THE USE OF ELECTROCARDIOGRAM FOR BIOMETRICS IDENTIFICATION BY CONVOLUTION NEURAL NETWORK

TITAYAWAT KHUMWONG

A Thesis Submitted in Partial Fulfillment of the Requirements for the

Degree of Master of Science in Applied Mathematics

Suranaree University of Technology

Academic Year 2021

การใช้ภาพคลื่นไฟฟ้าหัวใจสำหรับการระบุชีวมิติด้วยเครือข่ายประสาทสังวัฒนาการ

นายทิตยวัฒน์ คำวงษ์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต

สาขาวิชาคณิตศาสตร์ประยุกต์

มหาวิทยาลัยเทคโนโลยีสุรนารี

ปีการศึกษา 2564

THE USE OF ELECTROCARDIOGRAM FOR BIOMETRICS IDENTIFICATION BY CONVOLUTION NEURAL NETWORK

Suranaree University of Technology has approved this thesis submitted in partial fulfillment of the requirements for a Master's Degree.

Thesis Examining Committee

(Assoc. Prof. Dr. Eckart Schulz)
Chairperson

(Asst. Prof. Dr. Jessada Tanthannuch)
Member (Thesis Advisor)

(Assoc. Prof. Dr. Anirut Luadsong)
Member

(Asst. Prof. Dr. Benjawan Rodjanadid)
Member

(Assoc. Prof. Dr. Chatchai Jothityangkoon)
Vice Rector for Academic Affairs
and Quality Assurance

(Prof. Dr. Santi Maensiri)
Dean of Institute of Science

ทิตยวัฒน์ คำวงษ์ : การใช้ภาพคลื่นไฟฟ้าหัวใจสำหรับการระบุชีวมิติด้วยเครือข่ายประสาทสังวัฒนาการ (THE USE OF ELECTROCARDIOGRAM FOR BIOMETRICS IDENTIFICATION BY CONVOLUTION NEURAL NETWORK). อาจารย์ที่ปรึกษา : ผู้ช่วยศาสตราจารย์ ดร. เจษฎา ตัณฑนุช, 55 หน้า

คำสำคัญ: เครือข่ายประสาทเทียมแบบสังวัฒนาการ/ชีวมิติ/อีซีจี

การวิจัยครั้งนี้มีวัตถุประสงค์เพื่อประยุกต์ใช้เครือข่ายประสาทสังวัฒนาการเพื่อระบุชีวมิติจากข้อมูลคลื่นไฟฟ้าหัวใจ อีซีจี ทั้งนี้ได้ข้อมูลจาก Physionet จำนวน 310 ข้อมูล โดยเป็นข้อมูลของอาสาสมัครจำนวน 90 คน จากนั้นนำข้อมูลมาทำการตัดสัญญาณรบกวนที่มีความถี่สูงและต่ำมาเป็นข้อมูลสำหรับเรียนรู้ด้วยเครือข่ายประสาทเทียมแบบสังวัฒนาการ ผลการวิจัยพบว่าสามารถใช้ข้อมูลคลื่นไฟฟ้าหัวใจ อีซีจี เป็นข้อมูลชีวมิติสำหรับระบุตัวบุคคลได้ โดยมีความถูกต้องร้อยละ 99.52 และค่าสูญเสีย 0.12

สาขาวิชาคณิตศาสตร์	ลายมือชื่อนักศึกษา ______
ปีการศึกษา 2564	ลายมือชื่ออาจารย์ที่ปรึกษา J.Tanthanuch

THITAYAWAT KHUMWONG : THE USE OF ELECTROCARDIOGRAM FOR BIOMETRICS IDENTIFICATION BY CONVOLUTION NEURAL NETWORK. THESIS ADVISOR : ASST. PROF. JESSADA TANTHANUCH, Ph.D. 55 PP.

Keyword: BIOMETRICS/CONVOLUTION NEURAL NETWORK/IDENTIFICATION/ECG

This research is aimed at applying a convolutional neural network to biometric identification from an electrocardiogram (ECG) recording. ECG data consisting of 310 recordings from 90 persons was obtained from Physionet. The ECG data was prepared for having the first convolutional neural network by filtering high-frequency and low-frequency noise. The results show that the ECG data can be used for biometric identification with an accuracy of 99.52% and loss of 0.12.

School of Mathematics Student's Signature ________

Academic Year 2021 Advisor's Signature J. Tanthanuch

ACKNOWLEDGEMENTS

First of all, I would like to express my most profound appreciation to my thesis advisor Asst. Prof. Dr. Jessada Tanthanuch for encouragement, supervision through all obstacles, giving me the chance to do this research, and help along the way. I am very thankful to all teachers in mathematics at the Institute of Science, Suranaree University of Technology (SUT), for teaching and giving me a lot of knowledge. I am very grateful to my classmates for the support and discussion about this thesis that we have done together. I acknowledge to Development and Promotion of Science and Technology Talents Project (DPST) for financial aid.

Finally, this project would not have existed without the contribution of many people. I have to thank you a lot for the support and inspiration from everyone.

Thitayawat Khumwong

CONTENTS

CONTENTS (Continued)

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES (Continued)

CHAPTER I
INTRODUCTION

Biometrics is the measurement and statistical analysis of a person's unique characteristics. Biometrics verification is fast becoming common in corporate and public security systems, consumer electronics, and point of sale applications. Biometrics includes a reader or scanning device, software to transform biometric data into a digital format, and a database to store the biometrics data for comparison. There are two main types of biometrics identifiers. The first one is a physiological identifier, which includes things like facial, iris, voice recognition, vein recognition, retina scanning, and DNA matching. Another type is a behavioral identifier, which includes the unique ways in which individuals act like typing patterns or walking gait (Alexander S. Gillis, 2020).

The oldest and most widely used form is digital fingerprinting, which is unique and permanent except when a person suffers severe burns (Henry C. Lee and R. E. Gaensslen, 2000). The patterns making up our fingerprints are identifiable to regions known as Delta's forks and ridge endings. Each print contains up to 90 of these points but using just a dozen or so should be enough to identify one identical twin from another (Beavan, 2001). In a similar way, by the distinct structure of each individual's iris, reliable identification by iris scan is possible, however the technique is costly. A less expensive option is using a palm morphology reader; the only problem is that the form of a person's hand changes over time. This technique can be useful, however for quick identification purposes even in dusty or dirty conditions and this is why it is often used on construction sites and at harbors. Finger or hand vein recognition is another way of checking identity as infrared light shows the pattern unique to each individual. Already present at birth, these patterns remain with us until we die (Daniel Hartung, 2012). Finally, there's the good old fashioned signing on the dotted line of contracts of contracts and papers pressure point pace and pattern all make up the specific identity of our signature that can correctly identify a person.

Electrocardiography is the process of recording a graph of voltage versus time, of the electrical activity of the heart using electrodes placed on the skin. Binghamton State University in New York has been conducting research to use the heartbeat as a human identifier (Schmidt R, et al., 2017). Each person's heart rate is different, and cannot be changed unlike faces and fingerprints. The heartbeat can be used to unlock smartphones, lock-in social networks, open doors, accessing safes, and various applications in every industry. Due to the feature of the heartbeat, when a shock occurs or there is an indication of heart failure, Some wearable devices can send information to a car and emergency medical team to save the patient's life (Bonow RO, et al., 2019). This distinct feature of the heartbeat is a distinct advantage over fingerprint identification, which cannot indicate health status, only individual identification is possible (InformedHealth.org, 2006).

Digital Signal Processing is the process whereby real world phenomena can be translated into digital data for analysis, manipulation, and synthesis. This is done by sampling a signal with an instrument like a camera or a microphone, which in turn generates a sequence of numbers that represent continuous variables in a domain like time, space or frequency. Applications of digital signal processing can be used in various ways such as audio, graphics, data compression, machine learning, quantum computer, medical, and many more.

Since the ECG is unique for an individual, stable (having a same pattern for different activity) and easy to be measured, then it can be used for a biometric identification. This will show that mathematics can enhance biometrics' abilities. From the collected ECG-ID signals from Physionet containing 310 ECG recordings, obtained from 90 persons, this research will use this data to find a process by which ECG signals can be used in biometrics. The mathematics in the preprocessing process and convolution neural network in feature seeking process are tools that will be used to study the relation between ECG and biometrics for individuals.

1.1 Research objectives

To make a model for a biometrics identification with ECG signal by the convolution neural network algorithm.

1.2 Scope and limitations

This research uses the ECG-ID Database from Physionet. The database contains 310 ECG recordings, obtained from 90 persons. The records were obtained from volunteers (44 men and 46 women aged from 13 to 75 years who were students, colleagues, and friends of the author). The number of records for each person varies from 2 (collected during one day) to 20 (collected periodically over 6 months).

To improve biometrics by using an ECG signal, combination of deep learning and digital signal processing the in preprocessing process are the tools used to achieve goal.

1.3 Research procedure

The research work proceed as follows:

1. Study about biometrics, ECG, Convolution Neural Network and digital signal processing related with biometrics.
2. Create and analyze the model to determine whom a ECG signal belongs to.
3. Verify the accuracy of the model.

1.4 Expected results

A powerful process for identifying people by using ECG signals.

CHAPTER II
LITERATURE REVIEW

In this section, the knowledge of basic mathematics related with applications of biometrics and Electrocardiography (ECG) signals is presented. Digital Signal Processing is the important tool used to implement biometric procedures.

2.1 Biometric

Biometrics definition: Biometrics are biological measurement or physical characteristics that can be used to identify individuals. For example, fingerprint, face recognition, voice recognition, cardiac sound, handwriting dynamics and retina scans are all forms of biometrics technology.

Biometrics have two main types:

1. Behavioral biometrics: Behavioral biometrics is the field of study related to the measure of uniquely identifying and measurable patterns they use for secure authentication.

    - Voice recognition.
    - Mouse dynamics.
    - Handwriting dynamics.
    - Cardiac sounds.

2. Physiological biometrics: Physiological biometrics technology measures the unique pattern of a user's automatic bodily functions for purposes of identification, authentication, and analysis.

    - Face recognition.
    - Fingerprint.
    - Hand geometry.

- Retina and iris.
- Electrocardiogram(ECG/EKG).

This research focuses on electrocardiograms that explains in ECG part. Electrocardiograms is one of biometrics that has specific feature for each person. Because people have difference hearts, that means heart size, blood pressure, heart rate, valve are different, therefore the heart activity measured in wave form will different for each individual.

