 4

Main results and discussion

4.1 Experimental Setup

The ECG dataset consists of 1416 records, and each record has a variable length of the signal. Of them, 75% of them were used for developing the model as a training set, 15% of them were used for testing model performance as a validation set, and the remaining 10% of them were used as the holdout test set. The model was implemented on Keras with a TensorFlow backend and trained using Colab Pro.

In this study, the researchers divided this section into two parts: tuning the deep learning model's hyperparameters and measuring the model's performance.

4.1.1. Tuning Model

In this section, the model training experiment is conducted to evaluate the performance of the interested model and find the optimal parameter by comparing the loss and accuracy of the model. The cross-entropy loss (CE) is selected for the classification problem for the training criterion.

In the training process of a deep learning model, the initial setting of parameters before tuning hyperparameters is shown in Table 1.2. For tuning the hyperparameter, some parameters were fixed, and another parameter was tuned.

Table 1. 2 Initial Training Settings of the Deep Learning Model

Parameter	Lead I, Lead II
Cost Function	Cross-Entropy (CE)
Optimizer	Adam
Batch Sizes	16
Learning Rate	0.1
Epochs	20

A. Tuning Learning Rate

For both lead I and lead II of the ECG signals, the model is fixed epoch and batch size following Table 1.2 and tuned by learning rate with three values, i.e., 0.1, 0.01, and 0.001 as shown in Table 1.3. This model's optimal learning rate was determined by comparing loss and accuracy.

Table 1. 3 Loss and Accuracy of ECG signal by tuning Learning Rate

Learning Rate	Epochs	Batch Size	Lead I		Lead II	
			Loss	Accuracy	Loss	Accuracy
0.1			4.4039	0.2824	862.7887	0.3518
0.01	20	16	1.0647	0.4352	1.0198	0.5417
0.001			0.2616	0.9120	0.3861	0.9028

From Table 1.3, for lead I of the ECG signal, the lowest loss is 0.2616 and the highest accuracy is 0.9120. Hence, the learning rate of 0.001 is the optimal value for this model. For lead II of the ECG signal, the lowest loss is 0.3861 and the highest accuracy is 0.9028. Hence, the learning rate of 0.001 is the optimal value for this model.

B. Tuning Batch Size

For both lead I and lead II of the ECG signals, the model is fixed at an epoch of 20 and a learning rate of 0.001, and tuned by batch size with three values, i.e., 24, 32, and 64, as shown in Table 1.4. This model's optimal batch size was determined by comparing loss and accuracy.

Table 1. 4 Loss and Accuracy of ECG signal by tuning Batch Size

Batch Size	Epoch	Learning Rate	Lead I		Lead II	
			Loss	Accuracy	Loss	Accuracy
24			0.3103	0.9212	0.4083	0.9305
32	20	0.001	0.2623	0.9212	0.4112	0.9074
64			0.4402	0.8842	0.4165	0.8889

From Table 1.4, for lead I of the ECG signal, the lowest loss is 0.2623 and the highest accuracy is 0.9212. Hence, the batch size of 32 is the optimal value for this model. For lead II of the ECG signal, the lowest loss is 0.4083 and the highest accuracy is 0.9305. Hence, the learning rate of 24 is the optimal value for this model.

C. Tuning Epoch

In this section, for lead I of ECG signal, the model is fixed at a batch size of 32 and a learning rate of 0.001, and then tuned by epoch with three values, i.e., 30, 50, and 100, as shown in Table 1.5. For lead II of the ECG signal, the model was fixed at a batch size of 24 and a learning rate of 0.001, and then tuned by epoch with three values, i.e., 30, 50, and 100, as shown in Table 1.5. This model's optimal epoch was determined by comparing loss and accuracy.

