COMPARING HEART RATE VARIABILITY AND PULSE RATE VARIABILITY FOR APNEA CLASSIFICATION BY USING DEEP LEARNING

BY

NID SUREECHAINIRUN

A PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF BACHELOR OF ENGINEERING IN BIOMEDICAL ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2023

SCHOOL OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

PROJECT CERTIFICATE

Project Title Comparing Heart Rate Variability and Pulse Rate Variability for Apnea Classification by Using Deep Learning

Student Name Miss Nid Sureechainirun Student ID. 63011211

Degree Bachelor of Engineering in Biomedical Engineering

Project Advisor Signed: _____

(Prof. Dr. Chuchart Pintavirooj)

Project Advisor Signed: _____

(Asst. Prof. Dr. Veerayuth Kittichai)

Project Advisor Signed: Ukkrit Jansri

(Aj. Ukkrit Jansri)

Committee Signed: _____

(Prof. Dr. Chuchart Pintavirooj)

Committee Signed: _____

(Asst. Prof. Dr. Treesukon Treebupachatsakul)

Committee Signed: _____

(Asst. Prof. Dr. Kasama Srirussamee)

Head of Department Signed: _____

(Assoc. Prof. Dr. SarinpornVisitsattapongse)

Project Title	Comparing Heart Rate Variability and Pulse Rate Variability for Apnea Classification by Using Deep Learning
Student Name	Miss Nid Sureechainirun
Degree	Bachelor of Engineering in Biomedical Engineering
Project Advisor	Prof. Dr. Chuchart Pintavirooj
	Asst. Prof. Dr. Veerayuth Kittichai
	Aj. Ukkrit Jansri
Academic Years	2023

Abstract

Many studies have been using different signals for apnea detection or classification as an alternative to polysomnography. Electrocardiogram or ECG signal is one of the most common signals that is reliable for apnea classification. The important features extracted from ECG called heart rate variability (HRV) is used for deep learning to find the significant patterns for apnea classification. Another signal that has some relationship with ECG is called plethysmography or PPG signals. It can be used to find pulse rate variability (PRV). Deep learning will be used to compare the efficiency of apnea classification between HRV and PRV.

This study pre-processes 115 datasets from Multi-Ethnic Study of Atherosclerosis to be trained in ResNet-50 in MATLAB. Those datasets contain 794.6167 hours of normal and apnea ECG and PPG signal. The process will start with noise removal and peak detection. Then HRV and PRV is calculated in time-domain before being segmented into a 60 seconds period and classified into normal and apnea parts based on the apnea labels. The input will be converted into images with HRV or PRV.

The validation accuracy of HRV and PRV is 90.27% and 86.69% respectively. The highest accuracy and specificity of HRV from the test result is 79.17% and 81.69%, respectively. And the highest accuracy and specificity of PRV from the test result is 84.55% and 87.85% respectively. However, the sensitivity of both signals is very low. Every position in the signal will be predicted whether it is normal or apnea.

In conclusion, the algorithm of apnea classification in this study still needs to be further improved. The number of datasets and training images should be increased. And datasets from other sources should be used to validate the result.

Acknowledgements

I would like to thank Aj. Ukkrit Jansri for being my project advisor and providing the dataset from MESA Sleep Study and Prof. Dr. Chuchart Pintavirooj for recommending the deep learning model. This project was accomplished with the kindness of my project advisor. In addition, I would like to thank Ned Sureechainirun for the R Peak Detection algorithm and code in MATLAB. With the suggestions and corrections throughout this project, I can reach the objective of the study. Moreover, I appreciate the department of Biomedical Engineering Department, Faculty of Engineering for providing the MATLAB license and budget to complete the project.

Furthermore, I would like to give credit to the Sleep-data Organization that provided me with a dataset of patients with apnea annotation by polysomnography test. Finally, I would like to give thanks to every person who is not mentioned in this thesis but was involved in helping and supporting me.

Miss Nid Sureechainirun

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS/ABBREVIATIONS

Symbols/Abbreviations	Terms
ECG	Electrocardiogram
PPG	Plethysmography
HRV	Heart Rate Variability
PRV	Pulse Rate Variability
OSA	Obstructive Sleep Apnea
CSA	Central Sleep Apnea
CompSA	Complex Sleep Apnea
Apnea	Sleep Apnea and Hypopnea

Chapter 1

Introduction

1.1 **Statement of the problems**

Sleep apnea and hypopnea is a sleep disorder that interrupts the breathing of the patients while sleeping (1). It is a dangerous condition that impacts patients' quality of life and may lead to serious health problems, especially when it is left untreated (2, 3). The prevalence of obstructive sleep apnea is around 2 to 7% (4), which widely affects almost a billion of populations around the world (5), depending on age, sex, obesity, health behaviors, and other factors . Patients will have poor quality sleeping which leads to fatigue and day-time sleepiness (4). However, the patients might not have noticed their condition or some clinical signs, such as snoring, that can be observed by other people only.

It is important for apnea patients to be diagnosed to prevent or treat it as soon as possible. The conventional gold standard test for apnea is called polysomnography (PSG) that requires various signals that shows the activity of specific organs such as electrocardiogram (ECG for heart), electromyogram (EMG for muscle), Electrooculography (EOG for eyes movement), and electroencephalogram (EEG for brain) (6). Although the result from PSG is accurate, it wastes time and money. Most apnea patients remain undiagnosed that may be due to inaccessibility of PSG (7). Therefore, other non-invasive and lower-cost methods have been developed to detect sleep apnea and hypopnea.

Many studies indicated that apnea can be detected by using only one type of signal or sensors such as ECG, nasal airflow, or SpO2 (8). It is to reduce the cost of the diagnostic test as much as possible. Most studies use ECG signals to monitor sleep apnea and/or hypopnea events. And one of the signals is single lead ECG that can be used to measure the heart's electrical activity by attaching only one electrode (9). The feature called heart rate variability (HRV) is then extracted to represent the variation of heart rate in the form of time in milliseconds between RR intervals of ECG (10).

There is a relationship between HRV, oxygen concentration and apnea, so many studies have been using the HRV feature that is derived from ECG for apnea screening. HRV can be used to diagnose various diseases that relate to the heart function (10) that is controlled by autonomic nervous system (ANS), including parasympathetic nervous system and sympathetic nervous system (11). In stressful situations, sympathetic NS induces heart rate to increase, so the HRV will decrease (12). Similar to the condition when a patient lacks oxygen, the body will be more stressed, and the heart will beat faster to pump blood to provide oxygen to tissue (13). Conversely, high HRV is during parasympathetic NS. And people with high HRV tend to have less stress and are healthier (12). Therefore, HRV can be used as a biomarker for sleep apnea, and can be used for apnea diagnostic or screening tests.

Although the accuracy of detection is up to 97%, it cannot reach the gold standard of PSG (9). The devices are also too expensive and complex to be performed by patients themselves. On the other hand, as the oxygen concentration relates to the condition of sleep apnea and hypopnea, another signal that can be used as the alternative is called

photoplethysmography (PPG). It is typically used to measure the rate of blood volume in the measuring contact area such as fingertips or ear lobe (14). It can demonstrate two main features: the heart rate and blood oxygen saturation (SpO2). Some studies propose pulse rate variability (PRV) that is quite similar to HRV (15), and it can be used for sleep apnea monitoring as well (16). The performance of classification might increase by using both PRV and SpO2. The benefit is that PPG is easy to be measured by the wearable devices.

The relationship between HRV, oxygen concentration and sleep apnea is not that simple, so deep learning is used to detect that specific pattern. Deep learning is a useful tool for recognizing the patterns of large amounts of data. So that it can be used to classify or detect sleep apnea by HRV or PRV. There are many types of neural networks such as convolutional neural networks (CNN), Long Short-Term Memory (LSTM), and Deep Belief Network (DBN) (9). They are recommended for training time-series signals. The combination of CNN and LSTM is widely used in several studies.

Moreover, although many types of signals are used in apnea screening, the conclusion whether which signal is the best cannot be answered completely (8). According to the meta-analysis, different studies used different datasets, method of data preparation, and model, so the efficiency of signal should not be compared between studies (8). One way to solve this problem is to compare many types of signal within a study.

In conclusion, this study will focus on converting ECG and PPG signals into HRV and PRV respectively before training in the CNN and LSTM model. The network should be able to

differentiate between the period with apnea and normal breathing. In addition, the accuracy, sensitivity, and specificity of classification by using HRV and PRV will be compared. Therefore, this study will compare HRV and PRV to see the performance of apnea classification along with developing the algorithm and deep learning model of classification.

1.2 **Objectives**

1.2.1 To detect sleep apnea and hypopnea duration of the patients by using HRV and PRV by using deep learning ResNet-50 model.

1.2.2 To compare the detection efficiency including accuracy, sensitivity, and specificity between using HRV and using PRV.

1.3 **Scope**

1.3.1 Calculate HRV and PRV from ECG signal and PPG signal, respectively.

1.3.2 Use deep learning to detect sleep apnea and hypopnea.

1.3.3 Use statistical methods to compare the performance between HRV and PRV within the study.

1.4 **Expected Benefits**

1.4.1 Be able to detect apnea by using ECG or PPG for more than 90% accuracy.

1.4.2 Be able to answer whether HRV or PRV is better for apnea classification

Chapter 2

Review of Theories and Principles

This chapter will explain the definition of related vocabulary and all materials that were used in this paper.

2.1 Sleep Apnea

Sleep apnea is a condition when a patient stops breathing while asleep for more than 10 seconds (17). It usually lasts for 10 to 20 seconds before the patient starts to breathe again. The frequency of sleep apnea will depend on the individual, and it can occur repeatedly. According to the National Heart, Lung, and Blood Center, the patient's breath may pause 5 to 30 times or more within an hour. It will lower the oxygen concentration in blood and tissues (hypoxia) because of lacking proper breath (18). Hypoxia will activate the sympathetic nervous system which leads to the release of epinephrine and norepinephrine (11,12). Consequently, heart rate and blood pressure will increase to provide more oxygen to body tissue. This will negatively impact the quality of sleep and can also be a risk factor of cardiovascular diseases such as hypertension and stroke (1).

There are three main types of sleep apnea including obstructive sleep apnea (OSA), central sleep apnea (CSA), and complex sleep apnea (CompSAS). This study will focus more on OSA, which is the most common type and is found in the dataset more than other two types.

OSA occurs when the airway is blocked either partially or completely. It usually occurs in men more than women (13). And it can be due to obesity, anatomy of the throat, or smoking that lead to a narrow airway. Most people have OSA because of excess weight because of the soft tissue that relaxes and blocks the airway during sleep as shown in figure 1 (19).

CSA is another type of sleep apnea, but the cause of airway blockage is different from OSA. OSA is caused by physical blockage of the airway while CSA happens because the brain fails to send signals to the muscles that are responsible for controlling breathing (20). The last type of sleep apnea is called mixed or complex sleep apnea (CompSAS). It is the combination of OSA and CSA for more than five hours. It develops when a patient with OSA starts using continuous positive airway pressure (CPAP). It might because the therapy affects the central respiratory control, and it then causes CSA (21). Even though there are many types of sleep apnea, the negative consequences are similar and should be diagnosed for treatment.

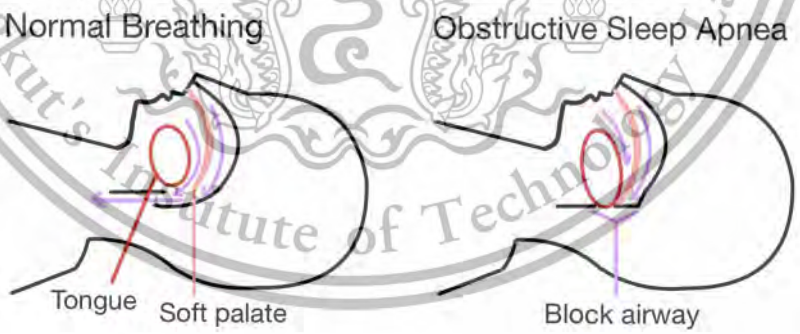


Figure 1 Difference between normal and OSA

The severity of sleep apnea can be classified into normal, mild, moderate, and severe sleep apnea by using the apnea-hypopnea index AHI that represents the number of apnea per hour during sleep (22). The American Academy of Sleep Medicine set the criteria for mild, moderate, and severe for 5-15 times/hour, 15-30 times/hour, and more than 30 times/hours, respectively.

From the literally based analysis of Benjafield et. al., there are almost 1 billion people from all over the world who have OSA (5). According to the systematic review of Senaratna et. al., the prevalence of OSA in adults is 9-38%, depending on different studies(23). The number of prevalence will be high in some subgroups such as elderly and male population up to 90% (5). In Thailand, according to HRH Princess Maha Chakri Sirinthorn Medical Center, 11.4% is reported as a prevalence of OSA (24). From the study of Finkel et. al., in US adults, about 90% of sleep apnea patients remain undiagnosed (3). It can be because of not being aware of or inaccessible for the diagnosis test (7).

2.2 Hypopnea

Hypopnea is another condition that is similar to sleep apnea though it is less severe. According to the American Academy of Sleep Medicine, hypopnea patients will have slow or shallow breath during sleep due to reduction of airflow for at least 30% (25). However, there is another criteria that states at least 50% of airflow reduction (26). This condition will maintain for at least 10 seconds to be considered as hypopnea, so the concentration of oxygen in blood will

decrease as well as sleep apnea (25). The oxygen saturation might be reduced by 3 to 4% or more depending on the definition given by a specific organization (26).

Patients with hypopnea usually have a chance of developing sleep apnea or they might occur together (26, 27). Obstructive Sleep Apnea-Hypopnea syndrome is a sleeping disorder that combines both OSA and hypopnea by the alternating and repeating of two events. It will result in disrupting sleep patterns and inadequate airflow (26). The prevalence of this condition has been increasing over the past fifty years (26). It can lead to daytime sleepiness and tiredness all day. When the patient wakes up, they might be choking and tired. They will also usually wake up and snore during the night (27). This condition can also cause other cardiovascular diseases, cerebrovascular diseases, or metabolic syndrome that are causes leading to death (26). Therefore, the patient must undergo treatment as soon as possible.

2.3 Polysomnography

Polysomnography is a gold standard for diagnosis of sleep disorders, such as sleep apnea and hypopnea (28). PSG needs monitoring of several signals that show the activity of specific organs such as heart, muscle, eyes, and brain while the patient is sleeping. PSG also records sleep stages(1, 2, 3, 4, and rapid eye movement- REM), movements of limb, body position, and oxygen saturation as shown in figure 2.