2.1.1 Heart and nervous system

This part is a discussion of the important concepts that will help to understand the event inside the heart as it generate a heartbeat. The first concept is the conduction system of the heart.

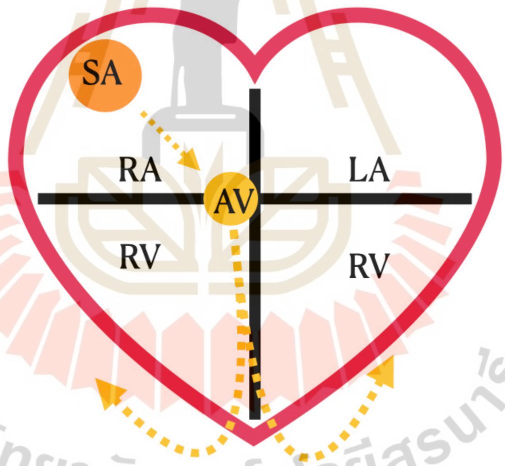


Figure 2.1 Conduction system of the heart. SA denotes sinoatral node. AV denotes atrial ventricular node.

The conduction system of the heart is the special cells in the left figure(2.1) that are responsible in conducting signals that cause the heart to contract. The first cell is called the sinoatrial node or the SA node; this cell is found within the right atrium of the heart. This is also known as the natural pacemaker of the heart because this is where the beginning of the conduction takes place. So, from the SA node the conduction goes to the atrial ventricular node or the AV node. By its name this cell can be found within the border of the right atrium and the right ventricle. The AV node is also known as the gatekeeper of the heart and the reason being is that it gets to decide whether what impulse to let through. Thus, if the SA node generates a weak impulse, the AV node will perceive that and will decide not to let the impulse go through and instead it will conduct its own impulse. From the AV node it goes through the single structure (dot line from SA node) called the His bundle and from the His bundle the cells will continue and will bifurcate into two separate cells, one to the left and one to the right. These are called the left and right bundle branches and from bundle branches they will extend within the apex of the heart. They are called the Purkinje fibers (MINT Nursing, 2017).

So, the conduction system of the heart at the SA node goes to the AV node next to the His bundle, the right and the left bundle branches and then eventually Purkinje fibers.

2.1.2 Electrocardiograms

The heart conduction system controls the generation and propagation of electrical signal or action potentials that cause the heart's muscles to contract and the heart to pump blood. This electrical activity can be measured at electrodes placed at specific points on the skin from which a composite recording is produced in the form of a graph. This recording is known as an electrocardiogram or ECG, is based on the German spelling (elektrokardiogramm – EKG). ECG records in graph of voltage versus time the small electrical changes that are a consequence of cardiac muscle depolarization followed by repolarization during each cardiac cycle (heartbeat).

As shown in Figure 2.2 the P wave represents atrial depolarization. During atrial depolarization the two atria are contracting. The QRS complex represents ventricular depolarization. During ventricular depolarization the ventriculars of the heat are contracting, depolarization is for contraction. That is P wave is for atria while QRS is for the two ventricles. The T wave represents ventricular repolarization, this is when the ventricles are relaxing. Keep in mind that every depolarization is always followed by repolarization so every contraction will be followed by relaxation. That being said, the question is where is atrial repolarization. It can be found in the QRS complex as can see the QRS com-

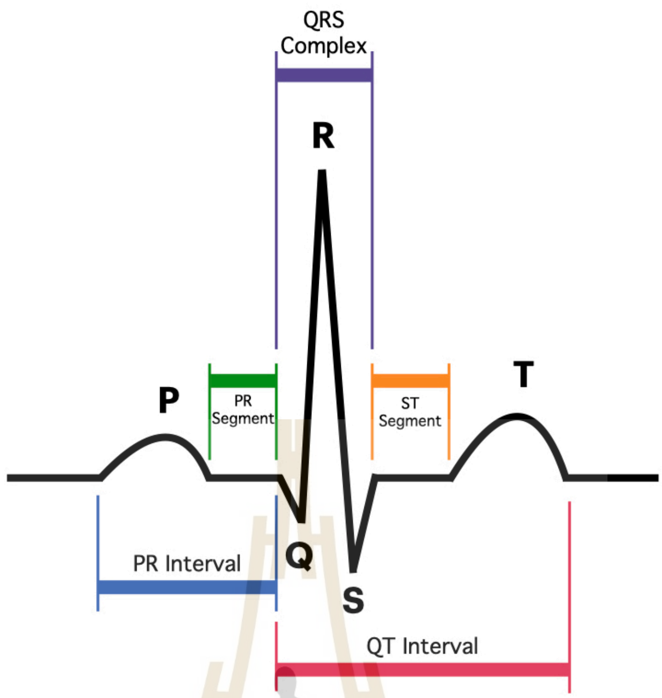


Figure 2.2 Electrocardiogram of a normal heart.

plex. It is a tall structure because ventricles tend to contract stronger than the atria. The QRS complex tends to mask the atria repolarization, so that being said atria repolarization and relaxation takes place after the P wave which can be found within the QRS complex (Lindsay M, 2013).

An ECG recording is a record of the voltage from the heart's activity the interval of interest. This information is on the electrode that is put on the skin. The electrode has the function to detect electrical impulses that came from a Cardiac pacemaker cell. The mechanic of the electrical impulse comes from the flow of ions in the body. The primary ions that cause the mechanic are sodium, calcium, and potassium. Individuals have differences and specific in the mechanic, so scientists try to use particular activities in the mechanic to identify them from biometric authentication.

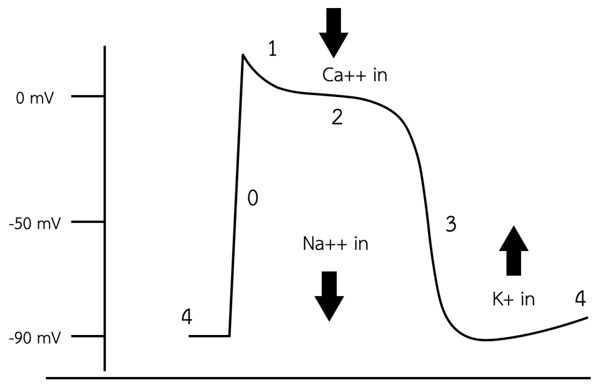


Figure 2.3 The mechanic of the heart's electrical impulse by the Ion flowing of Na, Ca, and K.

2.2 Digital signal processing

Before explaining about digital signal processing, one first has to know about analog and digital signals. Analog signals are the signals found in nature, that is the signals work on a continuous time domain such as signals of sound or temperature. Digital signals are different, they are signals that work on a discrete time domain, meaning that only after a certain period of time data will be collected. These signals could also be sound and temperature, however, in this case the curve would not be continuous.

Digital signal processing is processing of digital signals whereas analog processing is the processing of analog signals. Both types of processing have their own advantages, but digital signal processing is the only type of processing possible on today's computers.

2.2.1 Discrete Fourier transform.

Joseph Fourier was a French mathematician who had idea that a complex waveform, which could be light, heat or sound can be essentially broken down into its component parts which is just a group of simple sine waves. This technique can be use to map data between two domains and get important components in waveform to identify individuals with a unique wave in each person.

We can get various types of signals through sampling but once we have that information, how can we translate it into data that we are going to analyze, change or recreate. That is where the Fourier transform comes in. Fourier transform can break down the waveform into its components which is just a group of simple sine or cosine waves in the analog wave. In the digital case, one uses the Discrete Fourier Transform as shown in equation 2.1.

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-i2\pi kn/N} \qquad (K = 0...N-1) \tag{2.1}$$

- x denotes the signal level at various points in time, and X denotes the signal level at various frequencies.
- The signal level at frequency k is equal to the sum of the signal level at each time n multiplied by a complex exponential

Essentially the Discrete Fourier Transform is a mapping of data between two domains, input is the time domain and the output is the frequency domain. x_n represents the n^{th} element of the input and X_k represents the k element of output.

2.2.2 Infinite Impulse Response (IIR) filters

An infinite Impulse Response (IIR) filter is a digital filter that can be applied to many linear time-invariant systems. The ECG signals are recored by electrodes, which detect more than the electrical activity of the heart. Noise of ECG signals can be caused by electrodes impedance, respiration, movement around detector, muscle or electronics devices. To remove the distortions a bandpass IIR-filter is applied as shown in Equation 2.2. Commonly a digital filter is described in terms of a difference equation that defines how the output signal is related to the the input signal.

$$y_n = \sum_{i=0}^{P} b_i x_{n-i} - \sum_{j=1}^{Q} a_j y_{n-j} \tag{2.2}$$

where:

- P is the feedforward filter order;
- b_i are the feedforward filter coefficients of x_{n-i};

- Q is the feedback filter order;
- a_j are the feedback filter coefficients of y_{n-j};
- $x[n]$ is the input signal at time n;
- $y[n]$ is the output signal at time n.

2.2.3 Convolution Neural Networks

A convolution Neural Network or CNN is a tool that imitates the brain of humans. A human can distinguish objects by looking because our brain can remember objects that have different structure such as the number of corners, colors, shape and sound. Actually, a CNN is not the brain of human that can distinguish objects by looking but it can tell that it is by features of the object. However, CNN is a multilayer structure network, and each layer produces a data that can be used to be the feature in the last layer. CNNs have three main parts:

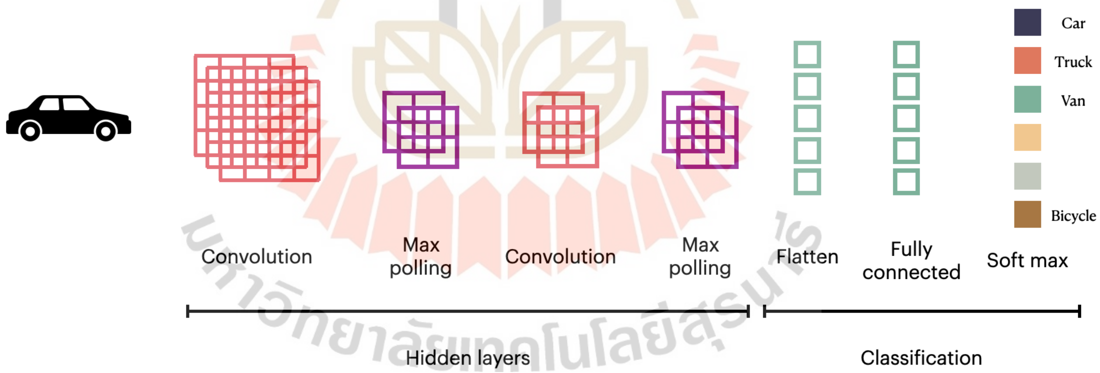


Figure 2.4 Convolution Neural Network.