Table 1. 5 Loss and Accuracy of ECG signal by tuning Epochs

Epochs	Batch Size	Learning Rate	Lead I		Lead II	
			Loss	Accuracy	Loss	Accuracy
30	32 (For Lead I)	0.001	0.3385	0.9074	0.3471	0.8981

50	24 (For Lead II)	0.3883	0.8889	0.4623	0.9259
100		0.3883	0.8889	0.3426	0.9167

The results in epoch 20 of Tables 1.4 and 1.5 for both lead I and lead II of the ECG signals were better than all the epochs in Table 1.5. Hence, an epoch of 20 is the optimal value for this model.

D. Tuning Dropout

In this section, for lead I of ECG signal, the model is fixed to a batch size of 32, a learning rate of 0.001, and an epoch of 20, and then tuned by dropout with three values, i.e., 0.2, 0.3, and 0.5, as shown in Table 1.6. For lead II of the ECG signal, the model was fixed at a batch size of 24, learning rate of 0.001, and epoch of 20, and then tuned by dropout with three values, i.e., 0.1, 0.2, and 0.5, as shown in Table 1.6. This model's optimal dropout was determined by comparing loss and accuracy.

Table 1. 6 Loss and Accuracy of ECG signal by tuning Dropout

Dropout	Batch Size	Epoch	Learning Rate	Lead I		Lead II	
				Loss	Accuracy	Loss	Accuracy
0.2	32 (For Lead I)			0.3019	0.9120	0.5189	0.8935
0.3	24 (For Lead II)	20	0.001	0.3838	0.8703	0.3767	0.8703
0.5	24 (For Lead II)			0.5343	0.7870	0.4859	0.8240

From Tables 1.4 and 1.6, for both lead I and lead II of the ECG signals, the results in dropout of 0.1 better than all the dropout in Table 1.6.5. Hence, the dropout of 0.1 is the optimal value for this model.

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

After the parameters of the deep learning model were tuned, the optimal parameters are shown in Table 1.7.

Table 1. 7 Final Training Settings of the Deep Learning Model

Parameter	Lead I	Lead II
Cost Function	Cross-Entropy (CE)	Cross-Entropy (CE)
Optimizer	Adam	Adam
Batch Sizes	32	24
Learning Rate	0.001	0.001
Epochs	20	20
Dropout	0.1	0.1

4.1.2. Measure the performance of deep learning model (without SMOTE)

In this section, the tuned model was measured in performance by using the ECG signals by splitting 15-minute of ECG signals (both lead I and lead II) into a consecutive 30-window as shown in Figure 4.1, and records decreased from 1416 records to 535 records. Next, each window is fed into the next data preprocessing except oversampling with SMOTE, which is described in Section 3.3. Then, the cleaned ECG signals are fed into the deep learning model and give the results, i.e., accuracy, precision, recall, and f1-score.

The model cannot classify paroxysmal atrial fibrillation (PAF) without using SMOTE for oversampling. The model only classifies two classes, which are Non-Atrial Fibrillation (NON-AF) and Atrial Fibrillation (AF). The results of classification metrics for paroxysmal classification metrics are zero in every window.

4.1.3. Measure the performance of deep learning model (with SMOTE)

In this section, the tuned model was measured in performance by using the ECG signals by splitting 15-minute of ECG signals (both lead I and lead II) into a consecutive 30-window as shown in Figure 4.1, and records decreased from 1416

records to 535 records. Next, each window is fed into the next data preprocessing, which is described in Section 3.3. Then, the cleaned ECG signals are fed into the deep learning model and give the results, i.e., accuracy, precision, recall, and f1-score [29], as shown in Table 1.8.