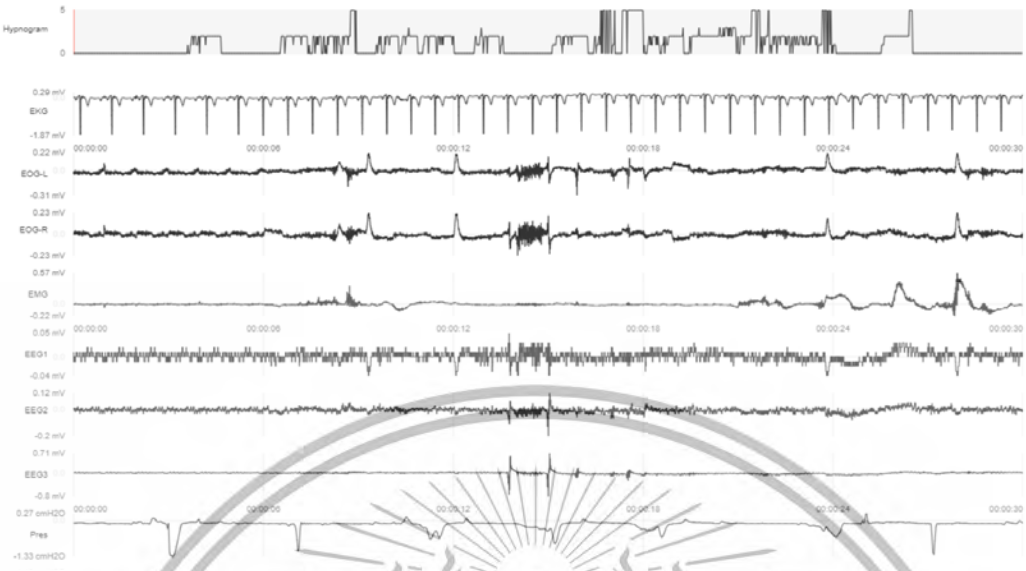


Figure 2 Polysomnography test

PSG requires various signals including electrocardiogram (ECG), electromyogram (EMG), Electrooculography (EOG), and electroencephalogram (EEG) (6). EMG electrodes are placed on the muscles at the chin or legs to monitor the muscle activity and movement during sleep. It is to identify the REM stage and periodic limb movements of sleep (PLMS) or sleep-related movement disorder (29). EOG is to record the eyes movement activity by placing electrodes near the eyes to detect REM sleep stage that shows normal sleep (30). EEG is to measure the activity of the brain. The EEG electrode is placed on the scalp of the patient to assess different stages of sleep such as wakefulness, light sleep, deep sleep, and REM and identify the abnormal brain wave pattern (31).

Moreover, PSG also needs to record other respiratory parameters such as nasal airflow, abdominal and thoracic movement, and pulse oximetry (6). They are used to detect abnormal breathing patterns. Nasal airflow is to measure the pressure of airflow through the nasal cavity by

using the pressure sensor (30). It is to detect reduction of airflow to further classify types of apnea: hypopnea, OSA, CSA, or CompSA (30). And the abdominal sensor is to assess the effort of breathing (30). There are also other records in the form of sound or video to observe physical movement, position, and snoring of the patients. Other parameters such as ECG and pulse oximetry will be explained in the next section.

PSG is a non-invasive test, so it is safe for patients. However, it takes at least a night to perform at the hospital (28). The polysomnography test is also inconvenient because the participants must avoid caffeine or alcohol in the afternoon before PSG. It is because that beverage can impact the result of sleep patterns (33).

2.4 Electrocardiogram

Electrocardiogram (ECG) is a test for recording the heart's electrical activity. It can be a diagnostic tool for cardiovascular disease that is non-invasive. To measure ECG, it needs an ECG machine and electrodes. The ECG machine is used to record, control, analyze, and display ECG signals. ECG electrodes are small adhesive patches made of conductive materials and wire to connect to the main machine (34).

There are many ways to measure ECG by attaching different electrodes to specific places such as twelve-lead ECG, five-lead ECG, three-lead ECG, and single-lead ECG. The potential difference between each of every two electrodes will be measured as shown in figure 3 that is the example of three-lead ECG.

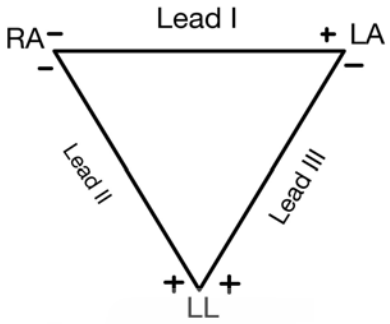


Figure 3 Voltage in Three-lead ECG

First, twelves-lead ECG that places 10 electrodes at the limb and chest. The limb lead consists of the right arm, left arm, right leg and left leg while the chest lead includes V1, V2, V3, V4, V5, and V6 as shown in the figure 4 (35). V1 is the fourth intercostal space on the right sternal edge while V2 is on the left. V4 is at the fifth intercostal space at the midclavicular line. And V3 is between the V2 and V4. V6 is at the midaxillary line beside V4 while V5 is located between the V4 and V6 (35). Three-lead ECG uses only three electrodes to attach on the right arm, left arm and left leg. And a single-lead ECG that measures the small voltage difference between only two electrodes on the left and right arm.

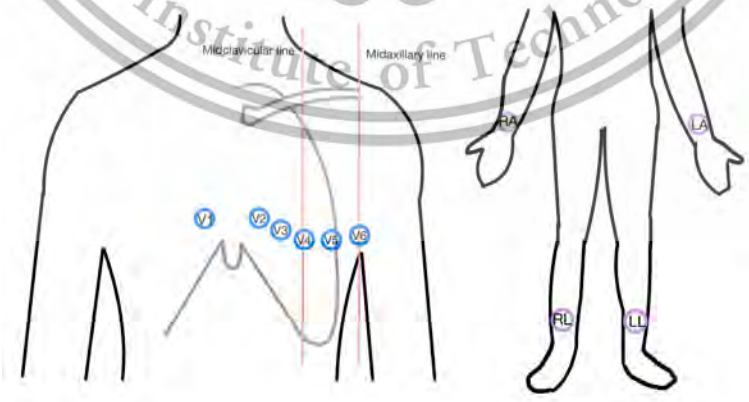


Figure 4 Position of Electrodes in 12-lead ECG

There are three main components including P wave, QRS complex, and T wave as in the figure 5. ECG can be used to determine heart rate, rhythm, and some abnormalities such as cardiac hypertrophy by using the amplitude and duration of the components. First, the P wave is the electrical potentials produced by atrial depolarization or before the atrium of the heart starts to contract. Then the QRS complex is the potential generated when the heart ventricle depolarizes before contraction. After that T wave is generated when the ventricles begin the repolarization state which means that it recovers from depolarization (36).

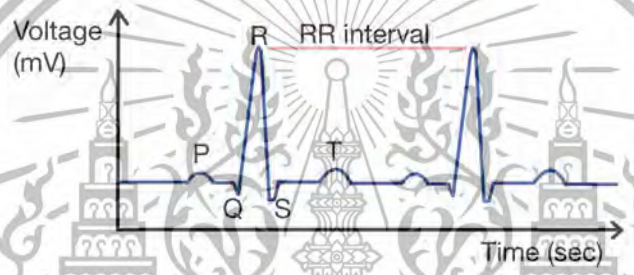


Figure 5 Electrocardiogram

The interval between P to T waves in each cycle is 0.8 seconds (37). Normally, one cycle of ECG or the duration between RR intervals is around 0.6 to 1.2 seconds (38). This study will mainly focus on R peak and RR intervals that are the duration of time between each peak. There are many ways of R peak detection, but the easy way is to segment the signal and find the peak in each segment.

2.5 Heart Rate Variability

Heart Rate Variability (HRV) is the variation in time between RR intervals of ECG (10). It can be used to diagnose various diseases that relate to the heart function that is controlled by

autonomic nervous system (ANS), including parasympathetic nervous system and sympathetic nervous system (11). In stressful situations, the heart rate will increase because of sympathetic NS. It will affect HRV to decrease since the range between RR intervals is shortened (39). Similarly, when a patient lacks oxygen, the body will be more stressed, and the heart will beat faster to pump blood to provide oxygen to tissue (13). Therefore, HRV will be lower than normal.

However, parasympathetic NS is working when humans are in the relaxation state such as sleeping or sitting calmly (40). The hormone called acetylcholine is released to the bloodstream to reduce the heart rate (40). As when the heart beats slower, HRV will be higher compared to the previous state of the same individual. People with high HRV tend to have less stress and are healthier (32). Consequently, HRV can be used as a biomarker for sleep apnea, and can be used for apnea diagnostic or screening tests.

There might be other factors that influence HRV, such as age, gender, and emotion. When the age increases, HRV will generally decrease due to the decline of ANS. However, children might have lower HRV compared to young adults because they have a higher heart rate. (41) And for genders, males tend to have higher HRV than females. Emotion that influences the heart rate also affects heart rate and HRV. For example, anxiety, stress, or angriness can decrease HRV (42). HRV while asleep and awake is also different in the same person.

There are two main methods to calculate HRV: time-domain and frequency-domain (43). Time-domain HRV can be calculated by using the variation of difference in time in microseconds

between R to R peaks in ECG. There are many ways to calculate the variation, such as standard deviation and root mean square of differences. This study will mainly focus on the square root of the mean of the squares of successive differences between RR intervals (RMSSD). The HRV formula (1) is shown below.

$$RMSSD = \sqrt{\frac{\sum_{i=1}^{N-1} (RR_i - RR_{i+1})^2}{N-1}} \quad \text{-------(1)}$$

For the frequency domain, the easy way is to convert the time domain by a fast fourier transform. There is a built-in function in MATLAB called fft to convert time-domain into the frequency-domain (44). Then the array of HRV can be converted into images by spectrogram. Spectrogram is a two-dimensional graph between time and frequency that presents the third dimension in a form of color (45).

2.6 Plethysmography

Plethysmography is a low-cost and non-invasive method to measure volume of blood change in the body part, such as the finger and earlobe. One type of plethysmography is called photoplethysmography (PPG) (14). PPG is commonly used to measure oxygen saturation and heart rate in wearable devices (14). PPG can be measured by using light sources and a photodetector on the skin surface as shown in the figure 6 (46). There are two light sources: red light with about 660 nm that is in a range of visible light and near infrared with 940 nm that is

invisible light. Both of them will pass through the skin and arteries under the skin. Some of the light will be absorbed by the skin and hemoglobin, so there will be less light that is emitted and gets detected by the photodetector. According to the graph in the figure 6, oxyhemoglobin is more likely to absorb near-infrared light while deoxyhemoglobin absorbs more red light, resulting in lower absorbance of infrared light with 940 nm (47).

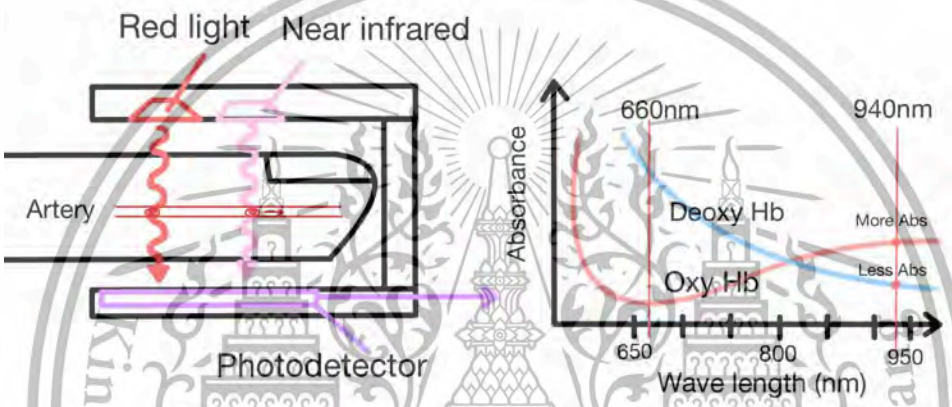


Figure 6 Photoplethymography

2.7 Peripheral Oxygen Saturation

Another feature found in PPG is called SpO2 that can be estimated by analyzing the changes in light absorption due to the amount of oxygenated and deoxygenated blood. Both can be the parameters to increase accuracy of sleep apnea detection, so using PPG might be more convenient and as accurate as using ECG.

Peripheral Oxygen Saturation (SpO2) shows the number of hemoglobin that carry oxygen. Decreasing in oxygen level in blood because of sleep apnea or hypopnea can affect oxygen desaturation. Normal people usually have 93-98% oxygen saturation in blood.

Oxygen saturation (SpO2) can be calculated based on the Beer-Lambert Law and the following equation in the figure 7 (48). In the formula, α is the absorption coefficient that is a constant. αo is for oxygenated hemoglobin while αr is for hemoglobin at a specific wavelength. Wavelength 1 is for the red light while wavelength 2 is for the infrared light. Co is the concentration of oxyhemoglobin, whereas Cr is concentration of hemoglobin. Therefore, SpO2 can be calculated when the concentration of hemoglobin is known or when R value is known.

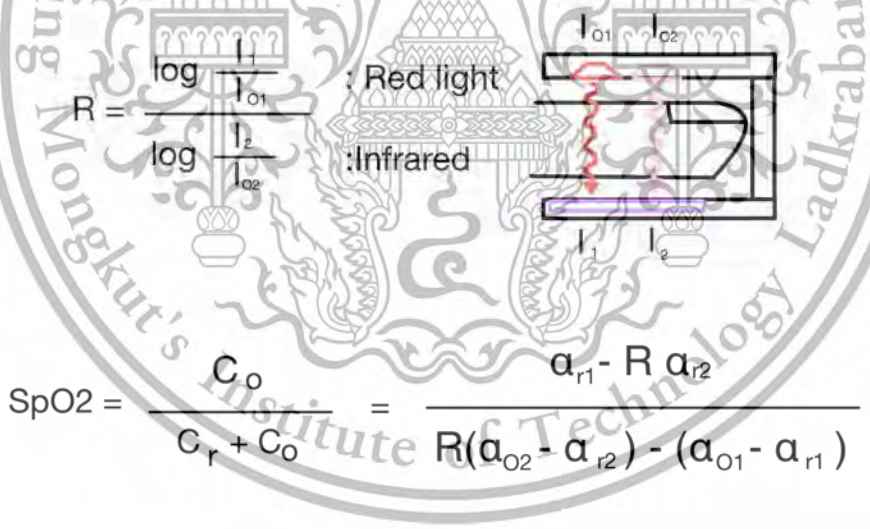


Figure 7 Oxygen Saturation Formula

Pulse transit time (PTT) is the duration when blood travels from the heart to a peripheral organ such as the finger. When ECG is compared to the PPG, the duration between the R peak that represents ventricular contraction and the plethysmography peak when oxygenated blood

reaches the fingertips. is called PTT. The flowrate of blood can be calculated by using time and distance. It can be used to identify heart rate (49) as the peak of PPG associated with ECG.

2.8 Pulse Rate Variability

Pulse Rate Variability (PRV) is similar to HRV in that it uses the peak of each cycle of PPG instead of R peak of ECG. It is used to assess autonomic nervous system activity. Peak of PPG will be detected by using the same method of detecting R peak in ECG by using the for loop in MATLAB. Then the PRV formula (2) of RMSSD will be used.

$$RMSSD = \sqrt{\frac{\sum_{i=1}^{N-1} (RR_i - RR_{i+1})^2}{N - 1}} \quad \text{-------(2)}$$

2.9 MATLAB

MATLAB is a program that is widely used in scientific and engineering applications. MATLAB stands for "MATrix LABoratory" and was originally developed by MathWorks (50). It has its own programming language. It allows users to perform numerical computation, data analysis, visualization, and algorithm development. It offers a broad range of built-in functions and toolboxes for various domains, including mathematics, statistics, signal processing, image processing, control systems, and more.

There are many key features in MATLAB (48):

1. Matrix Operations: MATLAB's core strength lies in its ability to perform matrix and array operations efficiently. It provides a comprehensive set of functions for linear algebra, numerical computations, and solving systems of equations.

2. Data Analysis and Visualization: MATLAB offers powerful tools for data analysis, manipulation, and visualization. It provides functions for statistical analysis, curve fitting, interpolation, filtering, and data visualization, allowing users to explore and understand their data easily.

3. Algorithm Development: MATLAB enables users to develop and implement algorithms quickly. It provides an extensive set of built-in functions, libraries, and tools for algorithm development, simulation, and prototyping.

4. Plotting and Graphics: MATLAB provides a wide range of plotting and visualization functions to create 2D and 3D plots, graphs, and custom visualizations. It offers extensive options for customizing plots and generating publication-quality graphics.