1. A convolution layer is a part for seeking the feature of input, such as color, shape, border, amplitude, frequency, etc., by using a filter to operate with each element of the input and get the feature map that can be used in next step. For the general convolution, if the input of convolution is $N \times N$ sized with an $m \times m$ filter, the output has a matrix size $(N - m + 1) \times (N - m + 1)$ and the activation function is calculated when the convolution is performed.

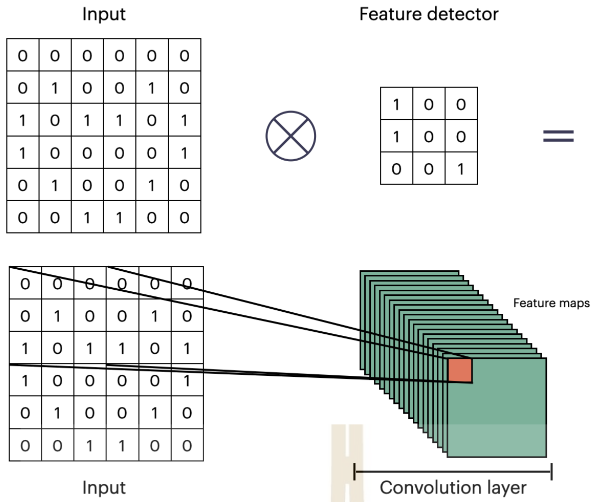


Figure 2.5 Convolution layer.

2. A pooling layer is a data scaling process or idea that instead of using all image data, one divides it into regions called tools and select some values from each post. (Like the way of looking at people when are sometimes look at the picture and tell what it is)

   In general, there are two types of Pooling Layers that one can choose from:

   - Max pooling: Select the maximum value from Pool size.
   - Average pooling: The average of pool size.

   The parameters that have to be set (Hyper parameters) are the move steps (Stride) and the size of the Pool.

3. Flattening is a map, sending a pooling feature matrix to a single vector.

4. Fully connected layers are an essential component of Convolutional Neural Networks (CNNs), which have been proven very successful in recognizing and classifying images for computer vision. The CNN process begins with convolution and pooling, breaking down the data into features, and analyzing them independently. The result of this process feeds into a fully connected neural network structure that drives the final classification decision.

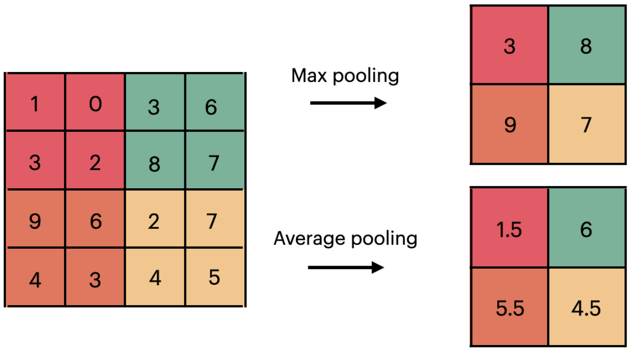


Figure 2.6 Max pooling and average pooling.

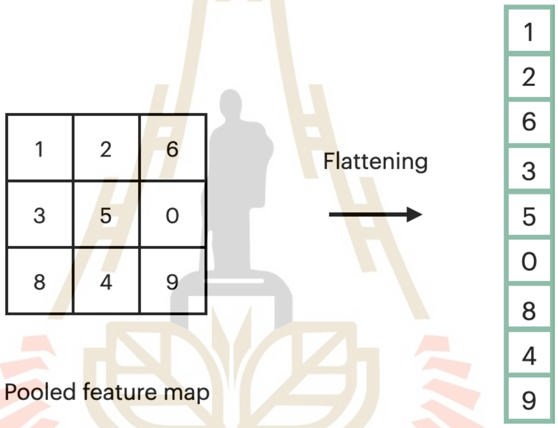


Figure 2.7 Flattening layer.

2.2.4 Optimization Algorithm

When discussing model creation for machine learning or deep learning, the optimizer is an integral part of the model development. In the present machine learning, AI, or deep learning, is the magic word that can solve every problem if one has enough data, which is true in the approximation theorem. But in real life, it is difficult to do because a complicated problem has to have a huge model and more computing power is required to find the features to solve them.

So, training the model is not just solving an equation, but it is a problem in the optimizing problem group. The model is just parameters that combine for the equation

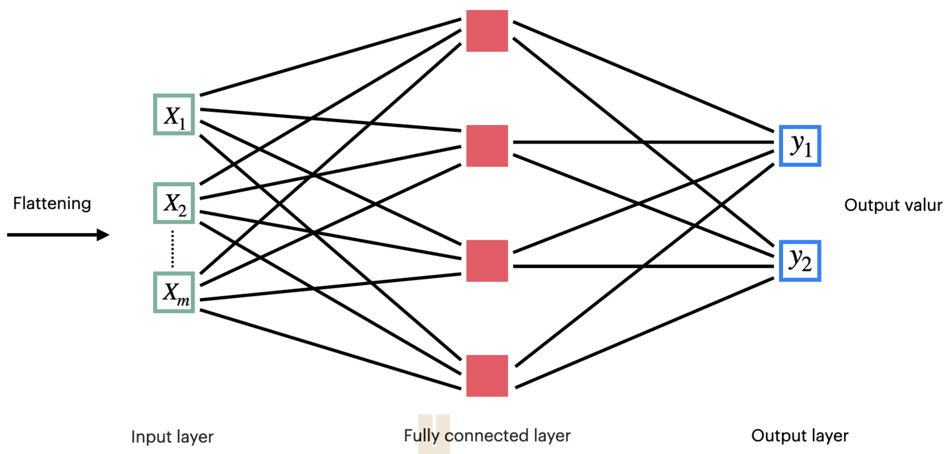


Figure 2.8 Fully connected layers.

and give the output related to input data.

Training the model means a little adjusting of parameters in the model, over and over, to find suitable parameters. The optimizer is the reason why the model is improved.

There are two types of optimizers

- First-order optimizer algorithms.
- Second-order optimizer algorithms.

The popular and famous optimizer.

- Gradient Descent.

  Algorithms to find the more minor and better angle(find cost Function = theta) like a normal equation, but a linear equation has the weakness that is it has to inverse matrix. For more parameters, have to have another equation to replace that is Gradient Descent.

$$\theta^{(nextstep)} = \theta - \eta \bigtriangledown_{\theta} J_{\theta} \quad (2.3)$$

- η is learning rate.
- $\theta \in R^d$ is model parameters. The $\theta^{(nextstep)}$ is next position and the θ is current position of the minimization procrss.

- $J(\theta)$ is objective function.

- Momentum. Adjusting the parameter has some problems finding the best optimal point, which is why a momentum optimizer is born. Momentum optimizer is coming to speed up for optimize of SGD.

- Adagrad is the optimizer that can adjust the applicable learning rate for parameters by updating a lot for a few parameters and a little updating for more parameters. That is why it is suitable for sparse data.

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{G_{t,ii} + \epsilon}} \cdot g_{t,i} \tag{2.4}$$

  - $G_{t,ii}$ is a diagonal metrix where each diagonal element (i, i) is the sum of square of the gradient $\theta_{t,i}$ up to time step t.
  - ϵ is a smoothing term that avoid division by zero (usually on the order of 10^{-8}).
  - η is learning rate.
  - $\theta_{t,i}$ is model parameters.
  - $g_{t,i} = \bigtriangledown_\theta J(\theta_{t,i})$ where $J(\theta_i)$ is objective function.

- AdaDelta is the next generation of Adagrad. It can reduce the decaying learning rate in AdaGrad by calculating's collection limit of Gradient to modify the weight that will occur.

$$\triangle\theta_t = -\frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} g_t \tag{2.5}$$

  - Replace the diagonal metrix G_t with the decaying average over past square gradients $E[g^2]_t$

- Adam or Adaptive moment estimation is an optimizer that can adjust the learning rate for parameters at any time. It can reduce the decaying of gradients in any step just like the AdaDelta optimizer, and it can explain how the decaying average of gradients occurs like the momentum optimizer.

2.2.5 Activation function

Activation function or transfer function is the function that gets the output node in a neural network. A neural network's output is can like be a number between 0 to 1 or between -1 to 1, depending on the requirements of the problem to be sloved.

Activate function has two types

- linear function
- nonlinear function

Linear function

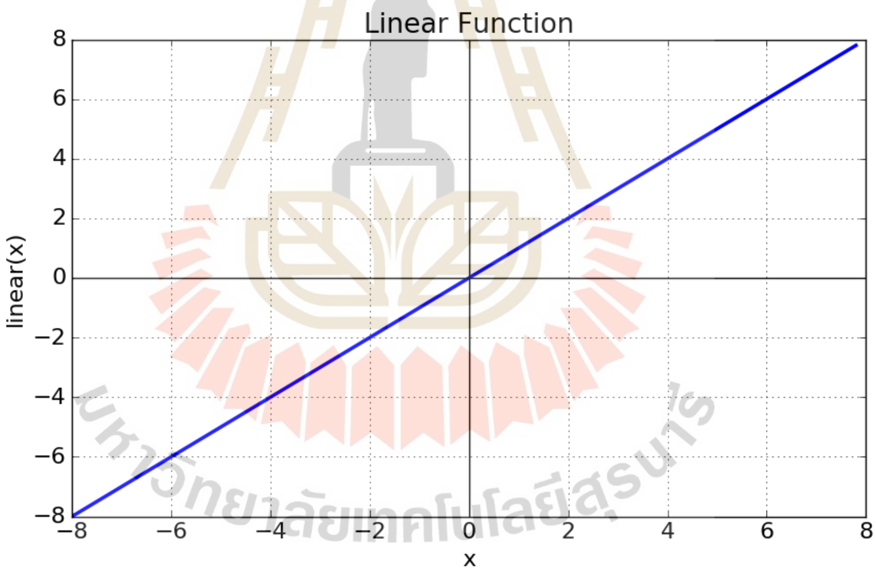


Figure 2.9 Linear function.

A linear function is just a function whose graph is a line.

Sigmoid function The sigmoid function has the form of an S curve, and it is a popular function because the output has a range between 0 to 1. So, the sigmoid function is suitable for finding the probability of output by the probability's value is between 0 to 1.