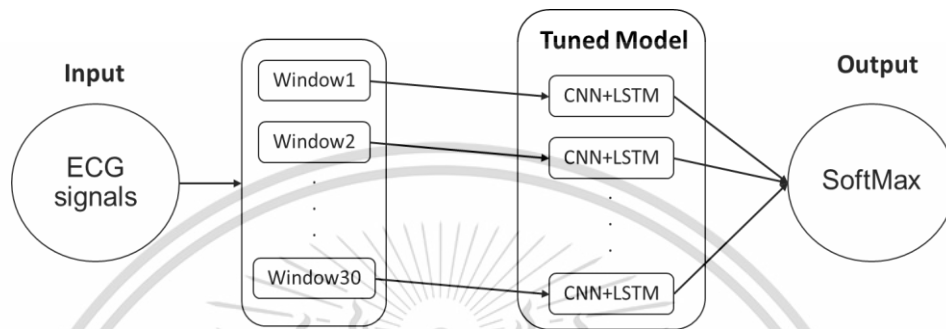


Figure 4.1 Schematic of Measuring Performance of Model

Table 1. 8 Classification Metrics of testing sets from Lead I and Lead II of ECG signal in 30 windows for Paroxysmal Atrial Classification

N	Lead I				Lead II			
	Accuracy	Precision	Recall	F1-Score	Accuracy	Precision	Recall	F1-Score
1	0.9401	0.9750	0.9750	0.9750	0.9743	0.9756	1.0000	0.9877
2	0.9316	0.9434	0.9615	0.9524	0.9401	0.9524	1.0000	0.9756
3	0.8974	0.8793	0.9808	0.9273	0.9658	0.9756	1.0000	0.9877
4	0.9230	0.9091	1.0000	0.9524	0.9743	0.9756	1.0000	0.9877
5	0.9658	0.9808	0.9808	0.9808	0.9572	0.9750	0.9750	0.9750
6	0.9401	0.9512	0.9750	0.9630	0.9230	0.9722	0.8750	0.9211
7	0.9401	1.0000	0.9500	0.9744	0.9658	0.9756	1.0000	0.9877
8	0.9572	0.9091	1.0000	0.9524	0.9316	0.8667	0.9750	0.9176

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

N	Lead I				Lead II			
	Accuracy	Precision	Recall	F1-Score	Accuracy	Precision	Recall	F1-Score
9	0.9658	0.9756	1.0000	0.9877	0.9572	0.9091	1.0000	0.9524
10	0.9230	0.9756	1.0000	0.9877	0.9572	0.9302	1.0000	0.9639
11	0.9401	0.9524	1.0000	0.9756	0.9658	0.9756	1.0000	0.9877
12	0.9230	0.9756	1.0000	0.9877	0.9487	0.9286	0.9750	0.9512
13	0.8974	0.9750	0.9750	0.9750	0.9735	0.9756	1.0000	0.9877
14	0.9658	0.9302	1.0000	0.9639	0.9658	0.9524	1.0000	0.9756
15	0.9059	0.9524	1.0000	0.9756	0.9829	0.9756	1.0000	0.9877
16	0.9743	0.9750	0.9750	0.9750	0.9572	0.9286	0.9750	0.9512
17	0.9658	1.0000	1.0000	1.0000	0.9401	0.9091	1.0000	0.9524
18	0.9402	0.9524	1.0000	0.9756	0.9487	0.9091	1.0000	0.9524
19	0.9230	0.8889	1.0000	0.9412	0.9743	0.9302	1.0000	0.9639
20	0.9487	0.9302	1.0000	0.9639	0.9487	0.9750	0.9750	0.9750
21	0.9316	0.9524	1.0000	0.9756	0.9572	0.9091	1.0000	0.9524
22	0.9230	0.9091	1.0000	0.9524	0.9481	0.9512	0.9750	0.9630
23	0.9316	0.9091	1.0000	0.9524	0.9401	0.9070	0.9750	0.9398
24	0.9059	1.0000	0.8250	0.9041	0.9230	0.8889	1.0000	0.9412
25	0.9829	1.0000	1.0000	1.0000	0.9743	0.9756	1.0000	0.9877
26	0.9829	1.0000	1.0000	1.0000	0.9658	0.9756	1.0000	0.9877
27	0.9658	1.0000	1.0000	1.0000	0.9829	0.9756	1.0000	0.9877