5. Toolboxes and Add-Ons: MATLAB offers various toolboxes and add-ons that provide specialized functionality for specific applications, such as image processing, control systems design, optimization, machine learning, and more. These toolboxes expand the

capabilities of MATLAB and provide ready-to-use functions and algorithms for specific domains.

There are many other features of MATLAB. The main toolbox that will be used in this study is called “Deep learning toolbox”.

2.10 Deep Learning

Deep learning is a type of artificial intelligence that uses neural networks to train data as shown in figure 8. The network is composed of many layers connected together (51). Each layer will have a different function. It can be used to classify images or predict signals that have similar patterns. Input data that have complex features can be detected when using deep learning. It is inspired by the structure of the human brain.

There are visible layers and hidden layers in the deep neural network. The input layers and output layers that predict the final outcome are visible layers while the others are hidden layers (51). They are trained through a process called forward propagation, where data is processed layer by layer to make predictions or classifications (51). The errors in these predictions are then calculated using backpropagation, which adjusts the weights and biases of the network to improve accuracy. Over time, the algorithm becomes more precise.

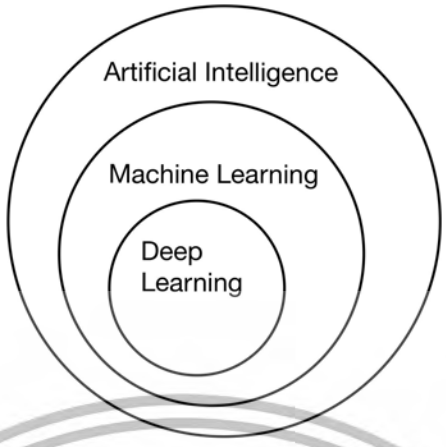


Figure 8 Subset of AI

There are many types of neural networks such as convoluted neural networks (CNN) and Recurrent neural networks (RNN). CNN is usually used for image classification while RNN is recommended for sequential or times series data (51).

There are many types of layers in neural networks such as basic layers, convolutional layers, pooling layers, recurrent layers, normalization layers, and fully connected layers. Different layer types behave differently. The basic layer is the dense layer, which applies a basic transformation with an activation function, often used for changing the dimensions of the data. The convolution layer is used for computer vision tasks and analyzes images by focusing on specific parts to associate them with certain objects or features. The pooling layer reduces the data size by keeping only relevant information, using methods like max pooling or average pooling. Recurrent layers, also known as RNNs, are used in natural language processing (NLP) to analyze sequential data, allowing the model to have memory and consider the context of previous inputs. The normalization layer ensures that the data has a consistent distribution, making it easier for the model to learn. Finally, the fully connected layer is typically used in the

final stages of a neural network architecture, following a series of convolutional and pooling layers in the case of CNNs. They are often employed for tasks such as classification, where the network is required to output a probability distribution over predefined classes (52).

Deep learning toolbox in Matlab makes it easier to train deep learning for images classification. One of the recommended networks is called ResNet-50. In MATLAB, it is composed of 177 layers, but it can be adjusted depending on the size of input and class number of output data. (CNN) (53).

2.11 Noise Removal

ECG signals usually have noises from moving artifacts, respiration, or/and ECG detection devices (54). The first type of noise will cause ECG to be falsely detected which is the most difficult noise to be removed. Another type of noise is produced when the patient is breathing. It is called baseline drift that makes the signal move up and down along the y-axis. This type of noise is alright in this study by applying an adaptive threshold that will be explained in 2.12. The last common type of noise is called power line interference which is caused by a magnetic field. There will be small waves interfering with the ECG which make it more difficult to detect P, QRS, and T waves.

2.12 R Peak Detection

R peak is in the QRS complex of ECG, which is an important component for calculating HRV. It is not simple to detect R peak because of many factors, such as noises in the signal and physical shape of ECG that lead to false and missing in R peak detection (55). There are various methods of R peak detection: applying threshold or using derivative to detect R peak.

2.13 Confusion Matrix

Confusion matrix is used to show the number data that is predicted for true-false positive and negative (56). One axis is the predicted data while another is the actual class of data. Generally, in a 2 times 2 matrix, there will be two boxes that contain the number of true predicted outcomes: true positive and true negative. And there are other two boxes that show the false prediction as shown in figure 9.

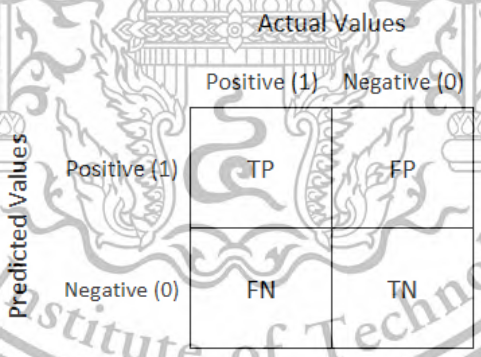


Figure 9 Confusion Matrix

Accuracy, sensitivity, and specificity can be calculated by using formula (3), (4), and (5), respectively. Accuracy is the ratio between true to false value, so the more true prediction, the more accurate. Sensitivity is the ratio between true positive and false negative plus true positive. False negative means that the actual value should be one, but the predicted result shows zero.

The less false negative, the more sensitivity that means it is more likely to include the people who truly have the condition. Conversely, specificity is true negative over false positive plus true negative. The more specificity means that the predicted outcome is more likely to exclude people without the condition.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad \text{—(3)}$$

$$\text{Sensitivity} = \frac{TP}{TP + FN} \quad \text{—(4)}$$

$$\text{Specificity} = \frac{TN}{TN + FP} \quad \text{—(5)}$$

when TP= True Positive, TN= True Negative, FP= False Positive, and FN= False Negative.

Cohen's Kappa is another statistical value that measures the reliability of trained deep learning networks calculated from the confusion matrix. Different ranges of kappa scores will have different meanings. If k is less than or equal to 0, there is no chance of agreement. If k is between 0.01 to 0.02, it means that there is slight agreement. When k is 0.21 to 0.40, it means fair agreement. When k is 0.41 to 0.60, it means moderate agreement. If k is between 0.61 to 0.80, it is a substantial agreement. Lastly, if k is 0.81 to 0.99, it means that the model almost has perfect agreement. The formula (6) is used to calculate Kappa score. (57)

$$\text{Cohen's Kappa} = \frac{2*(TP * TN - FN*FP)}{(TP + FP)*(FP+TN) + (TP+FN)*(FN+TN)} \quad \text{—(6)}$$

when TP= True Positive, TN= True Negative, FP= False Positive, and FN= False Negative.

2.14 Literature Review

There are many studies that use HRV in sleep apnea screening or detection. According to the study of Tyagi et. al., single lead ECG has been used for sleep apnea detection. The study used an ECG dataset of normal and sleep apnea samples that contain 1 minute of annotation, and the signal is in the frequency domain before training by enhanced-deep belief network. the accuracy is up to 90%. Another study from Taiwan converted HRV to images of 10 to 60 seconds into spectrograms (58). They found differences in HRV by observing normal and apnea ECG. The accuracy of the 60 seconds time window is up to 91.4%.

Moreover, some studies used PRV to detect sleep apnea. A study of Lazazzera et. al. PRV is extracted from PPG for CSA, OSA, and mixed sleep apnea classification. SpO2 is another signal that is used for training. According to Dehkordi et. al., there is a significant correlation between HRV and PRV in the time domain only. The estimation of value in the frequency domain is inaccurate (59). Yuda et. al. suggested that PRV should not be considered as a surrogate for HRV, but it should be a new biomarker (60).

The systematic review of Mostafa et. al. (8) collected 21 studies about sleep apnea detection using deep learning. The systematic review compared the effectiveness of different signals that use only a type of signal each, and found that SpO2 signal is the best. However, the systematic review cannot accurately conclude whether the ECG is the best signal or not due to different methods of data preparation and deep learning model (8). Generally ECG signals are

believed to be better, so this study discussed that the result should be compared within a study rather than using different methods of data preparation and model.

Numerous studies use ECG for sleep apnea detection because it is easy to perform and reliable. For example, the single lead ECG can be used to measure the heart's electrical activity by attaching only one electrode (10). As ECG can be converted to HRV that represents the autonomic nervous system, it will be a great choice if a study can pick only one signal. The systematic review of Mostafa et. al. collected 21 studies about sleep apnea detection using deep learning (8). It also found ECG the most common base of signals in sleep apnea detection studies.

However, the systematic review in 2017 compared the effectiveness of different signals, and found that SpO2 signal is the best. Generally ECG signals are believed to be better. According to the study in 2022 of Zarei et. al. that used single-lead ECG signals, it achieved an accuracy of 97.21% of classification, which is very high (61). Therefore, this study discussed that the result should be compared within a study rather than using different methods of data preparation and model. This led to a question whether ECG or PPG is the better one.

Some studies also proposed PRV to be used as the alternative of HRV (12). According to Dehkordi et. al., there is a significant correlation between HRV and PRV in the time domain only as the estimation of value in the frequency domain is inaccurate (59). And, Yuda et. al. suggested that PRVshould not be considered as a surrogate for HRV, but it should be a new biomarker (60). Therefore, this study will discuss the differences in results between using HRV and PRV.

Chapter 3

Methodology and Material

3.1 Introduction

This chapter presents the material and method of developing the algorithm for sleep apnea and hypopnea detection step by step.

3.2 Material

3.2.1 Dataset from MESA Sleep Study (MESA)

Dataset that is used in this study include ECG and PPG signals during the period when sleep apnea or hypopnea occur. It is from Multi-Ethnic Study of Atherosclerosis (MESA) that investigates factors of developing cardiovascular disease. 2,237 participants took a sleep exam called MESA sleep, including overnight polysomnography. The participants' ethnic are black, white, Hispanic, and Chinese-American for either gender who age over 55 years old.

There are three main exclusion criteria. First, the sleeping time record must be more than 6 hours to be included in the study. Second, there must be an annotation file that marks the apnea condition. Third, only the duration when the patient is sleeping will be used.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
1	Index	ID	Age	Age more 89	Sex	Race	ahi_a0h3a (aasm Recom)	ahi_a0h4 (aasm_Accepted)	nsrr_ahi_hp3u	nsrr_ahi_hp4r	nsrr_flag_spsw	scorid5	Start Time	Lights Out	Sleep Onset	Lights On	TST (min)	TST (hours)	End time	Sleep Latency (min)	Sleep Efficiency (%)	waso5 (min)	Study length (Mins)	Study length (Epoch-1)	Study length (Epochs-1)
2	1	2	83	no	0	1	10.75	3.85	8.34	7.38	full scoring	928	20:59:59	22:13:29	22:13:29	5:26:29	374	6.23	7:59:58		86.374134	59	660	1320	1319
3	2	14	60	no	0	3	12.94	6.54	11.37	9.24	full scoring	928	20:59:59	0:30:29	0:34:29	8:49:59	422	7.03	10:59:58	4	84.4	73.5	840	1680	1679
4	3	16	57	no	0	4	4.41	1.42	2.7	3.7	full scoring	939	21:59:59	22:00:29	22:00:59	5:40:59	422	7.03	7:59:58		91.54013	38	600	1200	1199
5	4	21	78	no	0	4	25.91	12.51	19.28	21.06	full scoring	928	21:59:59	22:00:29	22:00:29	6:59:29	470	7.83	6:59:58		87.198516	69	540	1080	1079
6	5	27	72	no	0	1	12.93	5.37	10	8.54	full scoring	939	21:59:59	22:49:29	22:49:29	7:25:59	492	8.20	7:59:58		95.16441	24.5	600	1200	1199
7	6	46	79	no	0	1	24.81	19.69	23.41	22.48	full scoring	928	21:59:59	22:09:59	22:09:59	6:34:59	387	6.45	7:59:58		76.633663	118	600	1200	1199
8	7	48	82	no	0	1	9.28	4.31	8.12	5.97	full scoring	928	21:59:59	23:00:29	23:02:29	7:36:59	362	6.03	7:59:58	2	70.019342	152.5	600	1200	1199
9	8	50	68	no	1	3	9.97	5.59	8.92	7.1	full scoring	928	20:59:59	20:59:59	20:59:59	5:01:29	397	6.62	6:59:58		82.365145	84.5	600	1200	1199
10	9	56	63	no	0	3	5.03	1.55	4.39	2.45	full scoring	928	21:59:59	23:00:29	23:10:29	8:16:29	465	7.75	9:59:58	10	83.633094	81	720	1440	1439
11	10	70	59	no	0	1	9.21	4.36	8.97	4.73	full scoring	928	21:59:59	23:19:29	23:19:29	8:12:29	495	8.25	8:59:58		92.870544	38	660	1320	1319
12	11	74	75	no	1	3	22.46	11.08	17.44	18.92	full scoring	928	19:59:59	19:59:59	19:59:59	5:32:59	406	6.77	7:59:57		70.855148	167	720	1440	1439
13	12	77	61	no	1	4	6.34	2.76	4.55	4.97	full scoring	928	22:29:59	23:05:29	23:05:29	7:08:29	435	7.25	8:29:59		90.062112	48	600	1200	1199
14	13	84	58	no	0	4	8.41	4.35	6.67	7.1	full scoring	928	20:59:59	22:24:59	22:24:59	6:27:29	414	6.90	6:59:58		85.714286	68.5	600	1200	1199
15	14	85	60	no	0	3	12.46	3.31	4.96	10.93	full scoring	939	18:29:59	19:30:59	20:43:59	5:34:29	472	7.87	6:29:58	73	78.145695	58.5	720	1440	1439
16	15	87	75	no	0	3	12.63	7.46	11.91	8.61	full scoring	939	21:30:00	22:25:30	22:25:30	5:42:00	418	6.97	7:30:00		95.652174	18.5	600	1200	1199
17	16	99	71	no	1	1	26.54	16.98	24.4	21.76	full scoring	916	19:29:59	20:24:59	20:24:59	4:03:59	364	6.07	5:59:58		79.302832	95	630	1260	1259
18	17	101	63	no	0	4	7.67	2.92	6.14	5.06	full scoring	928	21:59:59	21:59:59	21:59:59	5:40:29	391	6.52	7:59:58		84.815618	69.5	600	1200	1199
19	18	107	82	no	1	1	58.83	49.96	52.57	56.61	full scoring	939	21:29:59	22:23:29	22:23:29	6:34:29	460	7.67	9:29:58		93.686354	31	720	1440	1439
20	19	109	66	no	1	4	23.02	8.73	16.03	17.3	full scoring	939	20:03:05	23:23:05	23:23:05	7:01:35	378	6.30	8:00:00		82.352941	80.5	716.9	1433.8	1433
21	20	113	61	no	0	2	3.27	0.33	1.96	2.29	full scoring	928	20:59:59	22:00:59	22:37:59	5:09:29	367	6.12	8:59:58	37	85.547786	24.5	720	1440	1439
22	21	118	56	no	0	4	13.59	7.95	12.87	9.69	full scoring	939	20:59:59	23:00:59	23:07:29	6:59:29	415	6.92	6:59:58	7	86.638831	57	600	1200	1199
23	22	125	82	no	1	1	46.65	38.14	43.72	41.66	full scoring	939	19:29:59	19:57:29	19:57:29	4:46:29	409	6.82	7:29:59		77.31569	120	720	1440	1439
24	23	132	62	no	1	4	6.83	1.25	4.55	4.21	full scoring	916	20:59:59	21:03:29	21:03:29	6:59:29	527	8.78	6:59:58		88.422819	69	600	1200	1199
25	24	144	67	no	0	4	20.95	9.42	16.61	15.25	full scoring	928	19:59:59	21:30:59	21:46:59	6:57:29	484	8.07	7:59:58	16	85.361552	66.5	720	1440	1439
26	25	155	76	no	1	4	34.9	25.84	29.55	31.63	full scoring	939	21:59:59	22:07:29	22:07:29	6:59:59	404	6.73	6:59:59		75.797373	128.5	540	1080	1079
27	26	159	58	no	0	4	4.1	0.68	1.37	3.42	full scoring	928	20:29:59	21:00:59	21:50:29	6:14:59	439	7.32	6:29:58	50	79.241877	65.5	600	1200	1199

Figure 10 Inclusion File

The dataset includes only patients who complete more than six hours of sleeping as shown in figure 10. And all datasets must contain apnea annotation. After excluding all datasets that do not match this criteria, there will be 1,122 datasets left. Finally, only 500 patients' ECG and PPG signals with annotation files will be picked from the total number of patients that pass the inclusion and exclusion criteria by using the random table. The dataset is in edf file and the apnea label is in xml file. There are more than 5,529,600 doubles in each file of the dataset. The frequency of this dataset is 256 Hz.