The weakness of sigmoid function is the time required when running the training process. So, the softmax function is more popular for the multiclass classification.

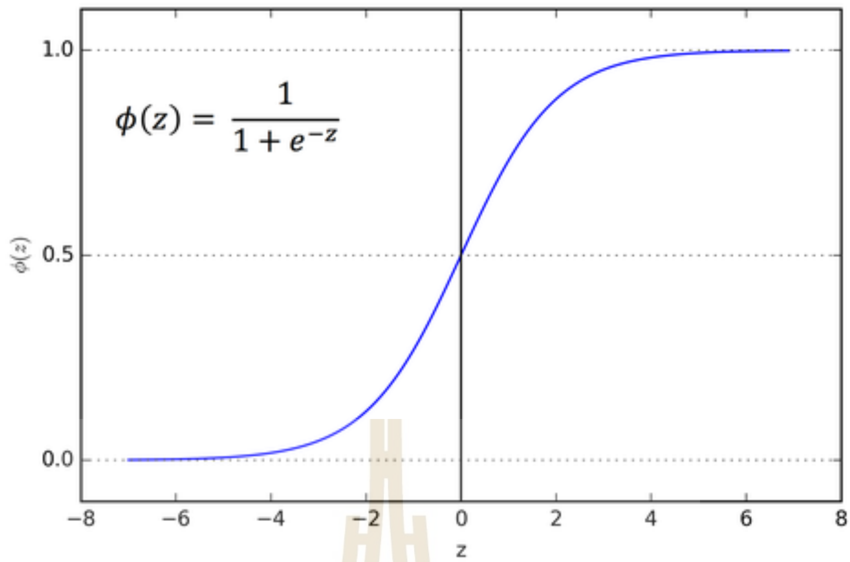


Figure 2.10 Sigmoid function.

The hyperbolic tangent function (tanh(x))

The **tanh**(x) function looks like a sigmoid function, but the output's range from -1 to 1.

ReLU function

ReLU function is the most popular for now, and it is used in various neural networks and deep learning.

The ReLU function is linear on positive input and zero on negative input.

Leaky ReLU function

The Leaky ReLU function is also linear on negative input, but at a different slope. The leaky help to expand the range of ReLU to -infinity to infinity, called randomized ReLU.

2.3 Performance

For biometrics identification accuracy is one important performance indicator or accuracy is the ratio of number of correct prediction and total number of input data. In the designed algorithm, the accuracy is calculated by using the mathematics formula given in equation 2.6:

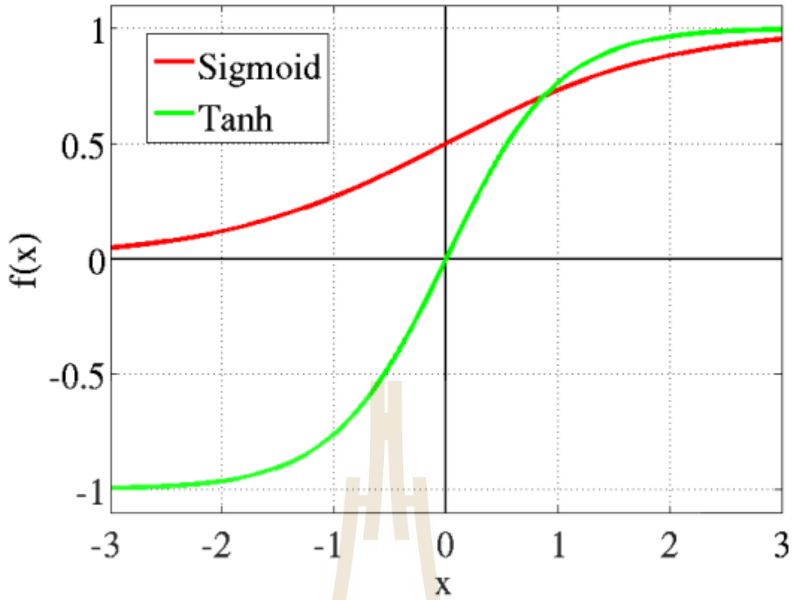


Figure 2.11 Sigmoid function and Tanh function.

$$Accuracy = \frac{Number\ of\ correct\ prediction}{Total\ number\ of\ data\ tested} \times 100 \tag{2.6}$$

To evaluate the model, the confusion matrix is one technique for scoring the model. Specifically, True positive (TP) represent the correctly accepted instances, False Positive (FP) represent incorrectly accepted instances, True Negative (TN) standing for correctly reject instances. and False Negative (FN) standing for incorrectly rejected instances. Precision (also called positive predictive value) is the fraction of relevant instances among the retrieved instances as shown in Equation 2.7. Precision computes the identification's correctness, and the relevance the ratio of positive identification.

$$Precision = \frac{TP}{TP + FP} \tag{2.7}$$

Recall (also known as sensitivity) is the fraction of relevant instances that were retrieved as shown in Equation 2.8.

$$Recall = \frac{TP}{TP + FN} \tag{2.8}$$

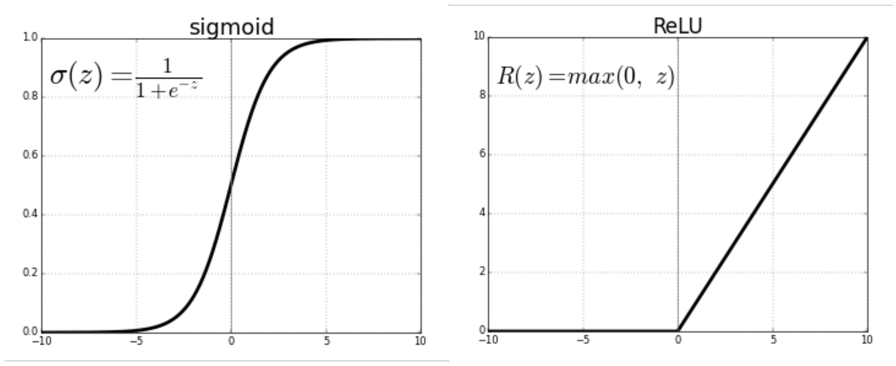


Figure 2.12 Sigmoid function and ReLU function.

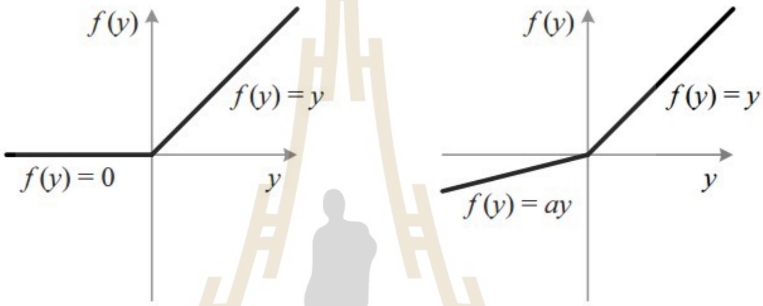


Figure 2.13 ReLU function and Leaky ReLU function.

Youden's J statistic (Youden's Index) is measurement of the receiver operating characteristic (ROC) curve for the performance curve of for the accuracy of a diagnostic test with ordinal or continuous endpoints. The Youden's Index is shown in equation 2.9

$$Youden's\,Index = [\frac{TP}{TP+FN}] + [\frac{TP}{TP+FP}] - 1 \tag{2.9}$$

2.4 Related researches

David Belo (2020) use pattern recognition for biometric to work on identifying individual by using ECG signals, obtained form MIT-BIH and CYBHi database. Deep Neural Network (DNN) is used for obtaining a high performance in this work. Temporal Convolution Neural Network (TCNN) and Recurrent Neural Network (RNN) is a processes that is used to improve current results in both identification (finding the registered person from a sample) and authentication (prove that the person is whom it claims) processes.

Lukasz Wiexlaw et al. (2017) present a discussion of biometric identification based on ECG signals data from Lviv Biometric Data Set by using Deep Neural Network (DNN), various signal pre-processing process and outlier detection techniques to improve overall system accuracy. This work measures the ECG signal on three fingers for easy application.

Deshmane (2018) designed an ECG-based biometric system that uses machine learning and deep learning techniques. This research uses data from MITDB, FANTASIA, NSRDB and QT, these data are collected from healthy participants and some with heart diseases such as arrhythmia and atrial fibrillation. SVM and KNN machine learning were used. CNN is used to compare performance of results of any database.

Kiran (2017) proposed an effective feature extraction method in which for each record of ECG signals. Most features depend on the P, Q, R, S and T points on the ECG graph and This research gets 72 different features. ANN was used for performance calculation and obtained ECG signals from MIT-BIH ECG-ID database signals.

Matteo (2019) proposed an effective approach for peak point detection and localization in noisy electrocardiogram (ECG) signals. This research uses Hilbert transform and thresholding technique for the detection of area inside the ECG signal, using wavelet transform to detect and localize R point. The MIT-BIH database is the source of data used in this work.

CHAPTER III
RESEARCH METHODOLOGY

The Electrocardiogram data in this research was obtained from PhisioNet. The ECG data consist of a record for 20 seconds in 500Hz digitized with 12-bit resolution over a nominal +10mV range, and each ECG data has annotated beats of the heart for 10 data (from R-wave and T-wave).

The ECG data used for this project were obtained from 90 volunteers aged between 13 to 75 years old for 44 males and 46 females who were students, colleagues, and friends of Tatiana Lugovaya who created and contributed to her master's thesis in Faculty of Computing Technologies and Informatics, Electrotechnical University (LETI). The ECG of each person varies from 2 to 20 data sets (collected during one day to 6 mouths).

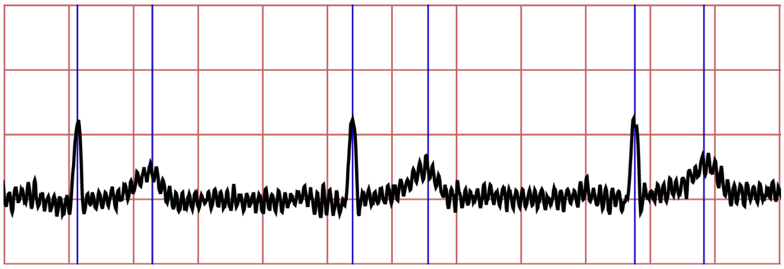
3.1 Tools

A computer program that is used in this research is Python language version 2.7 to prepare ECG data and create a model for identification by using the wfdb library keras, pandas, and numpy packages.