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

N	Lead I				Lead II			
	Accuracy	Precision	Recall	F1-Score	Accuracy	Precision	Recall	F1-Score
28	0.9487	1.0000	0.9500	0.9744	0.9743	0.9524	1.0000	0.9756
29	0.9487	0.9756	1.0000	0.9877	0.8119	0.7000	0.8750	0.7778
30	0.9572	1.0000	0.9500	0.9744	0.9230	0.8696	1.0000	0.9302
Ave	0.9415	0.9592	0.9832	0.9702	0.9518	0.9357	0.9858	0.9594

From Table 1.8, for lead I of ECG signal, the average of accuracy is 0.9415, the average of precision is 0.9592, the average of recall is 0.9832, and the average of f1-score is 0.9702. For lead II of the ECG signal, the average accuracy is 0.9518, the average precision is 0.9357, the average recall is 0.9858, and the average f1-score is 0.9594.

Table 1.9 Area under the curve for Paroxysmal Atrial Fibrillation Classification

Area Under the Curve (AUC)					
N	Lead I	Lead II	N	Lead I	Lead II
1	0.9997	1.0000	16	0.9994	0.9987
2	0.9968	1.0000	17	1.0000	0.9990
3	0.9974	1.0000	18	1.0000	0.9971
4	0.9916	1.0000	19	0.9997	0.9994
5	1.0000	0.9971	20	0.9914	0.9994
6	0.9981	0.9980	21	1.0000	0.9951

This material is reserved for educational use only, not allowed for commercial use.

Forbidden to modify the content, and cite the document when use.

N	Lead I	Lead II	N	Lead I	Lead II
7	0.9987	0.9984	22	1.0000	0.9909
8	1.0000	0.9881	23	0.9977	0.9964
9	1.0000	0.9990	24	0.9971	1.0000
10	1.0000	0.9997	25	1.0000	1.0000
11	1.0000	1.0000	26	1.0000	1.0000
12	1.0000	0.9948	27	1.0000	1.0000
13	0.9981	0.9994	28	0.9994	1.0000
14	1.0000	1.0000	29	0.9977	0.9997
15	0.9997	0.9997	30	0.9997	1.0000
Average (Lead I)		0.9987	Average (Lead II)		0.9983

From Table 1.9, for lead I of the ECG signal, the average ROC score is 0.9987. The average ROC score is 0.9983 for lead II of the ECG signal. As a result, both lead I and lead II have an area under the curve (AUC) greater than 0.9, which means this model has high test accuracy.

4.2 Summary and Discussion

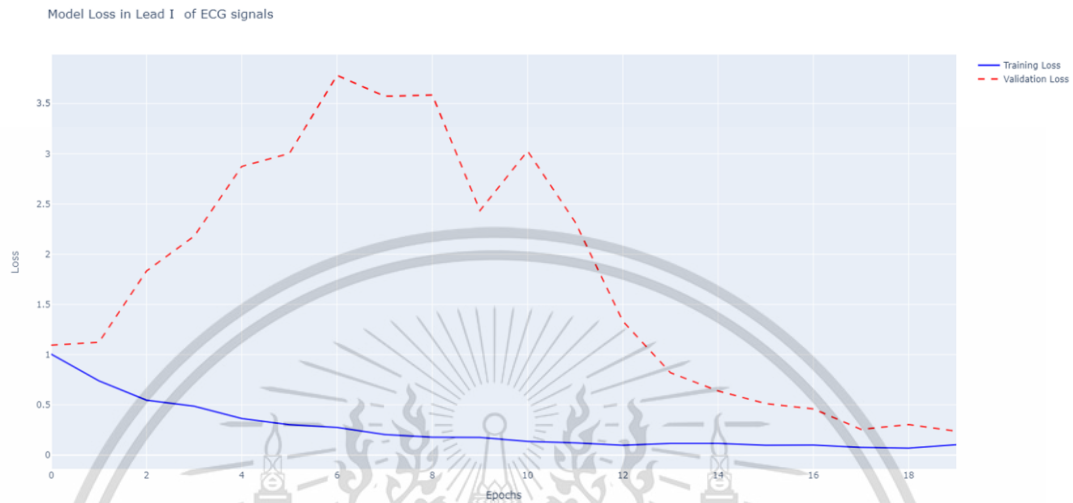
The results of the model, which were trained by 30 windows, are described in two topics, consisting of the learning curve and classification metrics. The learning curve is based on training sets and validation sets, and the classification metrics are based on testing sets.