3.2.2 MATLAB Deep Network Designer and ResNet-50

Deep Network Designer is a tool in the MATLAB program for deep learning (53). The input is recommended to be images. The pre-constructed neural network that will be used in this study is called RessNet-50. The input layer is for images with 224 x 224 x 3 pixels or for RGB images. It can be adjusted by adding new layers conveniently within the deep network designer.

3.3 **Methods**

There are three main steps of the methodology, including data preparation, training, and testing. In data preparation, two main signals: ECG and PPG will be processed in two separate parts. The overall process is in the box number one to eight in the figure 11 below. And the training and testing part is in box number 9.

Figure 11 Overall Process

3.3.1 Data Preparation

- ECG Time Domain Data:

Step 1: Set inclusion criteria for the dataset. First sleep time of patients must be more than 6 hours. Second, use only the signal of patients that have apnea annotation. Excel will be

used for filtering to include only the patient number that matches this inclusion criteria. The inclusion numbers can be exported into a .mat file for automatic file preparation.

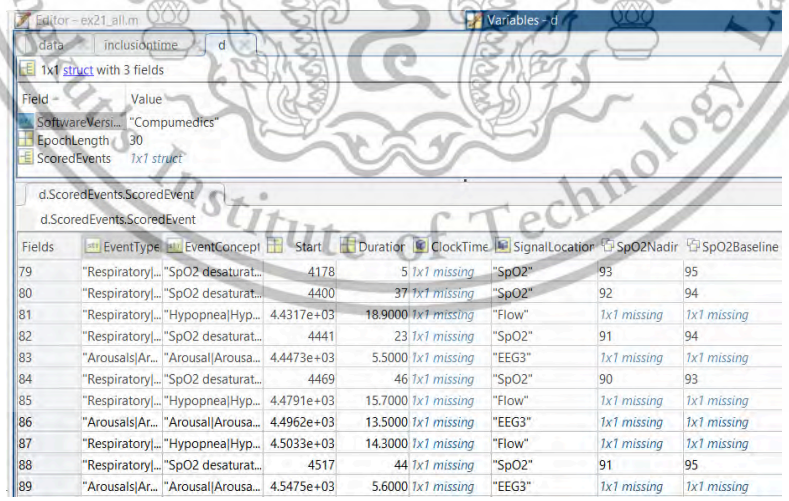
Step 2: Open the included dataset that is the edf file by using the 'edfread' builtin function in matlab. The Names of different files from the open source can be automatically uploaded by using only counting number arrays by using the following code in table 1. The full text of coding is located in the appendix A.

Table 1 Code for Automatic File Preparation

Line 1	for i= 1:170
Line 2	i
Line 3	if fileno(i) <= 9
Line 4	file= append('mesasleep\mesa-sleep-000',num2str(fileno(i)),'.edf');
Line 5	elseif fileno(i) <=99
Line 6	file= append('mesasleep\mesa-sleep-00',num2str(fileno(i)),'.edf');
Line 7	else
Line 8	file= append('mesasleep\mesa-sleep-0',num2str(fileno(i)),'.edf');
Line 9	end
Line 10	data= edfread(file);
Line 11	ECG= cell2mat(data.EKG);

Line 1 is to run “i” equal 1 then plus one until it reaches 170 and line 2 is to show where the loop is going on. The end of this for loop will be at the bottom of overall code. And line 3 to 11 is the condition to write the file name that matches the file name from the open source. ‘Append’ is a builtin-function in MATLAB to connect string variable. Those file are saved in the folder “mesasleep” that is in the same directory. Finally, line 10 is to read the edf file and ECG signal will be imported into double by using the code in line 11.

Step 3: Open xml file into MATLAB by using a builtin function called ‘readstruct’. This xml file keeps all information of respiratory events such as SpO2 and sleep stage as shown in the figure 12, so only information about apnea will be picked out. Then create a table to note all periods of apnea, including obstructive sleep apnea, hypopnea, and unsure that is also a period of hypopnea. The code below is only the important part for constructing the table, and the others will be in the appendix A.



Fields	EventType	EventConcept	Start	Duration	ClockTime	SignalLocation	SpO2Nadir	SpO2Baseline
79	"Respiratory|...	"SpO2 desaturat...	4178	5	1x1 missing	"SpO2"	93	95
80	"Respiratory|...	"SpO2 desaturat...	4400	37	1x1 missing	"SpO2"	92	94
81	"Respiratory|...	"Hypopnea|Hyp...	4.4317e+03	18.9000	1x1 missing	"Flow"	1x1 missing	1x1 missing
82	"Respiratory|...	"SpO2 desaturat...	4441	23	1x1 missing	"SpO2"	91	94
83	"Arousals|Ar...	"Arousal|Arousa...	4.4473e+03	5.5000	1x1 missing	"EEG3"	1x1 missing	1x1 missing
84	"Respiratory|...	"SpO2 desaturat...	4469	46	1x1 missing	"SpO2"	90	93
85	"Respiratory|...	"Hypopnea|Hyp...	4.4791e+03	15.7000	1x1 missing	"Flow"	1x1 missing	1x1 missing
86	"Arousals|Ar...	"Arousal|Arousa...	4.4962e+03	13.5000	1x1 missing	"EEG3"	1x1 missing	1x1 missing
87	"Respiratory|...	"Hypopnea|Hyp...	4.5033e+03	14.3000	1x1 missing	"Flow"	1x1 missing	1x1 missing
88	"Respiratory|...	"SpO2 desaturat...	4517	44	1x1 missing	"SpO2"	91	95
89	"Arousals|Ar...	"Arousal|Arousa...	4.5475e+03	5.6000	1x1 missing	"EEG3"	1x1 missing	1x1 missing

Figure 12 xml file

Table 2 Constructing Table for Apnea Period

Line 1	for s1= 1:length(d2)
Line 2	if (d2(s1).EventConcept == "Hypopnea|Hypopnea" || d2(s1).EventConcept == "Obstructive apnea|Obstructive Apnea" || d2(s1).EventConcept == "Unsure|Unsure")
Line 3	Events(end+1)=d2(s1).EventConcept;
Line 4	Start(end+1) = d2(s1).Start;
Line 5	Duration(end+1) = d2(s1).Duration;
Line 6	end
Line 7	end

According to table 2, line 2 to 5 is to check whether the event concepts that are in the second column of figure 12 match the word 'Hypopnea', 'Obstructive Apnea', or 'unsure' or not. If it is matched, the start time, duration, and event type will be recorded in a table. The period where sleep apnea and hypopnea will be labeled after calculating HRV.

Step 4: Select only the period after the patient falls asleep. Coding by using for-loop will be used to identify the start time of sleep onset and remove the signal that is recorded before that time of every dataset. The start time and end time can be saved within the same file with inclusion file number as in the figure 13.

	ID	TST (hours)	Time start sec	End time sec
2	2	6.23 22440.00	4410	26850.00
3	14	7.03 25320.00	12630	37950.00
4	16	7.03 25320.00	30	25350.00
5	21	7.83 28200.00	30	28230.00
6	27	8.20 29520.00	2970	32490.00
7	46	6.45 23220.00	600	23820.00
8	48	6.03 21720.00	3630	25350.00
9	50	6.62 23820.00	0	23820.00
10	56	7.75 27900.00	3630	31530.00
11	70	8.25 29700.00	4770	34470.00
12	74	6.77 24360.00	0	24360.00
13	77	7.25 26100.00	2130	28230.00
14	84	6.90 24840.00	5100	29940.00
15	85	7.87 28320.00	3660	31980.00

Figure 13 Inclusion File in mat Type

The following code is to fix the range of data based on the inclusion time. Inclusiontime is an array of time in seconds when the patient is sleeping until the end. However, in 1 second of recording, there are 256 data points. So, according to the table 3 below, ECGtime in line 2 is to expand inclusion time to divide 1 second into 256 seconds while ECGfil in line 3 is the amplitude of ECG.

Table 3 Code for Creating Array in x and y Axis

Line 1	inclusiontime= double(starttime(i)+1:(tab.End(end)-(inv/2)+1));
Line 2	ECGfiltime= [inclusiontime(1): 1/fs : tab.End(end)]';
Line 3	ECGfil=ECGfil(inclusiontime(1)*fs : tab.End(end)*fs);

Step 5: Signal processing

5.1 Normalize ECG signals by using z-score based on the formula (7). $\bar{x}$ is the mean of voltage of all data points while σ is standard deviation. Each z will depend on the value of amplitude x in each data point and z will be used to replace amplitude x.

$$Z = \frac{x - \overline{X}}{\sigma} \quad \text{—(7)}$$

The value must be normalized into the unit of microseconds. As the frequency of this dataset is 256 Hz, the range of time will be divided by 256 and times 1000 before calculating HRV.

5.2 Remove noise in ECG signals by using band-pass filters that include both low-pass filter and high-pass filters. Built-in functions of MATLAB can be used to filter high and low frequencies.

Table 4 Code for Band Pass Filter

Line 1	ECGfil= lowpass(ECG,15,256);
Line 2	ECGfil= highpass(ECG,5,256); Or
Line 3	fil=[5 15]*2/fs;
Line 4	[a,b] = butter(3,fil);
Line 5	ecg_b = filtfilt(a,b,ecg_h);
Line 6	ecg_b = ecg_b/ max(abs(ecg_b));

According to table 4, line 1 means that frequencies that are higher than 15 Hz will be removed by a low pass filter while line 2 means that frequencies that are lower than 5 Hz will allow higher frequency to pass the filter. Another function in line 3 to 6 can also be used as a band pass filter. Line 3 is to let the signal with frequencies between 5 to 15 Hz to pass through the filter. In line 4, 3 is the third order of Butterworth filter design to create a band-pass filter. Line 5 and line 6 is the result of amplitude after filtering as shown in figure 14 and table 5. Other parts of the band pass filter are in the Appendix B.

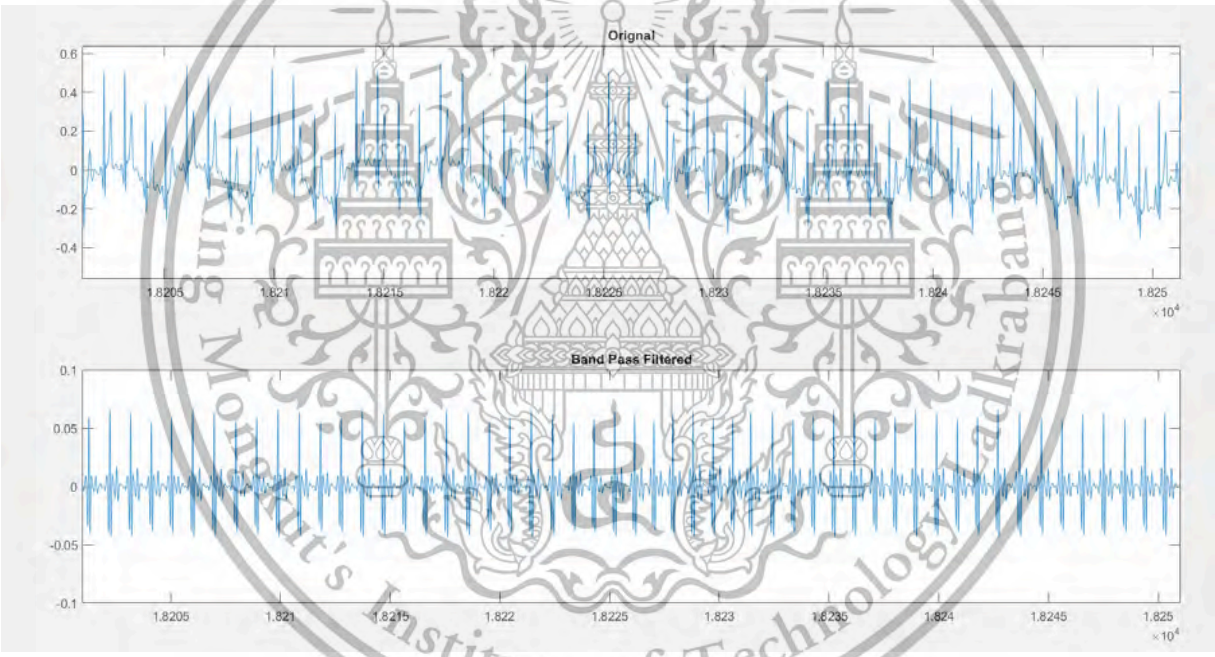
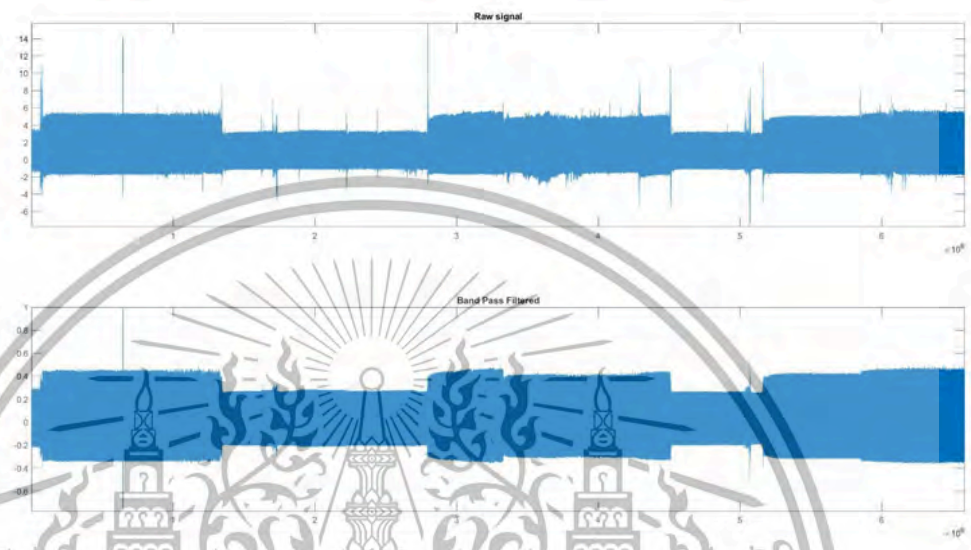
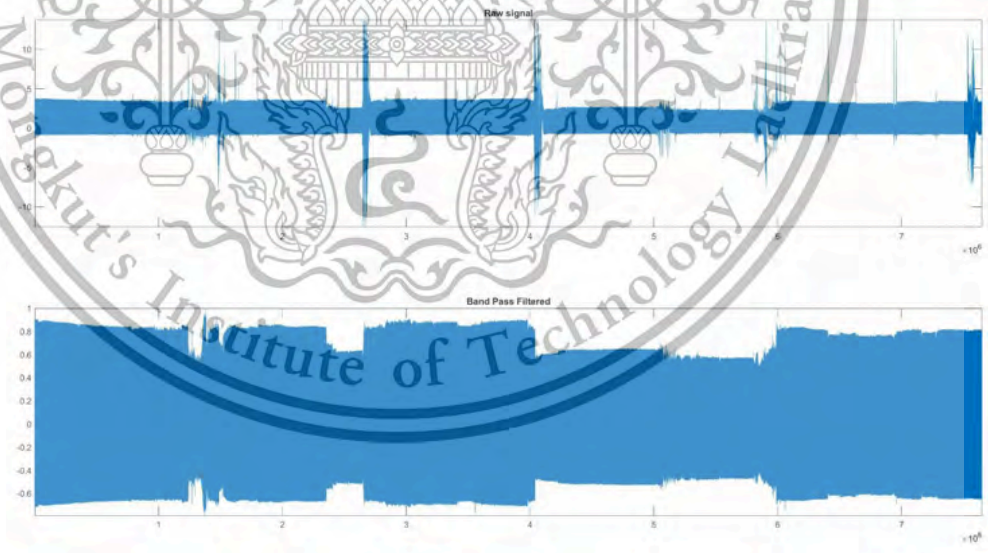