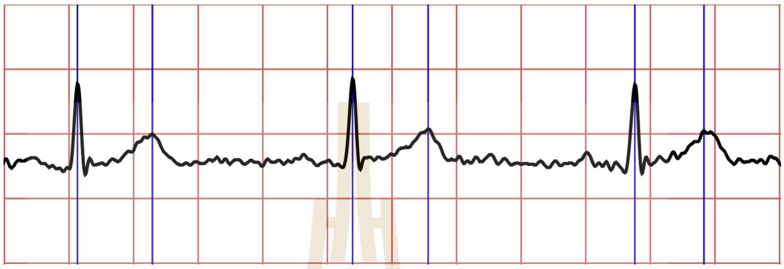
This research used a Macbook Pro CPU version I5 1.4 GHz quad-core with 8 GB of 2133 MHz LPDDR3 onboard memory running MacOS 64-bit and Intel Iris Plus Graphics 645.

3.2 Preparation of the ECG data for the Convolution Neural Network.

The ECG data available on Physionet.org is in the form of DAT files, and these data files cannot be used in a convolutional neural network in that form. A convolutional neural network is a technique that use to find features, create models, and a lot of advantages to making something new, but the data that can be use in a convolutional neural network has to be a specific data types such as JPG, JPEG, MP3, MP4, Video, and CSV. In this research, we have



(a) ECG signal with high and low frequency.



(b) Filtered ECG signal data.

Figure 3.1 Preparation of an ECG signal from the source and filtered ECG signal.

to transfer the ECG data to the CVS files first.

3.2.1 Development of a program for reading the ECG data and reducing noise.

The first step of the process is transferring Dat-files to CSV files by Python version used in all of this research. The ECG files obtained from Physionet have a lot of noise and contain high-frequency and low-frequency noise components. The Convolution Neural Network for classification or identification has two data sets for work: training and test sets. The training set consisted of the ECG data with high and low frequency noise removed, and the testing set was the original unfiltered ECG data

3.2.2 Develop part of Convolution neural network.

Convolution Neural Network is a technique that can seek features and create the model for identification for any person. The model for identification was create in this part.

Table 3.1 Model: Identification.

Layer (type)	Output shape	Parameter
convolution	(none, 1, 9995, 32)	192
max polling	(none, 1, 3331, 32)	0
convolution	(none, 1, 3327, 64)	10304
max polling	(none, 1, 1109, 64)	0
flatten	(none, 70976)	0
dense	(none, 128)	9085056
dense	(none, 90)	11610

- Total params : 9,107,162
- Trainable params : 9,107,162
- Non-Trainable prams : 0

Layer.

- The convolution layer accepts on input vector of length 1*9999 and give 32 filter outputs for tanh or hyperbolic tangent activation function.
- Pooling layer is the max pooling size 1*3 for size reduction of input.
- Flaten layer is the layer that transfers multi-dimension data to a vector.
- The first dense layer is the hidden layer to make the size of output 128 by tanh or hyperbolic tangent activation function.
- The second dense layer is the output layer of the network. This layer has to make the number of outputs equal to the number of ECG person data by a softmax activation function because the outcome is multi-class.

Configuration the model

The model was configures by categorical crossentropy loss-function. The model adjusts the value of the parameters during the training process to reduce this loss.

The optimizer is another variable that we have to assign to the model. We can either specify it as the optimizer name as in the example or create an optimizer object first. In this research, we use algorithms adam and adadelta.

We can specify additional metrics that we want the model to calculate during training and testing, with the most commonly specified metric being accuracy.

Now that we are ready for training, we instruct the model to start training with the fit command.

3.3 Test ECG biometrics by Convolution Neural Network.

This part will use a Convolution Neural Network for biometric identification for an individual.

3.3.1 Develop a computer program for training and identification of ECG data sets.

```
# fit model on training data
tensorboard = TensorBoard(log_dir="logs_personid/{}".format(time()))
earlystopping = EarlyStopping(monitor='val_loss', patience=499)
history = model.fit(X_train, Y_train, batch_size=30,
                    validation_data=(X_test, Y_test), nb_epoch=200,
                    verbose=1, callbacks = [earlystopping, tensorboard])
```

Figure 3.2 Training the model.

From the command 3.2, We divide the training data into 200 sessions (epochs), where each cycle uses all the data from the X train and Y train variables. For validation, we use test data.

However, we can set the training to stop before 200 cycles if the metric value matches the value we want. This technique also helps the model not to overfit. This technique relies on callbacks from EarlyStopping.

This research requires the model to monitor the loss from validation data, so if the loss in training cycles doesn't decrease or increase, stop training. Patience=499 means it stops once the loss increases by 499 cycles (epochs).

The last process is testing the ECG data in the model with two optimizers ten times and then finding the accuracy average of two optimizers that is Adam and Adadelta.

CHAPTER IV
RESULTS

In the Results section, we would like to present the differences between the two optimizers. The first is Adam or adaptive moment estimation, and the second one is Adadelta. The results from the two optimizers show little difference because the optimizer is just a tool for improving the model to be a better model by adjusting weight and bias that connect with neural and deep-learning use optimizer to amend the error and loss in the neural during the training process.

4.1 Results from the model with Adam optimizer

Adam optimizer is the most popular optimizer for classifications techniques in machine leaning and deep learning. This research uses the Adam optimizer ten times and gets the average accuracy equal to 0.9525806487 with 0.2460832926 loss.

4.2 Results from model with Adadelta optimizer

Adadelta optimizer is an optimizer that can reduce the decaying learning rate. We use this optimizer ten times and get the average accuracy equal to 0.9954838753 with 0.1217139436 loss.

The two tables above show the results of of the models results from two the optimizers (Adam and Adadelta). The first table shows the accuracy and loss of the model by using Adam optimizer. It has lower accuracy than the Adadelta model and has loss values greater than those Adadelta for every result. The average accuracy from the model by using Adam optimizer is equal to 0.9525806487 and 0.2460832926 loss.

Such an outcome can be used for security in a house such as a door, phone, microwave, computer, car, kitchen, or even restroom, but 95 percent accuracy is not enough to use for important medical information in health care

Table 4.1 Report of the model with Adam optimizer.

No.	Accuracy	Loss
1	0.9645161032676697	0.20183103872883704
2	0.9838709831237793	0.1335921866759177
3	0.9322580695152283	0.327027883116276
4	0.9709677696228027	0.15467047056844158
5	0.9580644965171814	0.23074037591295857
6	0.9354838728904724	0.30528024723452907
7	0.9161290526390076	0.39519031538117316
8	0.9451612830162048	0.26438847597568266
9	0.9483870863914498	0.27202804266445096
10	0.9709677696228027	0.17608388984395612
Average	0.9525806487	0.2460832926

Table 4.2 Report of the model with Adadelta optimizer.

No.	Accuracy	Loss
1	0.9967741966247559	0.1193623645651725
2	0.9903225898742676	0.12268781873487657
3	1.0000000000000000	0.11244822322360931
4	0.9935483932495117	0.13165235514602353
5	1.0000000000000000	0.11478796284044943
6	0.9935483932495117	0.12224157606401752
7	0.9967741966247559	0.12077939260390497
8	1.0000000000000000	0.11637334318891648
9	0.9935483932495117	0.1316745593663185
10	0.9903225898742676	0.1251318403790074
Average	0.9954838753	0.1217139436

that has to make sure who the user is, such as medical information, rocket launchers, and personal information.

Otherwise, the average accuracy of 99 percent from the model using an Adadelta optimizer might solve this problem.

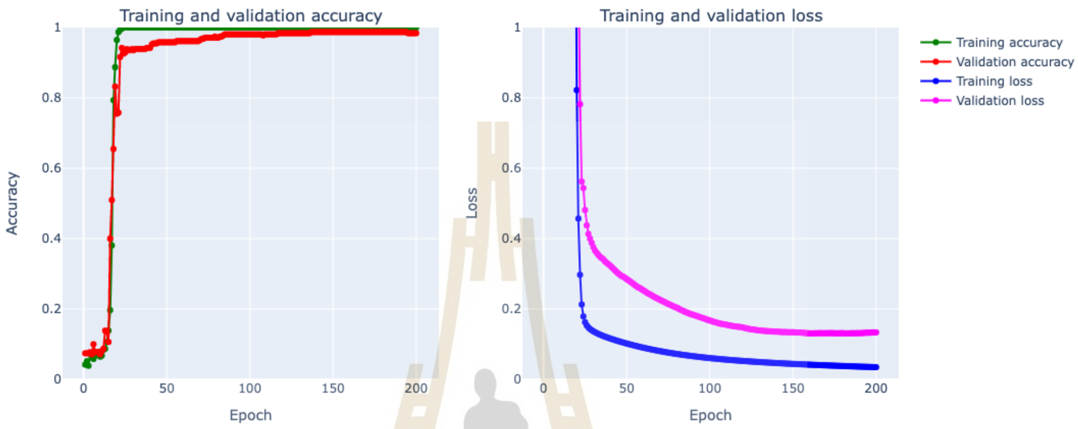
4.3 Comparation between two optimizer, Adam and Adadelta

The examples of the results from Adam and Adadelta optimizers shown in the graphs below 4.2 and 4.1 compare the difference between the best and worst results from any optimizer by the accuracy and loss from the model.

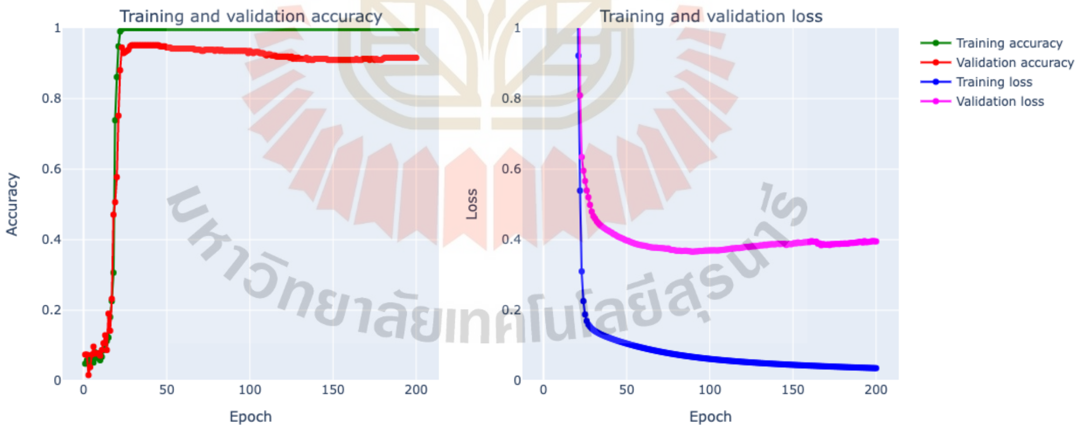
The first result is the best result 4.1a from Adam optimizer with an accuracy equal to 0.9838709831237793 and loss similar to 0.1335921866759177. The second result is a bad outcome 4.1b from Adam optimizer with that has an accuracy equal to 0.9161290526390076 and loss similar to 0.39519031538117316.