4.2.1 Learning Curve

In this section, the learning curve of loss and the learning curve of accuracy are described by averaging the loss and accuracy of 30 windows in each epoch. In

general, the learning curve of loss should decrease in each epoch and the learning curve of accuracy should increase in each epoch.

(a)



(b)

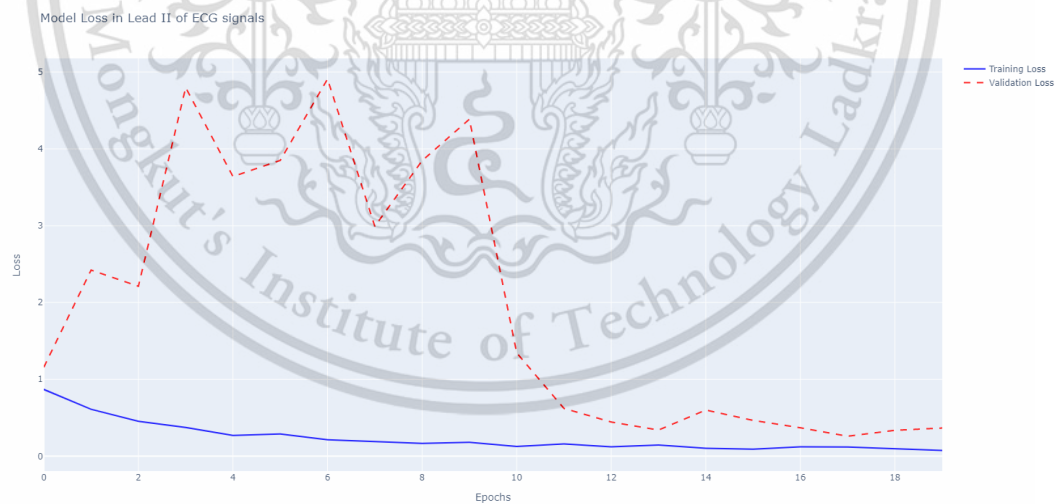
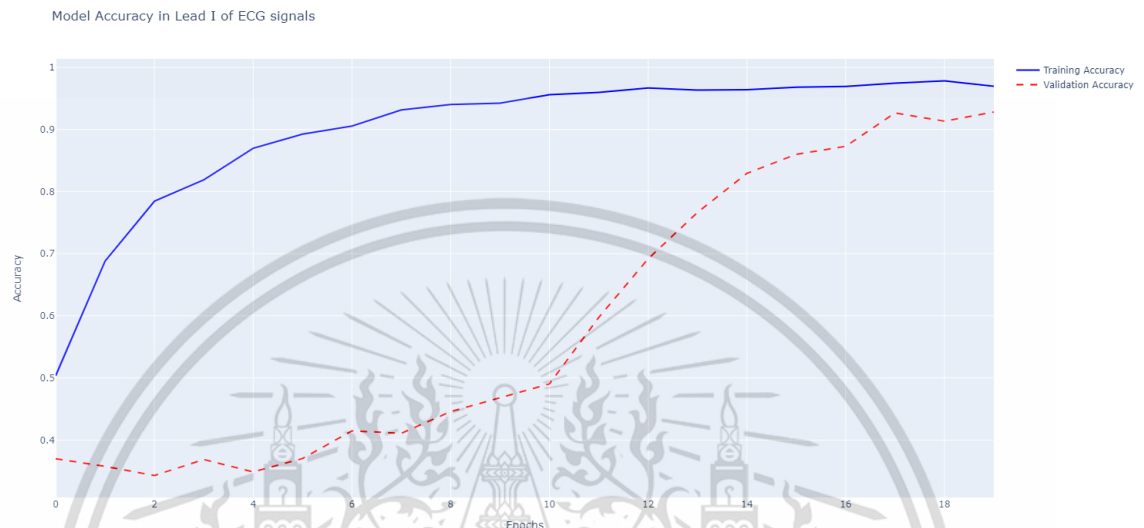