Figure 14 Filtered ECG Signal

Table 5 Filtered ECG Signal in Time Domain

dataset No.	Filtered ECG signal in time domain
2	
14	

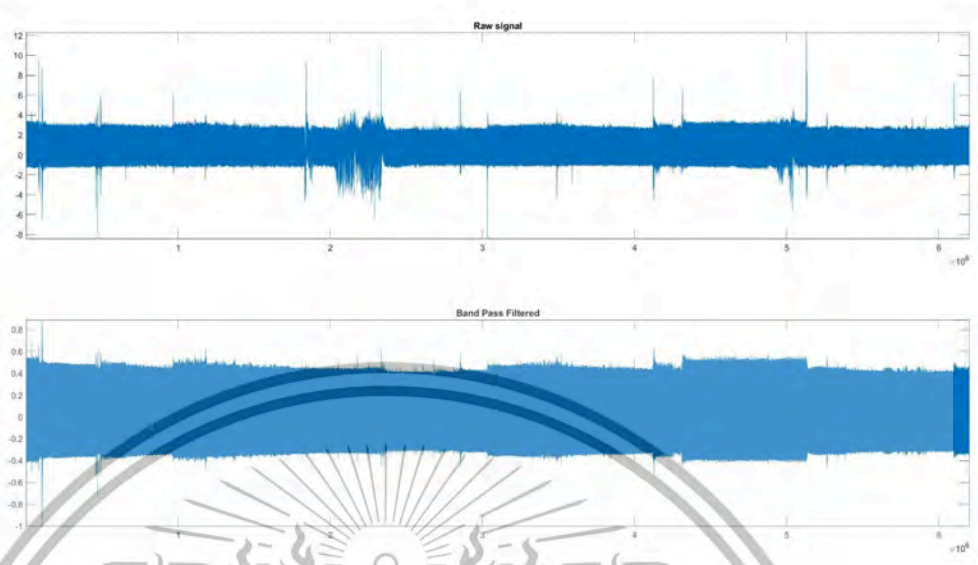
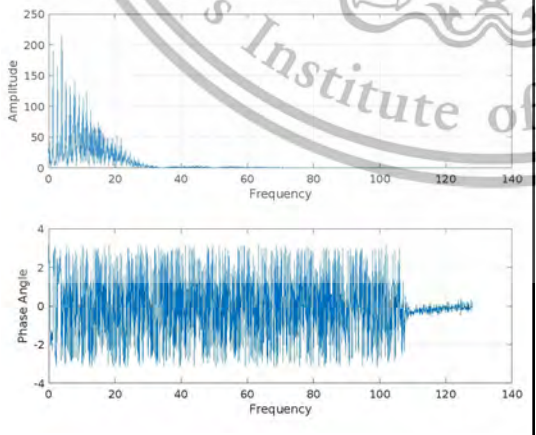
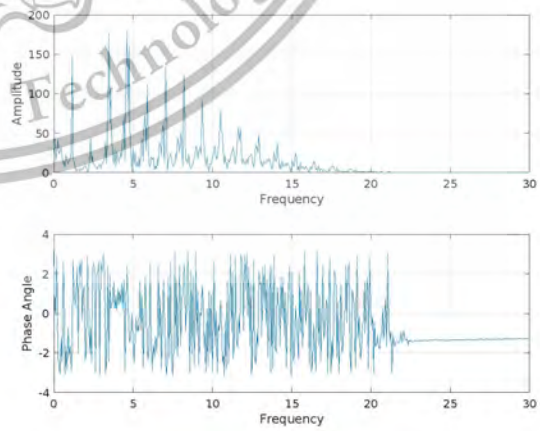
16	

Table 6 Filtered ECG Signal in Frequency Domain

dataset No.	Filtered ECG signal in frequency domain	
	Before Filtering	After Filtering
2		

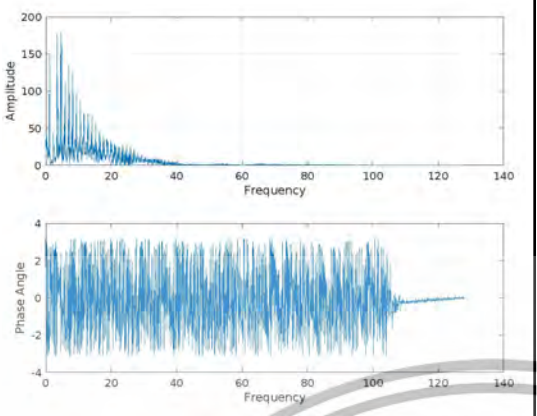
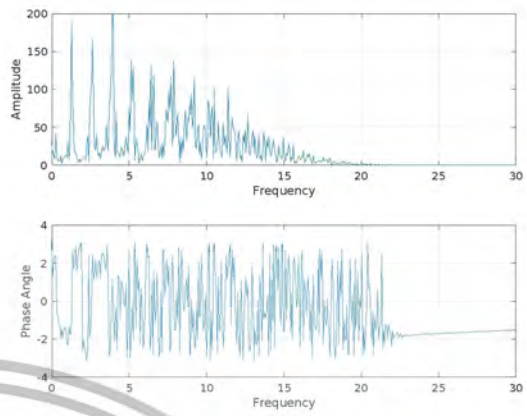
14		

Table 6 shows the frequency domain of the ECG signal by using the fourier transform "fft" in MATLAB. The figure on the left hand side is the ECG signal before passing the band pass filter, so there are frequencies from 0 Hz to over 120 Hz. The amplitude of a signal above 20 Hz is quite low which means that the small noisy signal interferes within the large ECG signal and it should be filtered out. However, on the right hand side figure, the frequencies over 15 Hz are removed. And some of the frequencies below 5 Hz are lessened.

5.3 R peaks of ECG are detected by using adaptive threshold and second derivative. The algorithm of R peak detection is shown in figure 15. The overall code is in the appendix A.

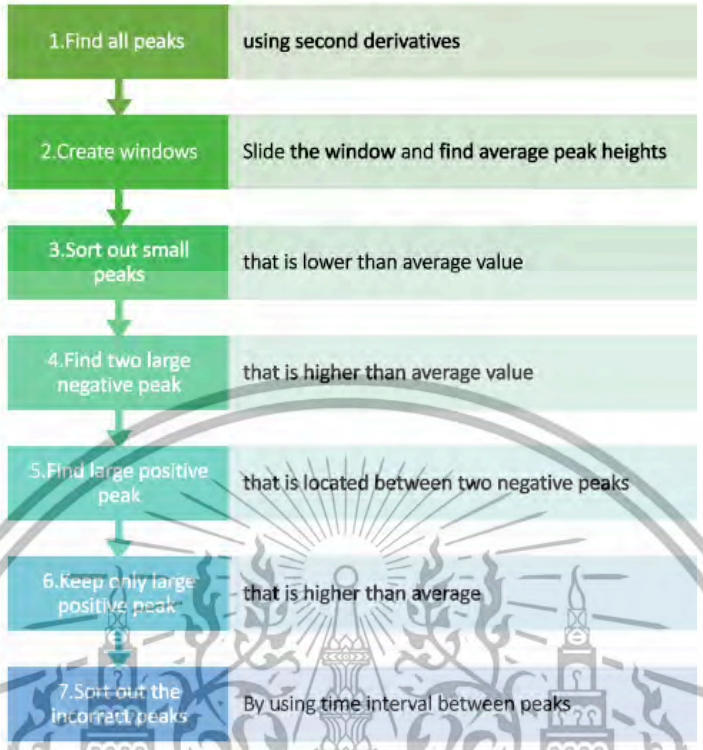


Figure 15 R Peak Detection Algorithm

According to figure 15, the first step can be done by using either built-in function or the formula of the first derivative as shown in table 6. Then if the first derivative changes from negative to positive value or vice versa, that will be a peak. All small peaks will be detected, so it is necessary to remove other peaks except R peak, which is the largest. In step 2, the mean absolute value of R peak amplitude will be used as a threshold. The point that is lower than the threshold will be removed in step 3. From observation, the R peak is usually located between two large negative peaks, so in step 4 and 5 will keep only the amplitude of the large positive peak. The last step is to remove the incorrect peak by using a condition. For example, if there are two points (x1, y1) and (x2, y2) as in figure 16. If x2-x1 is less than the half mean RR interval, the point with lower y will be removed.

Table 7 Find Peak Formula

Line 1	findpeaks(ECG, 'MinPeakHeight', threshold);
Line 2	m = zeros(1, length(Yt)-1);
Line 3	for inj = 1:length(Yt)-1
Line 4	m(inj) = Yt(inj+1) - Yt(inj);
Line 5	end

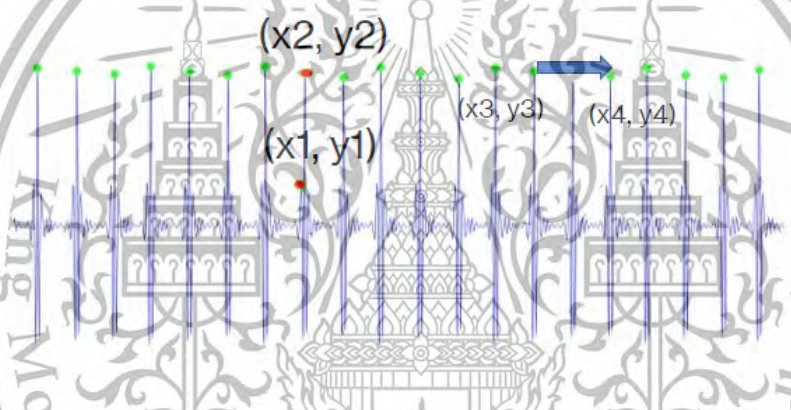


Figure 16 Incorrect Peak Detection

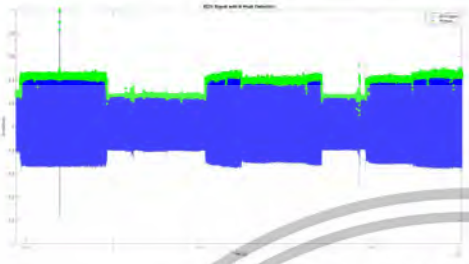
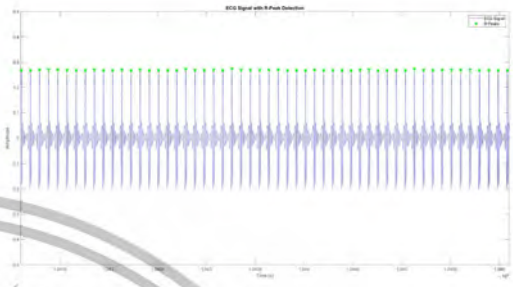
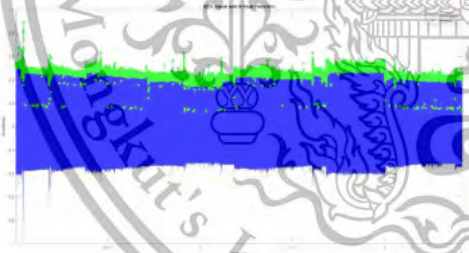
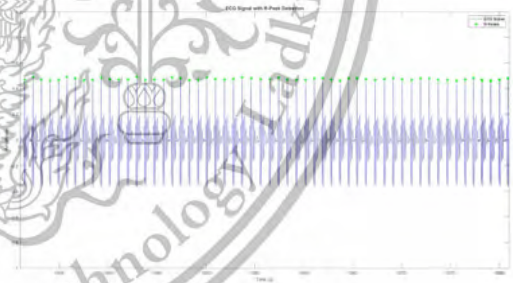
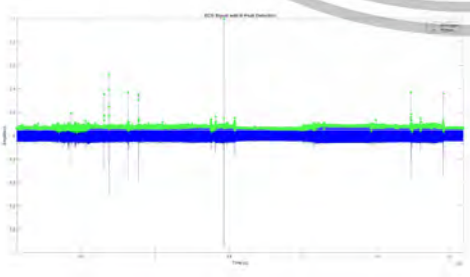
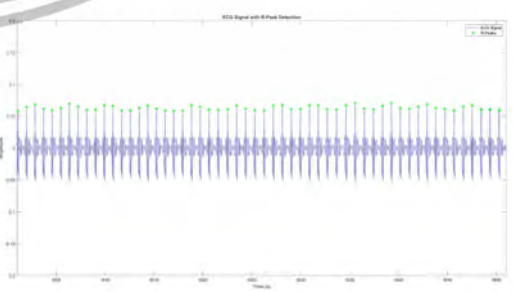
There is one parameter that can be used to detect the incorrect peak by counting numbers when the RR interval is too large. For example, based on figure 16, if x4 minus x3 is larger than average RR interval, missed peak will be counted. The datasets with high count numbers will be removed. Some files also have too much noise, so HRV values are much higher than other datasets. That kind of datasets will be removed as well and will be noted in a file called "exclusion". In total, there are 22 being removed from 137 datasets, so 115 datasets are left.

Step 6: Plot graphs of ECG. The x-axis is time while the y-axis is voltage. And Also plot R peak point by using the MATLAB function as shown in table 8. Line 1 is to plot time in x axis and ECG signal amplitude in y axis in blue color in figure 1. Line 2 is to move the axis to fit the graph. Line 3 and 4 is to plot R peak points. Line 5 to 7 is to label axis, legend, and title respectively. Line 9 is to zoom into the ECG signal as shown in the second column of table 8. The number in the axis can be adjusted then the figure will be saved into the folder within the same directory. If the code is automatically run, it is important to close the figure or use "hold off" before plotting a new figure.

Table 8 ECG Plot Code

Line 1	figure(1); plot(t, ecg_signal, 'b');
Line 2	axis tight;
Line 3	hold on;
Line 4	scatter(xout, yout, 'filled','g');
Line 5	xlabel('Time (s)'); ylabel('Amplitude');
Line 6	legend('ECG Signal', 'R-Peaks');
Line 7	title('ECG Signal with R-Peak Detection');
Line 8	saveas(figure(1),"figure4\"+inclusion(i,1)+".jpg");
Line 9	figure(1), axis([inclusiontime(1)+17600 inclusiontime(1)+17650 -1 1]);
Line 10	saveas(figure(1),"figure4\"+inclusion(i,1)+"rpeak-3.jpg");
Line 11	close(figure(1))

Table 9 R Peak Detection Result

No.	R Peak Detection	Zoom In
2		
14		
16		
46		

Step 7: Calculate the HRV in the form of RMSSD by using a for loop. The number of R peaks that will be converted into a value of HRV will be specified based on the criteria of sleep apnea more than 10 seconds. For easier understanding, 5-minutes HRV intervals are also calculated.