The first result is the best result 4.1a from Adadelta optimizer with an accuracy equal to 1 and loss similar to 0.11244822322360931. The second result is a bad outcome 4.1b from Adadelta optimizer with an accuracy equal to 0.9967741966247559 and loss similar to 0.12077939260390497

The results from the model with two optimizers can identify an individual, however the result by using Adadelta is more effective than Adam. The results show that the model with both optimizers can identify an individual, although the Adadelta optimizer has slights greater accuracy.

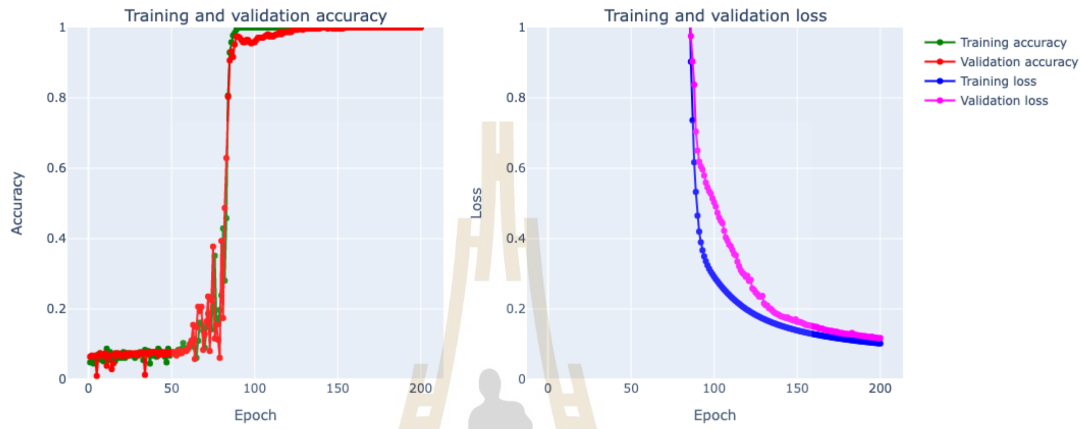


(a) The best result from Adam optimizer.

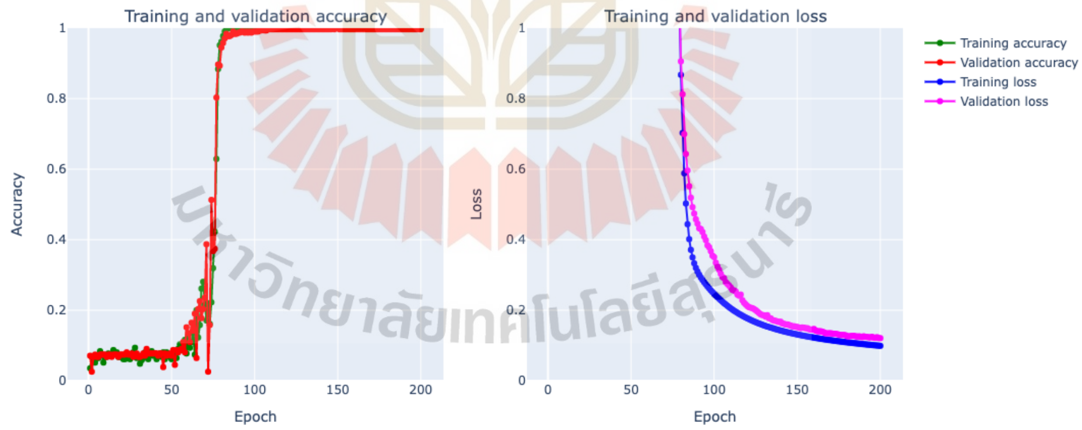


(b) the worst result from Adam optimizer

Figure 4.1 The best and worst from Adam optimizer.



(a) The best result from Adadelta optimizer.



(b) The worst result from Adadelta optimizer.

Figure 4.2 The best and worst from Adadelta optimizer.

CHAPTER V
DISCUSSION AND CONCLUSION

This chapter provides discussion and conclusion of the results appearing in this thesis.

5.1 Discussion

This section will discuss methods and results that somehow affect the following techniques or result in somehow.

First, the 310 ECG data from Physionet is in form of a file that can not be used directly in convolution neural networks and deep learning. We have to transfer the type of data to CSV type and use the Rdsamp package to read dat files, which means we did not use the data from the source or real-time data to calculate. But the ECG data that we transferred to CSV is not different from the original, and it can be used in the model that we build. The number of ECG data we have is not too much, but it can work on our scale. We have 310 ECG data from 90 volunteers, 20 data sets from some people, but only 2 or 3 data sets from others, and we get the best accuracy from every information set with 99 percent.

The convolution network or deep learning that we created is not a complex model for identifying or classifying the data. We use all of the data that we get and put it into the model, and we get the excellent output for identification data, but loss from the model is higher in the tail of the epoch that means the model is not the best for this data. We showed the output in the figure below.

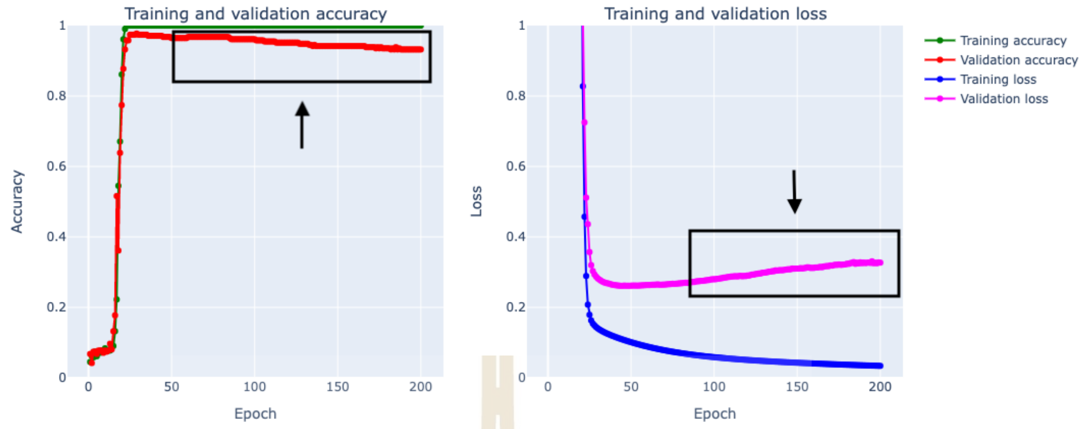


Figure 5.1 The results form the model with Adam optimizer. We get the accuracy from this optomizer is 95.25 percent and 0.24 loss.

From the upper figure 5.1, you can see the graph of accuracy and loss of the model with Adam optimizer. We get a loss value that increases in the tail of the graph's loss, and the accuracy is discrete at the end of the chart.

The problem is gone after changing the optimizer from the Adam optimizer to the Adadelta optimizer: the sagging of accuracy and the loss increase. We get better accuracy and loss from Adadelta, as shown in the figure 5.2 below.

From the result, we got 99 percent of the model and looked for 1 percent mistake. We found that from the 90 volunteers have just one or two errors. Table 5.1 shows the report of the model with Adam optimizer.

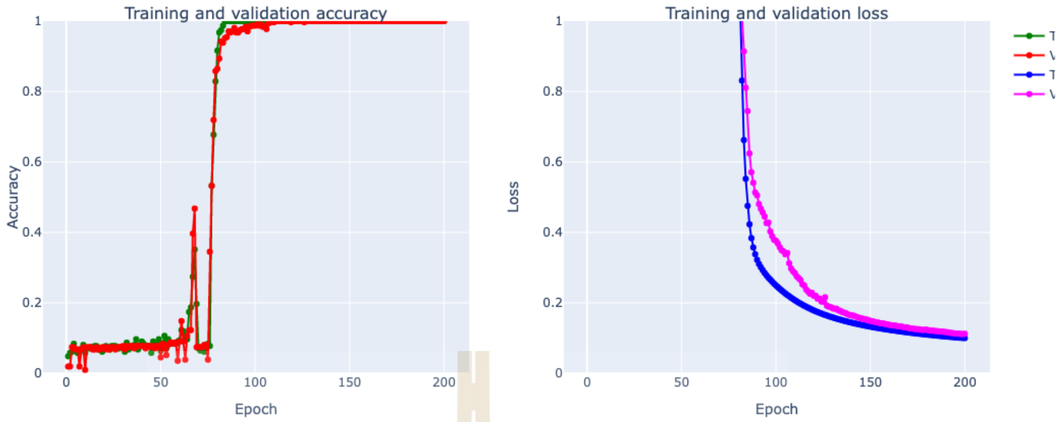


Figure 5.2 The results form the model with Adadelta optimizer. The accuracy from this optimizer is 99.54 percent and 0.12 loss.

Report of the model

The table shows a little mistake from the model with Adam optimizer. We have the ECG data for some data for 1, 2, or 3 data, and for any data that is 20 second ECG recording, so we use all of the data to put it on the model, but we get the correct output for 95 percent. And the table 5.2 below is a better report from the model with Adadelta optimizer, and we get better accuracy of 99 percent.

The report of the model by using Adadelta optimizer shows one or two fewer mistakes from any data than the Adam optimizer that we have employed in the previous process. This means that the Adadelta optimizer is suitable for multiple classification or identification.

Table 5.1 Report of the model with adam optimizer.

No.	Precision	Recall	f1-score	support
1	0.83	1.00	0.91	20
2	0.96	1.00	0.98	22
5	1.00	0.50	0.67	5
8	1.00	0.50	0.67	2
9	1.00	0.86	0.92	7
14	0.50	0.67	0.67	3
24	1.00	0.80	0.89	5
30	1.00	0.80	0.89	5
34	1.00	0.80	0.89	5
40	1.00	0.75	0.86	4
46	1.00	0.80	0.89	5
52	1.00	0.91	0.95	11
53	1.00	0.80	0.89	5
59	1.00	0.60	0.75	5
60	1.00	0.67	0.80	3
68	0.39	1.00	0.44	2
75	1.00	0.67	0.80	3
76	1.00	0.67	0.80	3
78	1.00	0.50	0.67	2
90	0.67	1.00	0.80	2
accuracy	-	-	0.94	310
macro avg	0.96	0.94	0.94	310
weighted avg	0.97	0.94	0.95	310

Table 5.2 Report of the model with adadelta optimizer.