Figure 4. 2 (a) Learning Curve of Loss of Lead I (b) Learning Curve of Loss of Lead II

From Figure 4.2, the validation loss of lead I begins to decrease at epoch of 10, but the validation loss of lead II begins to decrease before epoch of 10. That

means lead II is trying to convergence faster than lead II. Hence, lead II of ECG signals tries to fit the deep learning model faster than lead I of the ECG signals

(a)



(b)

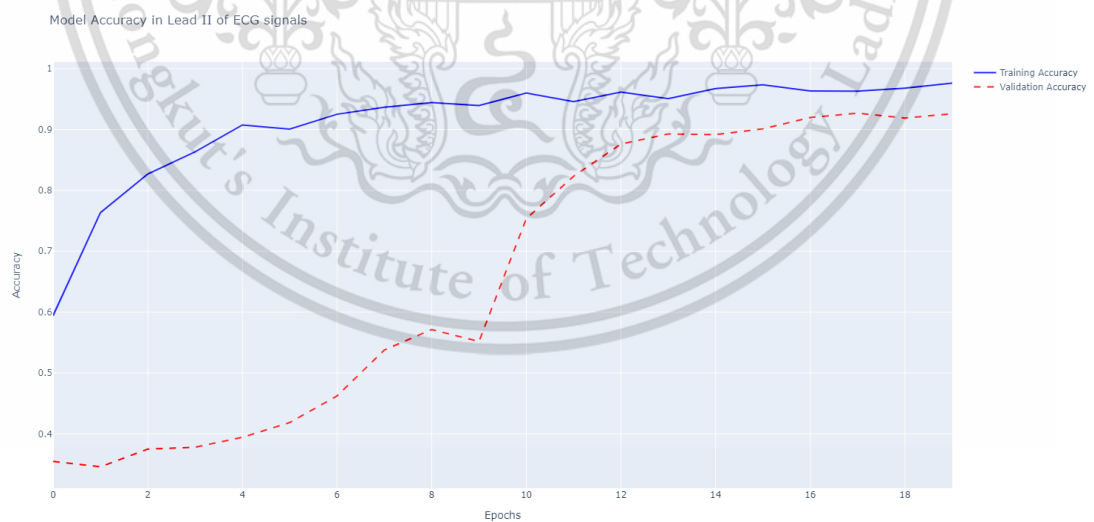


Figure 4. 3 (a) Learning Curve of Accuracy of Lead I (b) Learning Curve of Accuracy of Lead II

From Figure 4.3, the validation accuracy of lead I begins to increase at epoch 4, but lead II begins to increase at epoch 2. This means that lead II is attempting to reach convergence faster than lead I. Hence, lead II of ECG signals tries to fit the deep learning model faster than lead I of the ECG signals.

As a result, lead II of the ECG signals is trying to fit the deep learning model faster than lead I of ECG signals.