Table 10 RMSSD Formula

Explanation	HRV Formula MATLAB Code for RMSSD
Time of R point	HRV1e= rpeaktime*1000; %unit= ms
Time between RR interval	HRV2e=HRV1e(2)-HRV1e(1); for n=2:length(HRV1e)-1 HRV2e(end+1)= HRV1e(n+1)-HRV1e(n); end
Time difference between two RR interval then square (SSD)	HRV3e=HRV2e(2)-HRV2e(1); for m=2:length(HRV2e)-1 HRV3e(end+1)= HRV2e(m+1)-HRV2e(m); end HRV3e=HRV3e.^2;
Sum of SSD in a specific interval (inv = 10 and 300 seconds)	HRV4e=[]; for le= 1: length(HRV3e)-inv+1 HRV4e(le)= sum(HRV3e(le:le+inv-1)); end

Square root of sum ssd divided by number of RR interval	rmssde= sqrt(HRV4e./(inv-1));
Time of HRV	Time = HRV1e((inv/2)+1 : length(HRV1e)-(inv/2)-1); Time = (Time/1000)'; %unit:sec

Step 8: Label the apnea period by using the table that is created in step three as shown in table 2. To create a label for an RMSSD array, an array that is equal to the size of RMSSD is needed. The MATLAB code and further explanation is in the table 11 below. Moreover, the label for the ECG signal is different from the RMSSD because the size of the array is different. That part of code will be in the appendix A.

Table 11 MATLAB Code for Apnea Label

Explanation	MATLAB Code for Apnea Label

zz is the label. 0 means normal while 1 means apnea. If z is in the time interval between start and end of apnea, zz will equal to 1. Or else, zz will still equal 0.	zz= zeros(length(Time),1)';
	for z1= 1:length(Start)
	for z2 = 2: length(zz)-1
	if (Time(z2-1) <= End(z1) && Time(z2+1) >= Start(z1))
	zz(z2) = 1;
	end
	end
	end
	ze= zz.*rmssde;

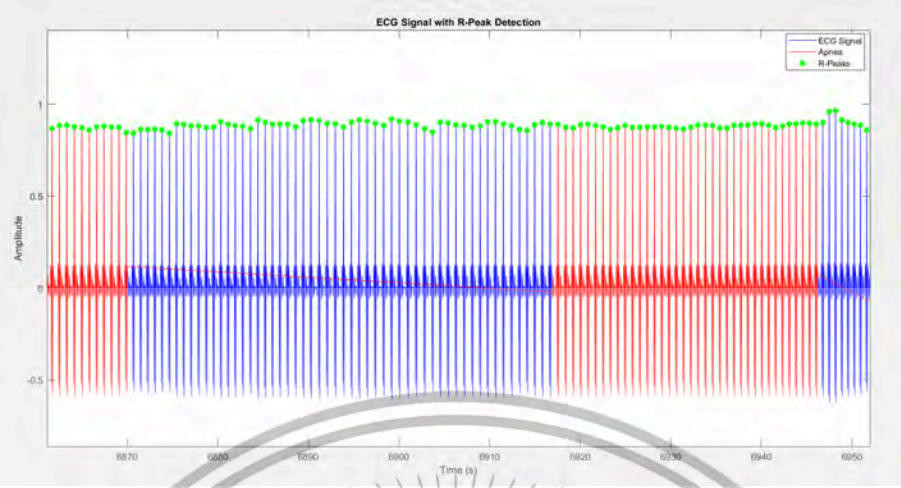
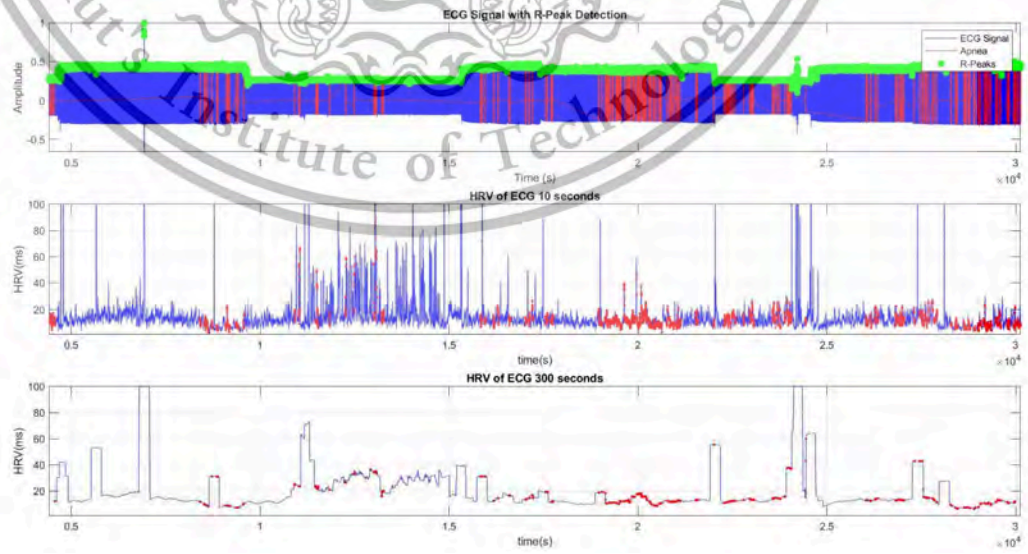
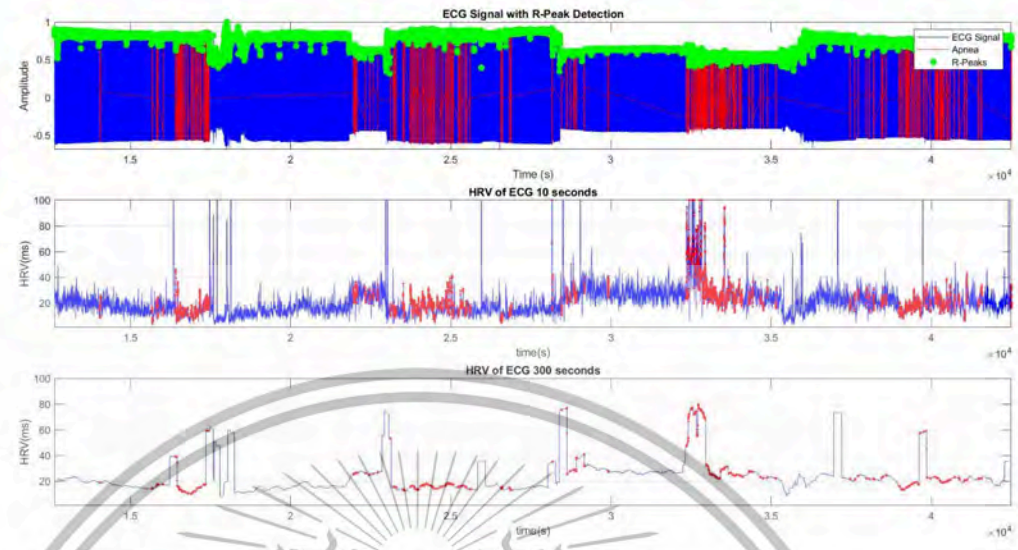
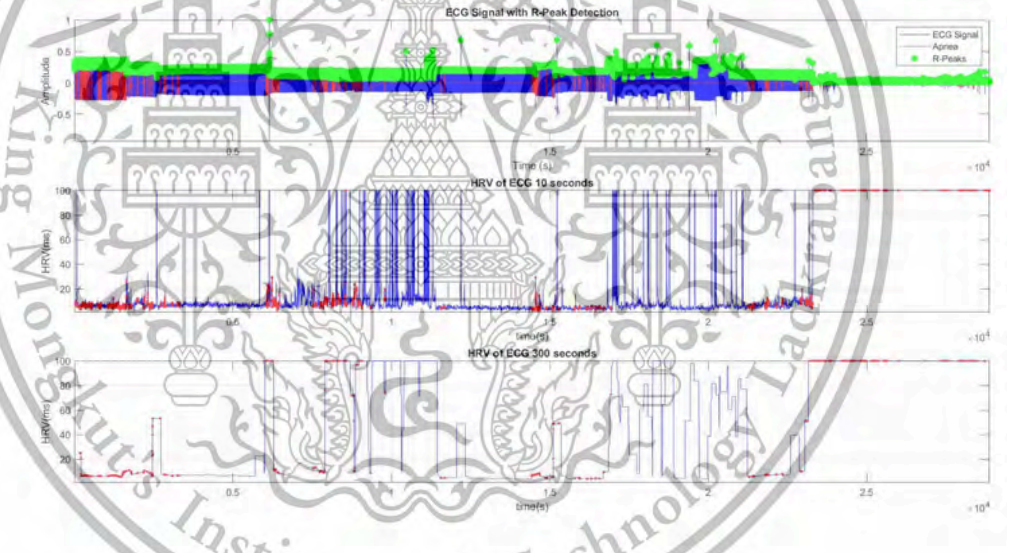


Figure 17 Normal and Apnea Label in ECG

Table 12 below shows the result example from step 7 and 8. The green dot means R peak of ECG while the red dots or lines are apnea labels. The first subplot in each figure is an ECG signal with apnea label. And the second and third subplot is HRV with apnea label.

Table 12 ECG and HRV of Three datasets

dataset No.	ECG and HRV with Apnea Label
2	

14	
50	

In AI training, RMSSD with an interval of 10 seconds, which is the subgraph in the middle of every figure in table 12, will be used. However, in some datasets, there are the areas that have noises or irregular shapes of ECG signals that make it difficult or impossible to perform R peak detection. The RR interval will become wider than normal, and the RMSSD value will be more than 100. One of the examples that is very clear is in dataset number 50 as

shown in the last figure of table 11. Therefore, this kind of area will be cleared when using the condition to remove when RMSSD is equal or more than 100. figure 18 below shows the RMSSD graph after removing the unusual value.

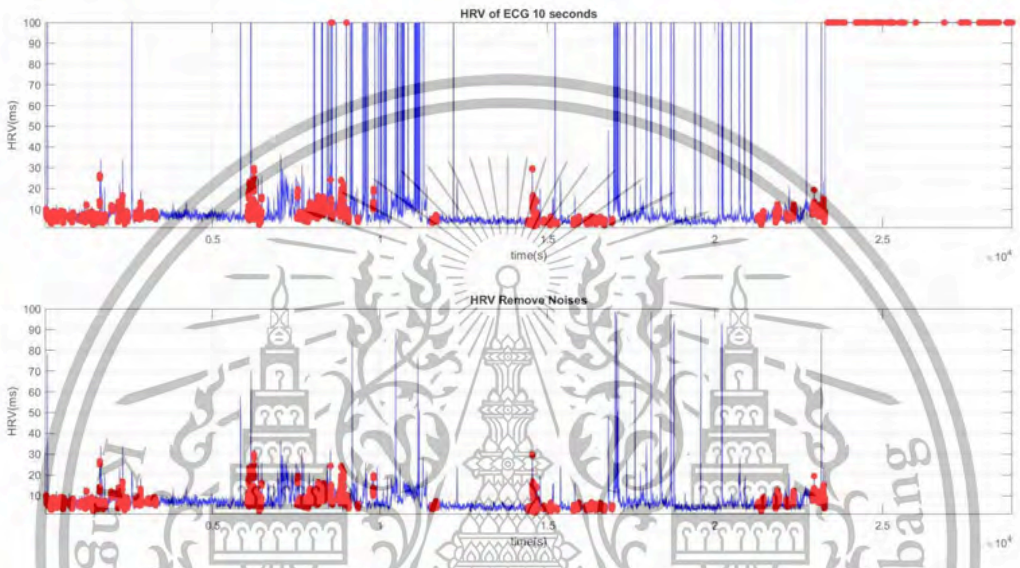


Figure 18 Compare RMSSD Before and After Remove Unusual Value

Step 8: From the previous step, the results are an array of the size equal to 3 rows times the length of RMSSD columns, depending on each dataset. The first row is the real time that matches RMSSD in the second row, and the third row is the label that 0 means normal and 1 means apnea. In this step, this array will be segmented into 60 seconds each, and be classified into two classes separately. Therefore, row 2 and row 3 will be mainly used for training while row 1 will be noted in the case of testing in the later part.

In this study, the array result name is “memat”, and the array of HRV in row 2 will be classified into the period of normal and apnea (sleep apnea and hypopnea) based on the label in

row 3, as shown in the code part 1 in table 13. The period of each apnea event is more than 10 seconds, which is the minimum criteria for apnea diagnosis. However, segmenting training data into 10 seconds is too short, and based on the literature review usually segmented data into 60 seconds.

There are also less apnea periods, so the condition for apnea should be less strict compared to the normal class. Training by using balanced data is more recommended. To increase the number of apnea classes, the window sliding of apnea will be half or one-third of normal. And the window sliding value of normal class is 60 seconds. If the sum of labels in 60 HRV in each window is 0, it will be classified into normal class. However, for apnea class, HRV that matches the position of apnea when the label is 1 will be selected into an array of 1 times 60 array. To prevent data preparation bias that might occur because of this process, in the testing part, the way to generate testing results for apnea class will be the same as the normal class. The MATLAB code for data segmentation and separation will be explained in table 13 below.

Table 13 Explanation for Data Segmentation and Separation Coding

Explanation	Code

Part 1
To run "memat" in all datasets.
Row 2= RMSSD
Row 3= label
Row 1= time
	for mm = 1:115 % 1: length(memat)
	me= memat{mm};
	rmssde= me(2,:);
	ze= me(3,:);
	time= me(1,:);

Part 2
This part is to remove RMSSD that exceeds 100 because of noises or incorrect R peak detection.
	r0=0;
	for r00=1:length(rmssde)
	  if rmssde(r00)==100
	  r0(r00)= r00;
	  end
	end
	r0=nonzeros(r0);
	rmssde(:,r0)=[];
	ze(:,r0)=[];

Part 3
 	
The variable r_one means the label of apnea while r_zero is the label of the normal part.
 	
If the label in the 60 seconds segment is all zero (normal), RMSSD of that part will be recorded in r_zero.
 	
However, all RMSSD when the label is 1 (apnea) will be
	r_zero= zeros(length(rmssde),60);
	z=ze;
	for r2= 1: 60: length(rmssde)-59
	  if sum(z(r2:r2+59))==0
	  r_zero(r2,:)=rmssde(r2:r2+59);
	  end
	end
	r_ap= zeros(length(z),1);
	
	for r2= 1:length(rmssde)-59
	  if z(r2)==1
	  r_ap(r2)=rmssde(r2);

recorded in r_ap before being segmented by using the window size of 40. All segments of the apnea part will be recorded in r_one.	end
	end
	r_ap= nonzeros(r_ap);
	
	for r2= 1: 40:length(r_ap)-59
	r_one(r2,:)=r_ap(r2:r2+59);
	end

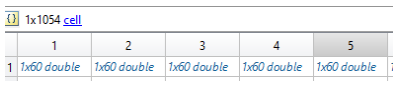
Part 4
This part is to remove unrelated rows that are equal to zero. Then all HRV values will be saved to a 'cell' called TApnea and TNormal separately.
 	