No.	Precision	Recall	f1-score	support
1	1.00	1.00	1.00	20
2	1.00	1.00	1.00	22
3	1.00	1.00	1.00	5
...	...	...	...	...
76	1.00	0.67	0.80	3
...	...	...	...	...
89	1.00	1.00	1.00	2
90	0.67	1.00	0.80	2
accuracy	-	-	1.00	310
macro avg	1.00	1.00	1.00	310
weighted avg	1.00	1.00	1.00	310

From the result above, we have the model that can identify an individual by using ECG data with a 20-second recording with 99 percent accuracy and 0.12 loss. But the model is not developed for real-time identification, as we can not detect the ECG signal in a second. We use all of the data that we have in the model. If we can find the unique part of ECG that is different for any people and detect it in a minute, the model will be more practical for daily use. For the important thing in the future, the ECG can be a key to use for the security part, such as rocket launches or the big decision of leaders who have to make sure who can do or decide this critical thing.

5.2 Conclusion

Whereas biometrics is already being used for personal identification, we have made an effect to improve on it by means of ECG biometric. We have learn about the heart's anatomy, heart signals, heart rate, electrical in nature, and ECG (electrocardio-gram). We try to find techniques to detect the feature that can be biometric in deep learning, so the ECG signal from individuals is different. ECG signal is the first character that we use for identification. Otherwise, the ECG data is not public data that can generally work, so we have to learn about dat files and use it on convolution neural network and deep learning.

We have learned to code for convolution neural networks and deep learning to create the model in this project. Our model is similar to the classification model, but we have more than two classes (90 classes for 90 people), and the ECG data that we have is not of equal sile for every class. All of the 310 ECG data were denoised and used for the training and the original ECG data we used for the testing data set. If we can use ECG data from the heart to use in biometrics, the vital thing that we have to protect or need to let them save will be easy to do because we can use passwords from our heart that Can't find it easily.

Finally, we have known that ECG data can identify an individual, and we use a convolution neural network to work inside. The results show that ECG can be used in biometrics with good accuracy. At present, the drawback is that taking an ECG requires sophisticated equipment and can not be recorded in an instance. However, if a smart watch can be developed that records ECG data, then people will be able to be identified immediately by ECG, and need not use a password for security.

REFERENCES

REFERENCES

Barros, A., Resque, P., Almeida, J., Mota, R., Oliveira, H., Rosário, D., and Cerqueira, E. (2020). Data Improvement Model Based on ECG Biometric for User Authentication and Identification. *Sensors, 20*(10), Article number 1920. doi:10.3390/s20102920

Belo, D., Bento, N., Silva, H., Fred, A., Fred, A., and Gamboa, H. (2020) ECG Biometrics Using Deep Learning and Relative Score Threshold Classification. *Sensors, 20*(15), Article number 4078. doi:10.3390/s20154078

Biel, L., Pettersson, O., Philipson, L., and Wide, P. (2001). ECG analysis: a new approach in human identification. *in IEEE Transactions on Instrumentation and Measurement, 50*(3), pp. 808-812, June 2001, doi:10.1109/19.930458.

Deshmane, M., and Madhe, S. (2018). ECG Based Biometric Human Identification Using Convolutional Neural Network in Smart Health Applications. *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, 16-18 August 2018, 1-6, doi:10.1109/ICCUBEA.2018.8697579

D'Aloia, M., Longo, A., and Rizzi, M. (2019). Noisy ECG Signal Analysis for Automatic Peak Detection. *Information*. https://doi.org/10.3390/info10020035

Israel, S. A., Irvine, J. M., Cheng, A., Wiederhold, M. D., and Wiederhold, B. K. (2005). ECG to identify individuals. *Pattern Recogn, 38*(1) (January, 2005), 133–142. doi:10.1016/j.patcog.2004.05.014

Jen, T., Yuki H., Pang, Ivy, L., Shu, L., Adam, M., Ru, T., Ming Chen, U., and Rajendra Acharya. (2018). Application of stacked convolutional and long short-term memory network for accurate identification of CAD ECG signals. *Computers in Biology and Medicine*. https://doi.org/10.1016/j.compbiomed.2017.12.023.

Kim, B. H., and Pyun, J. Y. (2020). ECG Identification For Personal Authentication Using LSTM-Based Deep Recurrent Neural Networks. *Sensors*. https://doi.org/10.3390/s20113069

Li, H., and Boulanger P. (2020) A Survey of Heart Anomaly Detection Using Ambulatory Electrocardiogram (ECG). (Sensors (Basel)). doi: 10.3390/s20051461. PMID: 32155930; PMCID: PMC7085598.

Llamedo, M., and Martínez, J. P. (2011). Heartbeat Classification Using Feature Selection Driven by Database Generalization Criteria. *IEEE Transactions on Biomedical Engineering, 58*(3), 616-625, doi:10.1109/TBME.2010.2068048

Mario Merone, Paolo Soda, Mario Sansone, and Carlo Sansone. (2017). ECG databases for biometric systems: A systematic review. *Expert Systems with Applications. 67*, Pages 189-202, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2016.09.030.

Odinaka, I., Lai, P., Kaplan, A. D., O'Sullivan, J. A., Sirevaag, E. J., and Rohrbaugh, J. W. (2012). ECG Biometric Recognition: A Comparative Analysis. *IEEE Transactions on Information Forensics and Security, 7*(6), 1812-1824, Dec. 2012, doi:10.1109/TIFS.2012.2215324

Patro, K. K., and Kumar, P. R. (2017). Effective Feature Extraction of ECG for Biometric Application. *Procedia Computer Science, 115*, 296-306. doi:10.1016/j.procs.2017.09.138.

Singh, N., Ayub, S., and Saini, J. P. (2013). Design of Digital IIR Filter for Noise Reduction in ECG Signal. *2013 5th International Conference on Computational Intelligence and Communication Networks*. doi:10.1109/CICN.2013.45.

Wieclaw, L., Khoma, Y., Falat, P., Sabodashko, D., and Herasymenko, V, (2017). Biometrie identification from raw ECG signal using deep learning techniques. *IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*. doi: 10.1109/IDAACS.2017.8095063.

Xie, L., Li, Z., Zhou, Y., He, Y., and Zhu, J. (2020). Computational Diagnostic Techniques for Electrocardiogram Signal Analysis. *Sensors (Basel, Switzerland).* https://doi.org/10.3390/s20216318

APPENDICES

APPENDIX A

REPORT OF MODEL

Table A.1 Report of the model with adam optimizer.

Begin of Table				
No.	Precision	Recall	f1-score	support
1	0.83	1.00	0.91	20
2	0.96	1.00	0.98	22
3	1.00	1.00	1.00	5
4	1.00	1.00	1.00	2
5	1.00	0.50	0.67	2
6	1.00	1.00	1.00	2
7	1.00	1.00	1.00	2
8	1.00	0.50	0.67	2
9	1.00	0.86	0.92	7
10	1.00	1.00	1.00	5
11	1.00	1.00	1.00	3
12	1.00	1.00	1.00	2
13	1.00	1.00	1.00	2
14	0.50	0.67	0.67	3
15	0.29	1.00	0.44	2
16	1.00	1.00	1.00	3
17	1.00	1.00	1.00	2
18	1.00	1.00	1.00	2
19	1.00	1.00	1.00	2
20	1.00	1.00	1.00	2
21	1.00	1.00	1.00	3
22	1.00	1.00	1.00	2
23	1.00	1.00	1.00	2
24	1.00	0.80	0.89	5
25	1.00	1.00	1.00	5

Continuation of Table A.1				
No.	Precision	Recall	f1-score	support
26	1.00	1.00	1.00	4
27	1.00	1.00	1.00	3
28	1.00	1.00	1.00	5
29	1.00	1.00	1.00	2
30	1.00	0.80	0.89	5
31	1.00	1.00	1.00	2
32	1.00	1.00	1.00	6
33	1.00	1.00	1.00	2
34	1.00	0.80	0.89	5
35	1.00	1.00	1.00	5
36	1.00	1.00	1.00	5
37	1.00	1.00	1.00	2
38	1.00	1.00	1.00	2
39	1.00	1.00	1.00	2
40	1.00	0.75	0.86	4
41	1.00	1.00	1.00	2
42	1.00	1.00	1.00	4
43	1.00	1.00	1.00	2
44	1.00	1.00	1.00	2
45	1.00	1.00	1.00	2
46	1.00	0.80	0.89	5
47	1.00	1.00	1.00	2
48	1.00	1.00	1.00	2
49	1.00	1.00	1.00	2
50	1.00	1.00	1.00	2
51	1.00	1.00	1.00	4
52	1.00	0.91	0.95	11

Continuation of Table A.1				
No.	Precision	Recall	f1-score	support
53	1.00	0.80	0.89	5
54	1.00	1.00	1.00	2
55	1.00	1.00	1.00	2
56	1.00	1.00	1.00	2
57	1.00	1.00	1.00	3
58	1.00	1.00	1.00	2
59	1.00	0.60	0.75	5
60	1.00	0.67	0.80	3
61	1.00	1.00	1.00	4
62	1.00	1.00	1.00	3
63	1.00	1.00	1.00	6
64	1.00	1.00	1.00	3
65	1.00	1.00	1.00	2
66	1.00	1.00	1.00	2
67	1.00	1.00	1.00	3
68	0.39	1.00	0.44	2
69	1.00	1.00	1.00	2
70	1.00	1.00	1.00	3
71	1.00	1.00	1.00	5
72	1.00	1.00	1.00	8
73	1.00	1.00	1.00	2
74	0.00	0.00	0.00	1
75	1.00	0.67	0.80	3
76	1.00	0.67	0.80	3
77	1.00	1.00	1.00	3
78	1.00	0.50	0.67	2
79	1.00	1.00	1.00	2

Continuation of Table A.1				
No.	Precision	Recall	f1-score	support
80	1.00	1.00	1.00	2
81	1.00	1.00	1.00	2
82	1.00	1.00	1.00	2
83	1.00	1.00	1.00	2
84	1.00	1.00	1.00	2
85	1.00	1.00	1.00	3
86	1.00	1.00	1.00	2
87	1.00	1.00	1.00	2
88	1.00	1.00	1.00	3
89	1.00	1.00	1.00	2
90	0.67	1.00	0.80	2
accuracy	-	-	0.94	310
macro avg	0.96	0.94	0.94	310
weighted avg	0.97	0.94	0.95	310
End of Table				

Table A.2 Report of the model with adadelta optimizer.