4.2.2 Classification Metrics

In this section, classification metrics are described, i.e., accuracy, precision, recall, and f1-score [29] by averaging all the results of testing sets of 30 windows in each epoch. From Table 1.8, the accuracy of lead II is greater than lead I of ECG signals, which means the deep learning model from lead II has an overall correct prediction more than the deep learning model from lead I. Next, the precision of lead I is greater than that of lead II of ECG signals, which means the correct prediction of the positive class (PAF) of lead I is better than that of lead II of ECG signals. Then, the recall of lead I is a little less than lead II of an ECG signal or nearly equal to lead II of an ECG signals, which means there is no difference between lead I and lead II of an ECG signal about the correct classification of positive class (PAF). Finally, the f1-score of lead I is higher than lead II of ECG signals, which means overall precision and recall of lead I are better than lead II of ECG signals. As a result, the classification of lead I is better than lead II of ECG signals, but lead I of ECG signals tries to fit the model slower than lead II.

For this study, the researcher understands the properties of ECG signals that are contaminated by noise in the collection process, so the ECG signal should be de-noised before training on the deep learning model. During the training process, the deep learning model can classify Paroxysmal Atrial Fibrillation (PAF) with over 90% accuracy, precision, recall, and f1-score in short periods of time and using a single lead of ECG signal. This model can be applied to any device, particularly a smart watch or an Apple Watch with an ECG function. This solution will be advantageous for decreasing the workload of the doctor.

Chapter 5

Conclusions and suggestions

5.1 Conclusions

For the study of paroxysmal atrial fibrillation classification from single-lead ECG using CNN and LSTM, the most necessary part of this study is preprocessing the ECG signals since the ECG signals consist of noise in many ways: imbalanced datasets, limitations of datasets, i.e., varying length of ECG signals in each record, and so on.

For the model training process, resources are important for training medical data, which uses more resources to train models. However, Colab Pro can be used for this situation.

As a result, the two main concerns for studying Paroxysmal Atrial Fibrillation Classification from Single-lead ECG using CNN and LSTM are data quality and quantity, as well as adequate resources.

5.2 Suggestions

The future study should be concerned with the resources and data in the first step. Another solution is to find a way to train the model and predict in real time on the cloud, such as using PySpark instead of Python.

References

- [1] <https://www.nhs.uk/conditions/cardiovascular-disease/>
- [2] International Statistical Classification of Diseases and Related Health Problems (ICD), <https://icd.who.int/browse10/2019/en>
- [3] Freedman, Ben, Tatjana S. Potpara, and Gregory YH Lip. "Stroke prevention in atrial fibrillation." *The Lancet* 388.10046 (2016): 806-817.
- [4] Roth, Gregory A., 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 (2018): 1736-1788
- [5] Campbell, Bruce CV, et al. "Ischaemic stroke." *Nature Reviews Disease Primers* 5.1 (2019): 1-22.
- [6] Blomström-Lundqvist, Carina, et al. "ACC/AHA/ESC guidelines for the management of patients with supraventricular arrhythmias—executive summary: a Report of the American College of Cardiology/American Heart Association Task Force on Practice Guidelines and the European Society of Cardiology Committee for Practice Guidelines (Writing Committee to Develop Guidelines for the Management of Patients with Supraventricular Arrhythmias) Developed in collaboration with NASPE–Heart Rhythm Society." *European Heart Journal* 24.20 (2003): 1857-1897.
- [7] Ghosh, Anirudha, et al. "Fundamental concepts of convolutional neural network." *Recent trends and advances in artificial intelligence and Internet of Things*. Springer, Cham, 2020. 519-567.
- [8] Sherstinsky, Alex. "Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network." *Physica D: Nonlinear Phenomena* 404 (2020): 132306.
- [9] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." *Neural computation* 9.8 (1997): 1735-1780.
- [10] Chung, Junyoung, et al. "Empirical evaluation of gated recurrent neural networks on sequence modeling." *arXiv preprint arXiv:1412.3555* (2014).
- [11] Verma, Dhvaj, and Sonali Agarwal. "Cardiac Arrhythmia Detection from Single-lead ECG using CNN and LSTM assisted by Oversampling." *2018 International*

Conference on Advances in Computing, Communications and Informatics (ICACCI). IEEE, 2018.