Therefore, in each cell, there are 115 subcells of 115 datasets. And there will be multiple segments of HRV value with length of 60 values in each sub-cell.
	r4=0; r5=0;
	for r3=1:length(rmssde)
	if r_zero(r3,:)==zeros(1,60)
	r4(end+1)= r3;
	end
	if r_one(r3,:) == zeros(1,60)
	r5(end+1)= r3;
	end
	end
	r4(1)=[]; r5(1)=[];
	r_zero(r4,:)=[];
	r_one(r5,:)=[];
	tsla=[];
	nor=[];
	for r6=1:size(r_one,1)

	tsla{r6}=r_one(r6,:);
	end
	for r7=1:size(r_zero,1)
	nor{r7}=r_zero(r7,:);
	end
	
	TApnea{mm}= tsla;
	TNormal{mm}= nor;
	end

Step 9: Convert the segment arrays of HRV into jpeg images by using a builtin function called 'image' and the option of CDataMapping 'scaled'. Set the size of the figure as 224 square pixels. Save them into two folders: normal and sleep apnea or hypopnea. Table 14 is the example of code that converts 1D HRV segments in apnea class into images. The overall code will be in the appendix C.

Table 14 Convert HRV Array to Images

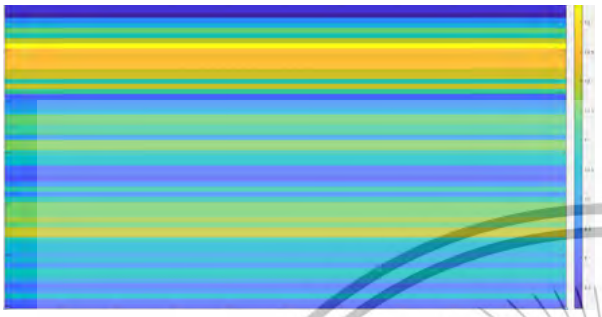
Explanation	**MATLAB Code**

- Connect all subcells
 
 	
- Convert all cells to double
	tA = [TApnea{:,1:115}];
	tA= cell2mat(tA);
	tsla= reshape(tA,[60 39]);
	size=[100 100 143.6 143.6];

then reshape into one large array.
- Each image will use 1 x 60 HRV values.
- Images will be saved into the specific folder with the name i.jpg when i is the number of the cell.
	for i=1:length(tsla)
	z=tsla(:,i);
	figure("Visible","off","Units","pixels","Position", size);
	axes("Position",[0 0 1 1]);
	image = image(z,'CDataMapping','scaled');
	saveas(image,"folder"+ i +".jpg");
	clear image;
	end

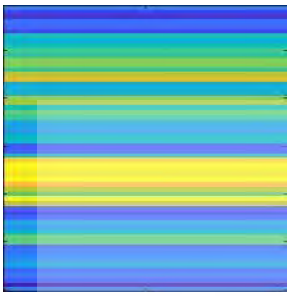
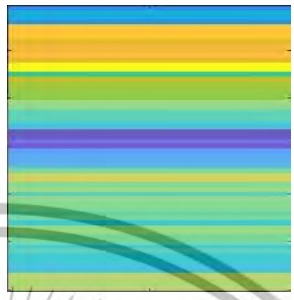
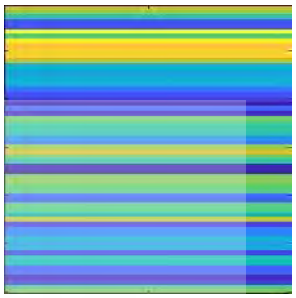
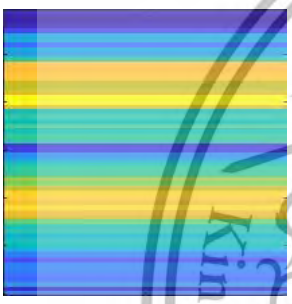
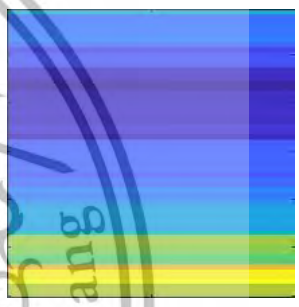
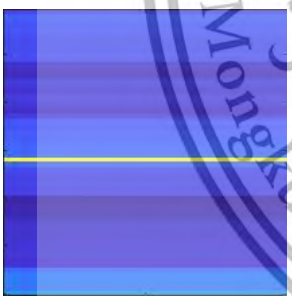
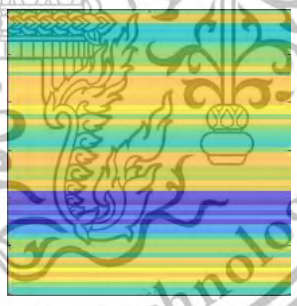
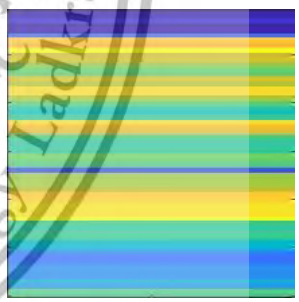
According to table 14, “image = image(z,'CDataMapping','scaled');” can generate images by mapping 60 HRV values into 224 pixels, and in one row is only one value. On the left hand side is a color scale that shows which value is equal to which value. Yellow color is the greatest value within that range while dark blue color is the lowest value. However, each image will have a different range of values. For example, the dark blue color in one picture might be 8 while another might be 16 in the image in row one of table 15. Therefore, an image will represent the variability in range of HRV value rather than an HRV value itself. There is also an option to delete the scale, the axis, and the frame to make images fit without being surrounded by unrelated objects, as shown in row two of table 15. And the images are adjusted into 224 x 224 x 3 pixels by using the code size=[100 100 143.6 143.6]; when positioning the image in the figure in MATLAB. Moreover, these images can be automatically saved by “saveas” in MATLAB as mentioned in table 14.

Table 15 Image Generating

Scale = 8 to 14	Scale = 15 to 36
	
	

The images in two classes are difficult to be classified by human's eyes. Examples of images from HRV of two classes are shown in table 16. And they are saved in two different folders as in figure 19 that will be used to train in the next section.

Table 16 Images of Normal and Apnea

Normal		Apnea	
			
			
			

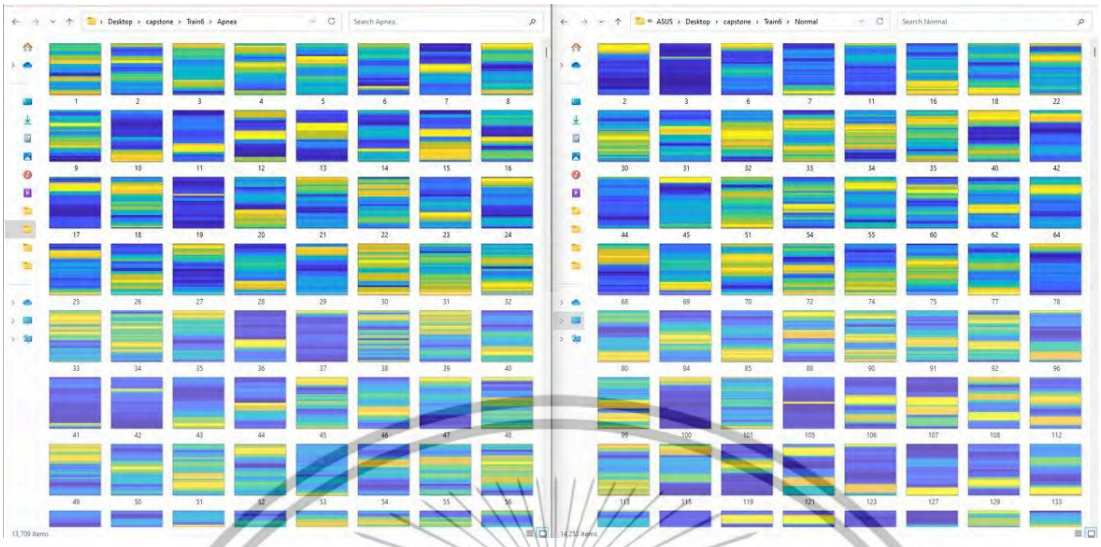


Figure 19 Two Folders

- PPG Time Domain Data:

Repeat the same step as the ECG signal, but peak detection will be different because there is only one peak in each cycle if the signal is smooth. 115 datasets from the previous section will undergo the following steps.

Step 1: Use the same inclusion numbers and criteria as the ECG part.

Step 2: Open the included dataset that is the edf file by using the 'edfread' builtin function in matlab as same as ECG, but the code in line 11 from table 1 will be changed to PPG= cell2mat(data.PPG);. The full text of coding from step 1 to 7 is located in the appendix E.

Step 3: Open xml file into MATLAB by using a builtin function called ‘readstruct’. Then create a table to note all periods of apnea, including obstructive sleep apnea, hypopnea, and unsure that is also a period of hypopnea as same as step 3 in the ECG part in table 2.

Step 4: Select only the period after the patient falls asleep as same as in the ECG part, but the name of the variable can be changed.

Step 5: Signal processing

5.1 Normalize PPG signals by using z-score as mentioned in line 3 in the table 17 below. Each z will depend on the value of amplitude x in each data point and z will be used to replace amplitude x. The value must be normalized into the unit of microseconds. As the frequency of this dataset is 256 Hz, the range of time will be divided by 256 and times 1000 before calculating HRV.

5.2 Remove noise in PPG signals by using both low-pass filter and high-pass filters in line 1 and line 2 respectively. Moreover, Gaussian and median filters should be applied to PPG to make the curve more smooth in line 4 and 5. Built-in functions of MATLAB can be used to filter and adjust smoothness of the signal. The result of filtering is shown in figure 20.

Table 17 Code for Filtering PPG

Line 1	PPGfil= lowpass(PPG,15,256);
Line 2	PPGfil= highpass(PPG,5,256);

Line 3	PPGfilz= (PPGfil- mean(PPGfil))/std(PPGfil); %z-score
Line 4	PPGfilz = imgaussfilt(PPGfilz);
Line 5	PPGfilz = medfilt1(PPGfilz);

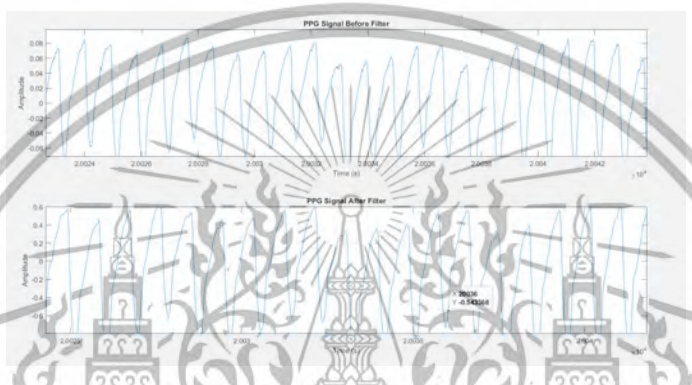
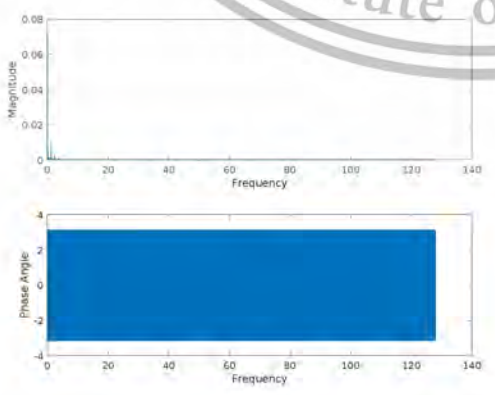
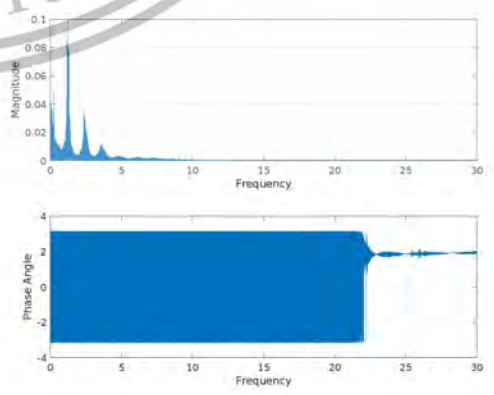


Figure 20 Filtered PPG Signal

Table 18 Filtered PPG Signal in Frequency Domain

dataset No.	Filtered PPG signal in frequency domain	
	Before Filtering	After Filtering
2		

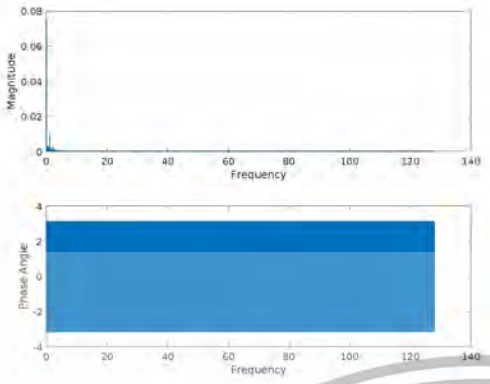
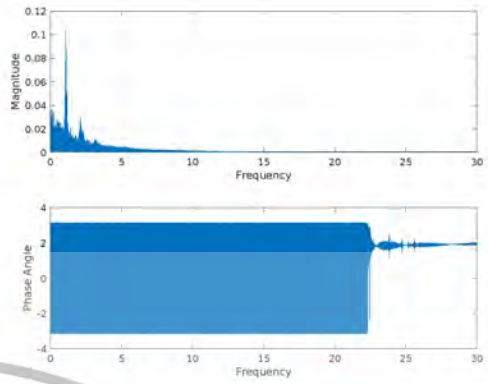
14		

Table 18 shows the frequency domain of the PPG signal by using the fourier transform "fft" in MATLAB. The figure on the left hand side is the ECG signal before passing the band pass filter, so there are frequencies from 0 Hz to over 120 Hz. And on the right hand side figure, the frequencies over 20 Hz are removed. And some of the frequencies below 5 Hz are lessened.

5.3 Peaks of PPG are detected by using fixed threshold and 'findpeaks' builtin function. The algorithm of peak detection is shown in figure 21. The overall code is in the appendix F.

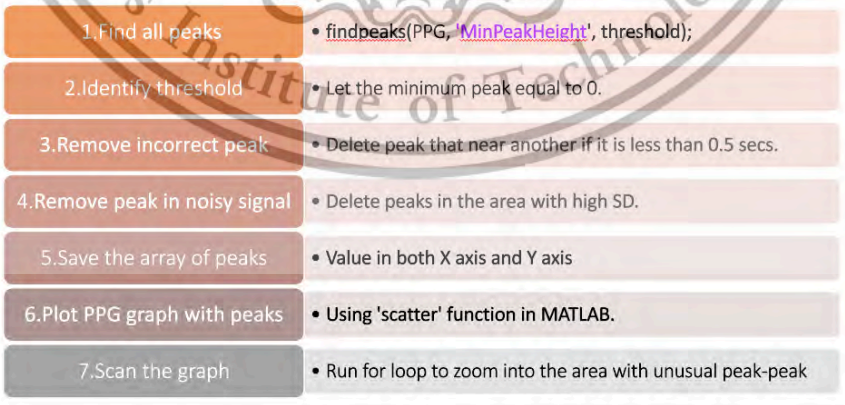
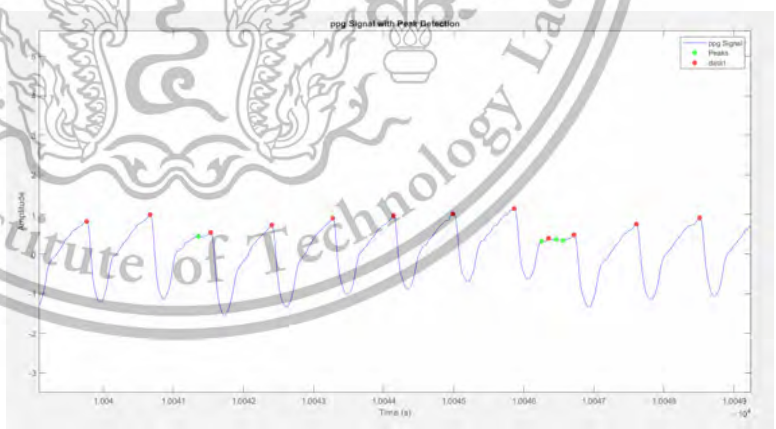
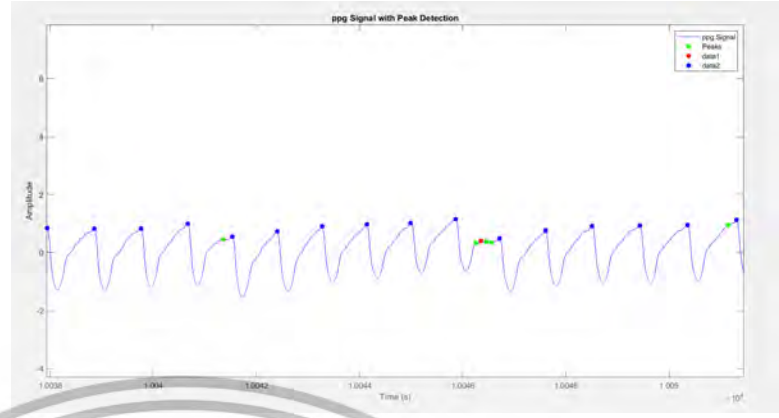


Figure 21 R Peak Detection Algorithm

According to figure 21, the first step can be done by using built-in functions that use the second derivative method to find peaks. In step 2, after plotting the PPG graph, the peak is normally more than 0, so the threshold can be fixed. In step 3, incorrect peaks that are too near to the previous point than 0.5 second will be deleted. This step will be done three times to ensure that the incorrect peaks are all removed as shown in table 19.