Begin of table				
No.	Precision	Recall	f1-score	support
1	1.00	1.00	1.00	20
2	0.96	1.00	0.98	22
3	1.00	1.00	1.00	5
4	1.00	1.00	1.00	2
5	1.00	1.00	1.00	2
6	1.00	1.00	1.00	2
7	1.00	1.00	1.00	2

Continuation of Table A.2				
No.	Precision	Recall	f1-score	support
8	1.00	1.00	1.00	2
9	1.00	1.00	1.00	7
10	1.00	1.00	1.00	5
11	1.00	1.00	1.00	3
12	1.00	1.00	1.00	2
13	1.00	1.00	1.00	2
14	1.00	1.00	1.00	3
15	1.00	1.00	1.00	2
16	1.00	1.00	1.00	3
17	1.00	1.00	1.00	2
18	1.00	1.00	1.00	2
19	1.00	1.00	1.00	2
20	1.00	1.00	1.00	2
21	1.00	1.00	1.00	3
22	1.00	1.00	1.00	2
23	1.00	1.00	1.00	2
24	1.00	1.00	1.00	5
25	1.00	1.00	1.00	5
26	1.00	1.00	1.00	4
27	1.00	1.00	1.00	3
28	1.00	1.00	1.00	5
29	1.00	1.00	1.00	2
30	1.00	1.00	1.00	5
31	1.00	1.00	1.00	2
32	1.00	1.00	1.00	6
33	1.00	1.00	1.00	2
34	1.00	1.00	1.00	5

Continuation of Table A.2				
No.	Precision	Recall	f1-score	support
35	1.00	1.00	1.00	5
36	1.00	1.00	1.00	5
37	1.00	1.00	1.00	2
38	1.00	1.00	1.00	2
39	1.00	1.00	1.00	2
40	1.00	1.00	1.00	4
41	1.00	1.00	1.00	2
42	1.00	1.00	1.00	4
43	1.00	1.00	1.00	2
44	1.00	1.00	1.00	2
45	1.00	1.00	1.00	2
46	1.00	0.80	0.89	5
47	1.00	1.00	1.00	2
48	1.00	1.00	1.00	2
49	1.00	1.00	1.00	2
50	1.00	1.00	1.00	2
51	1.00	1.00	1.00	4
52	1.00	1.00	1.00	11
53	1.00	1.00	1.00	5
54	1.00	1.00	1.00	2
55	1.00	1.00	1.00	2
56	1.00	1.00	1.00	2
57	1.00	1.00	1.00	3
58	1.00	1.00	1.00	2
59	1.00	1.00	1.00	5
60	1.00	1.00	1.00	3
61	1.00	1.00	1.00	4

Continuation of Table A.2				
No.	Precision	Recall	f1-score	support
62	1.00	1.00	1.00	3
63	1.00	1.00	1.00	6
64	1.00	1.00	1.00	3
65	1.00	1.00	1.00	2
66	1.00	1.00	1.00	2
67	1.00	1.00	1.00	3
68	1.00	1.00	1.00	2
69	1.00	1.00	1.00	2
70	1.00	1.00	1.00	3
71	1.00	1.00	1.00	5
72	1.00	1.00	1.00	8
73	1.00	1.00	1.00	2
74	1.00	1.00	1.00	1
75	1.00	1.00	1.00	3
76	1.00	1.00	1.00	3
77	1.00	1.00	1.00	3
78	1.00	1.00	1.00	2
79	1.00	1.00	1.00	2
80	1.00	1.00	1.00	2
81	1.00	1.00	1.00	2
82	1.00	1.00	1.00	2
83	1.00	1.00	1.00	2
84	1.00	1.00	1.00	2
85	1.00	1.00	1.00	3
86	1.00	1.00	1.00	2
87	1.00	1.00	1.00	2
88	1.00	1.00	1.00	3

Continuation of Table A.2				
No.	**Precision**	**Recall**	**f1-score**	**support**
89	1.00	1.00	1.00	2
90	1.00	1.00	1.00	2
accuracy	-	-	0.94	310
macro avg	0.96	0.94	0.94	310
weighted avg	0.97	0.94	0.95	310
End of Table				

APPENDIX B

CODE OF MODEL

```
from wfdb import rdsamp
from sklearn.model_selection import train_test_split
from itertools import combinations, zip_longest, islice  #izip
import matplotlib.pyplot as plt
from scipy import signal
from numpy import Inf
import pandas as pd
import numpy as np
import wfdb, math
import itertools
import os, sys
import wfdb.processing
```

Figure B.1 Code for importing library for the prepossessing work.

```
#.dat to .csv converter
class csvGenerator:
  def __init__(self):
    self.dir = os.path.join(os.getcwd(),'/content/drive/ecg-id-database-1.0.0')
    self.database = 'ecgiddb'

  def constructor(self, folder, filename):
    signals, fields = wfdb.rdsamp(filename, sampfrom=0,
                         pn_dir=os.path.join(self.database, folder))
    df = pd.DataFrame(signals)
    df.to_csv(os.path.join(self.dir, folder, filename + "." 'csv'),index=False)

  #crawls into every folder and sends .dat file to constructor
  def tocsv(self):
    for folders in os.listdir(self.dir):
      if (folders.startswith('Person_')):
        for inpersonsdir in os.listdir(os.path.join(self.dir, folders)):
          if (inpersonsdir.endswith('dat')):
            basename = inpersonsdir.split(".",1)[0]
            self.constructor(folders, basename)
```

Figure B.2 Code for converting DAT files to CSV files.

```
#generates features and labels
class ProcessData:
  def __init__(self):
    self.dir = os.path.join(os.getcwd(), '/content/drive/ecg-id-database-1.0.0')
    self.persons_labels = [] #who the person is
    self.date_labels = []    #month.day.year of ecg record
    self.ecg_filsignal = pd.DataFrame() #filtered ecg dataset
    self.ecg_signal = pd.DataFrame()  #unfiltered ecg dataset


#appends to a bigger global array
  def dumpfeats(self, array, flag):
    fil_df = pd.DataFrame(array)
    fil_df = fil_df.T
    ufil_df = pd.DataFrame(array)
    ufil_df = ufil_df.T
    if (flag == 1):
      self.ecg_filsignal = self.ecg_filsignal.append(fil_df, ignore_index=True)
    if (flag == 2):
      self.ecg_signal = self.ecg_signal.append(ufil_df, ignore_index=True)
```

Figure B.3 Code for generating features and labels.

```
%tensorflow_version 1.x
import tensorflow
tensorflow.__version__
from keras.layers import Dense, Dropout, Activation,
  Flatten, Convolution2D, MaxPooling2D
from sklearn.metrics import classification_report,confusion_matrix
from keras.callbacks import EarlyStopping, TensorBoard
from keras.models import Sequential, load_model
from keras import optimizers, regularizers
from keras.utils import np_utils
from keras.optimizers import SGD

from sklearn.model_selection import train_test_split

import numpy as np
from time import time
import os

import io
import pandas as pd
```

Figure B.4 Code for importing library for the model creation.

```
train = pd.read_csv('/content/drive/filecgdata.csv')
test = pd.read_csv('/content/drive/unfilecgdata.csv')

X_train = X_train.reshape(X_train.shape[0], 1, 9999, 1)

# preprocess data
X_train = X_train.reshape(X_train.shape[0], 1, 9999, 1)
X_test = X_test.reshape(X_test.shape[0], 1, 9999, 1)
print (X_train.shape)
print (X_test.shape)

# normalize data values to range [0, 1]
X_train /= 255
X_test /= 255

# convert flat array to [Person1 .. Person90] one-hot coded array
Y_train = Y_train - 1
Y_test = Y_test - 1
Y_train = np_utils.to_categorical(Y_train, 90)
Y_test = np_utils.to_categorical(Y_test, 90)
```

Figure B.5 Code for reshaping size of training and test sets.

```
#model architecture
model = Sequential()
model.add(Convolution2D(32, 1, 5, activation='tanh',
                        input_shape=(1,9999,1),
                        kernel_regularizer=regularizers.l2(0.001)))
model.add(MaxPooling2D(pool_size=(1,3)))

model.add(Convolution2D(64, 1, 5, activation='tanh',
                        kernel_regularizer=regularizers.l2(0.001)))
model.add(MaxPooling2D(pool_size=(1,3)))

model.add(Flatten())
model.add(Dense(128, activation='tanh'))
model.add(Dense(90, activation='softmax'))

# compile model
model.compile(loss='categorical_crossentropy',
              optimizer='adadelta',
              metrics=['accuracy'])
```

Figure B.6 Code for model creation.

```
# fit model on training data
tensorboard = TensorBoard(log_dir="logs_personid/{}".format(time()))
earlystopping = EarlyStopping(monitor='val_loss', patience=499)
history = model.fit(X_train, Y_train, batch_size=30,
                    validation_data=(X_test, Y_test), nb_epoch=200,
                    verbose=1, callbacks = [earlystopping, tensorboard])

# evaluate model on test data
print ("Evaluating model")
score = model.evaluate(X_test, Y_test, verbose=1)
print('Test loss:', score[0])
print('Test accuracy:', score[1])
```

Figure B.7 Code for training and model display.

CURRICULUM VITAE

NAME : Thitayawat Khumwong **GENDER :** Male

EDUCATION BACKGROUND:

- Bachelor of Science (Mathematics), Honors Program (Second class honors), Suranaree University of Technology, Thailand, 2019

SCHOLARSHIP:

- Development and Promotion of Science and Technology Talents Project Scholarship

CONFERENCE:

- Khumwong, T., and Tanthanuch, J. (2021) The Study of ECG for Biometrics Identification by Convolution Neural Network., **The 4th National Conference on Science and Technology: NCST 4th 2021**, Chandrakasem Rajabaht University, Bangkok, May 22rd, 2021 (336-341)

  Award: Best paper for research article

EXPERIENCE:

- Teaching assistant in Suranaree University of Technology, Calculus I (Thai course) Term 1/2020

- Teaching assistant in Suranaree University of Technology, Calculus II (Thai&International course) Term 2/2020

- Work & Travel, Yellow Stone, Montana, USA (June-September, 2021)