[12] Xiong, Zhaohan, Martin K. Stiles, and Jichao Zhao. "Robust ECG signal classification for detection of atrial fibrillation using a novel neural network." 2017 Computing in Cardiology (CinC). IEEE, 2017.

[13] Limam, Mohamed, and Frederic Precioso. "Atrial fibrillation detection and ECG classification based on convolutional recurrent neural network." 2017 Computing in Cardiology (CinC). IEEE, 2017.

[14] Yuen, Brosnan, Xiaodai Dong, and Tao Lu. "Inter-patient CNN-LSTM for QRS complex detection in noisy ECG signals." IEEE Access 7 (2019): 169359-169370.,

[15] Shashikumar, Supreeth P., Amit J. Shah, Gari D. Clifford, and Shamim Nemati. "Detection of paroxysmal atrial fibrillation using attention-based bidirectional recurrent neural networks." In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 715-723. 2018.,

[16] Hochreiter, Sepp. "The vanishing gradient problem during learning recurrent neural nets and problem solutions." International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 6, no. 02 (1998): 107-116.

[17] Kłosowski, Grzegorz, Tomasz Rymarczyk, Dariusz Wójcik, Stanisław Skowron, Tomasz Cieplak, and Przemysław Adamkiewicz. "The use of time-frequency moments as inputs of lstm network for ecg signal classification." Electronics 9, no. 9 (2020): 1452.,

[18] Tan, Jen Hong, et al. "Application of stacked convolutional and long short-term memory network for accurate identification of CAD ECG signals." Computers in biology and medicine 94 (2018): 19-26.

[19] Oh, Shu Lih, et al. "Automated diagnosis of arrhythmia using combination of CNN and LSTM techniques with variable length heart beats." Computers in biology and medicine 102 (2018): 278-287.

[20] Chen, Chen, et al. "Automated arrhythmia classification based on a combination network of CNN and LSTM." Biomedical Signal Processing and Control 57 (2020): 101819.

[21] Dang, Hao, et al. "A novel deep arrhythmia-diagnosis network for atrial fibrillation classification using electrocardiogram signals." IEEE Access 7 (2019): 75577-75590.

- [22] Petmezas, Georgios, et al. "Automated atrial fibrillation detection using a hybrid CNN-LSTM network on imbalanced ECG datasets." *Biomedical Signal Processing and Control* 63 (2021): 102194.
- [23] Ma, Fengying, et al. "An automatic system for atrial fibrillation by using a CNN-LSTM Model." *Discrete Dynamics in Nature and Society* 2020 (2020).
- [24] Ping, Yongjie, et al. "Automatic detection of atrial fibrillation based on CNN-LSTM and shortcut connection." *Healthcare*. Vol. 8. No. 2. MDPI, 2020.
- [25] Londhe, Aboli N., and Mithilesh Atulkar. "Semantic segmentation of ECG waves using hybrid channel-mix convolutional and bidirectional LSTM." *Biomedical Signal Processing and Control* 63 (2021): 102162.
- [26] <https://physionet.org/content/cpsc2021/1.0.0/>
- [27] Berkaya, Selcan Kaplan, et al. "A survey on ECG analysis." *Biomedical Signal Processing and Control* 43 (2018): 216-235.
- [28] Chawla, Nitesh V., et al. "SMOTE: synthetic minority over-sampling technique." *Journal of artificial intelligence research* 16 (2002): 321-357.
- [29] Hossin, Mohammad, and Md Nasir Sulaiman. "A review on evaluation metrics for data classification evaluations." *International journal of data mining & knowledge management process* 5.2 (2015): 1.

Author biography

Name Mr. Nopporn Subsermsong
Date of Birth 22 July 1996
Address 19/30 M.1, Soi. Tesaban 29, Sangchuto Rd., Thamuang,
Thamuang, Kanchanaburi 71110
Education (2018) Bachelor of Engineering in Mechanical Engineering
GPA 2.94
Kasetsart University

Scholarship -
Academic Publication(s) -