Table 19 PPG Incorrect Peaks Removal

Green dots are all peaks detected by 'findpeaks' before removing incorrect peaks.	
Red dots are all peaks detected after removing incorrect peaks for the first time.	

Blue dots are all peaks detected after removing incorrect peaks for the second time.	

In step 4, the window of 10 seconds will be slided over the signal and standard deviation of value in y axis (volume) in that area will be calculated. The area with high SD means that it is interfered with noisy signals as shown in figure 22. The points will be removed, so PRV in that area will be much higher than others so that it can be removed.

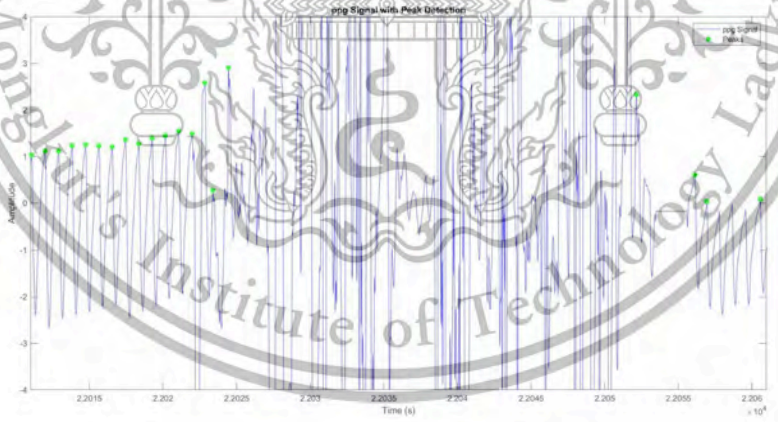


Figure 22 Remove Peaks In The Noisy Area

Step 6: Plot graphs of PPG. The x-axis is time while the y-axis is volume. And Also plot peak points by using the MATLAB function as shown in table 20. Line 1 is to plot time in x axis and PPG signal amplitude in y axis in blue color in figure 1. Line 2 and 3 is to plot R peak

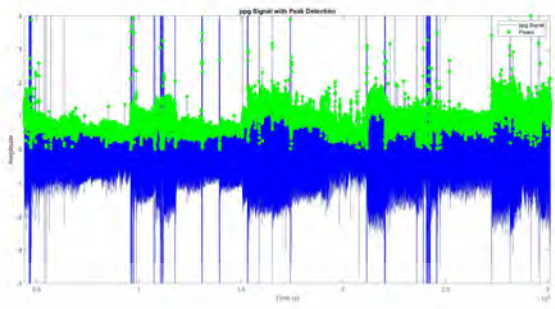
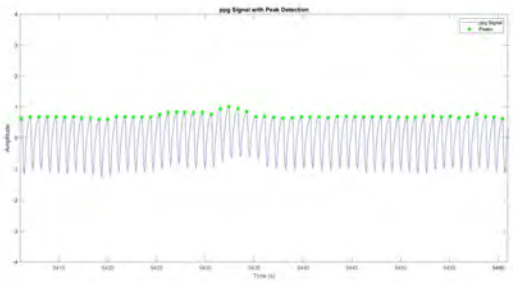
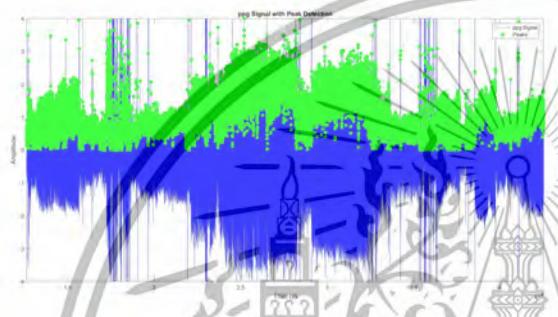
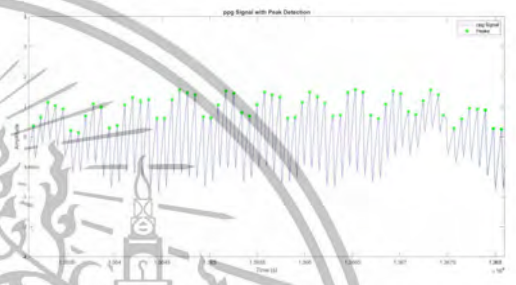
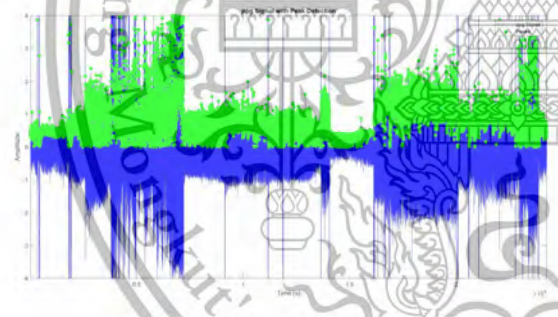
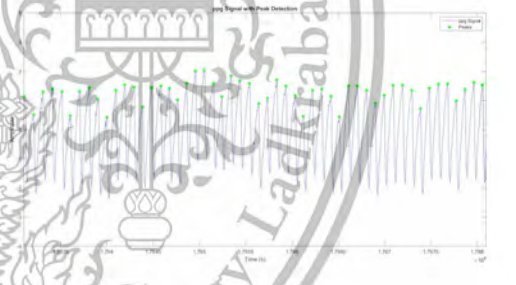
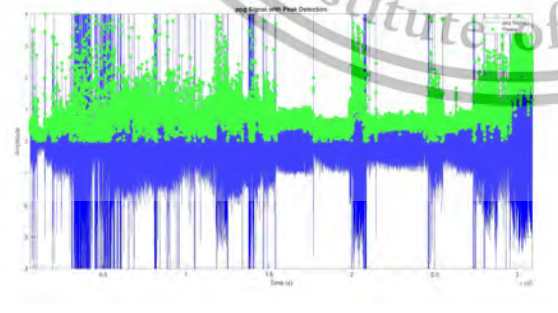
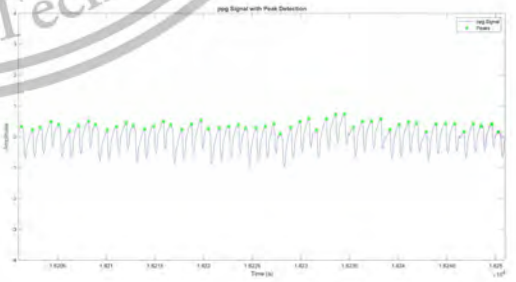
points. Line 4 is to move the axis to fit the graph. Line 5 to 7 is to label axis, legend, and title respectively. Line 9 is to zoom into the PPG signal as shown in the second column of table 21. The number in the axis can be adjusted then the figure will be saved into the folder within the same directory. If the code is automatically run, it is important to close the figure or use "hold off" before plotting a new figure. The overall code is located in the appendix F.

Table 20 PPG Plot Code

Line 1	figure(1),plot(PPGfiltime, PPGfilz, 'b');
Line 2	hold on;
Line 3	scatter(xout, yout, 'filled','g');
Line 4	axis tight;
Line 5	xlabel('Time (s)'), ylabel('Amplitude');
Line 6	legend('ppg Signal', 'Peaks');
Line 7	title('ppg Signal with Peak Detection');
Line 8	figure(1), axis([inclusiontime(1)+17600 inclusiontime(1)+17650 -1 1]);
Line 9	saveas(figure(1),"folder\"+inclusion(i,1)+"rpeak-3.jpg");
Line 10	close(figure(1))

Table 21 Peak Detection Result

No.	PPG Peak Detection	Zoom In

2		
14		
16		
46		

Step 7: Calculate the PRV in the form of RMSSD by using a for loop. The number of peaks that will be converted into a value of PRV will be specified based on the criteria of sleep apnea more than 10 seconds. 5-minutes PRV intervals are also calculated to compare with HRV from ECG.

Table 22 RMSSD Formula

Explanation	PRV Formula MATLAB Code for RMSSD
Time between peak to peak	peaktime= xyout(:,1); PRV1p= peaktime*1000; %unit= ms
Time between peak-peak and peak-peak (interval)	PRV2p=PRV1p(2)-PRV1p(1); for n=2:length(PRV1p)-1 PRV2p(end+1)= PRV1p(n+1)-PRV1p(n); end
Time difference between two RR interval then square (SSD)	PRV3p=PRV2p(2)-PRV2p(1); for m=2:length(PRV2p)-1 PRV3p(end+1)= PRV2p(m+1)-PRV2p(m); end PRV3p=PRV3p.^2;
Sum of SSD in a specific interval (inv = 10 and 300 seconds)	for le= 1: length(PRV3p)-inv+1 PRV4p(le)= sum(PRV3p(le:le+inv-1)); end

Square root of sum ssd divided by number of RR interval	rmssd= sqrt(HRV4e./(inv-1));
Time interval that is plotted in x-axis along with RMSSD in y-axis	Time = PRV1p((inv/2)+1 : length(PRV1p)-(inv/2)-1); Time = Time/1000; Time = Time'; %unit:sec
Three parameters (Time, RMSSD, and label) are recorded in a cell called "memat".	me= [Time ; rmssd ; zz]; memat{i}=me;

Step 8: Label the apnea period by using the table same as ECG in table 2. The coding is the same as ECG in step 8 as shown in table 11 in the previous section. The result is as in figure 23.

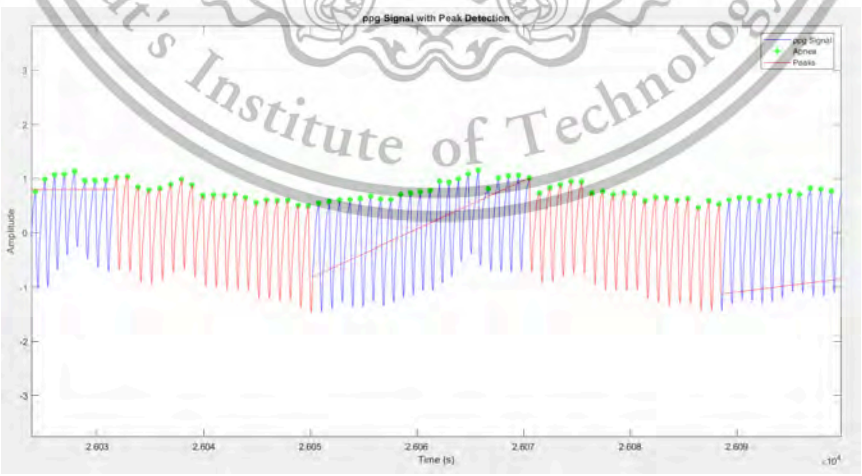
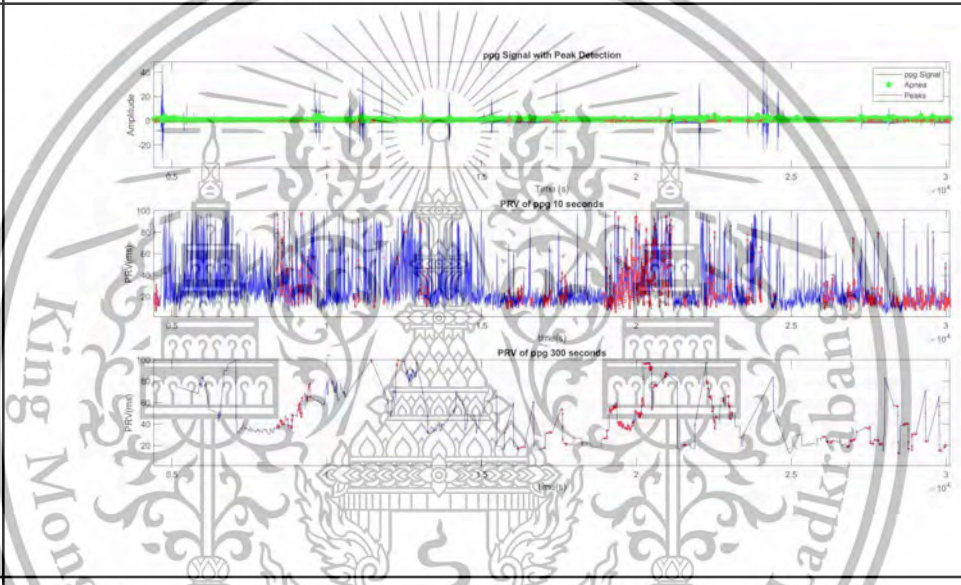
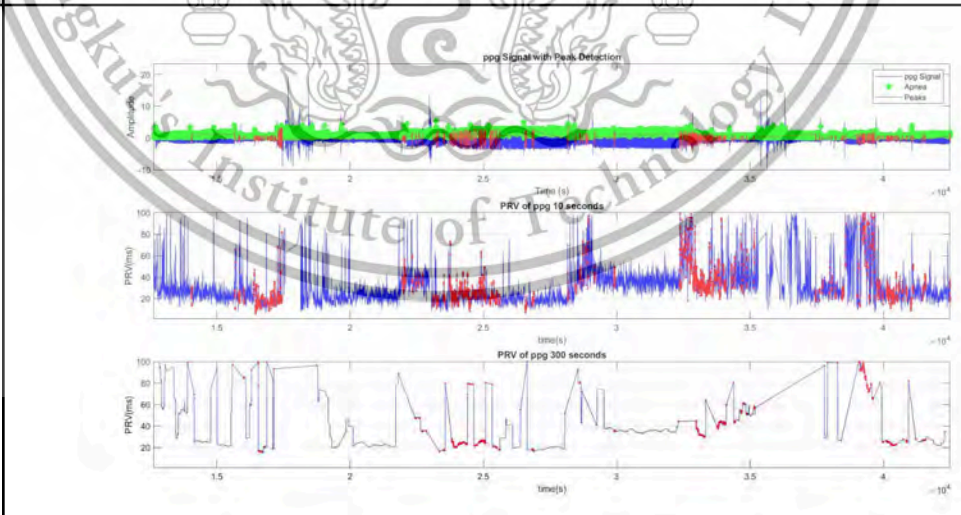
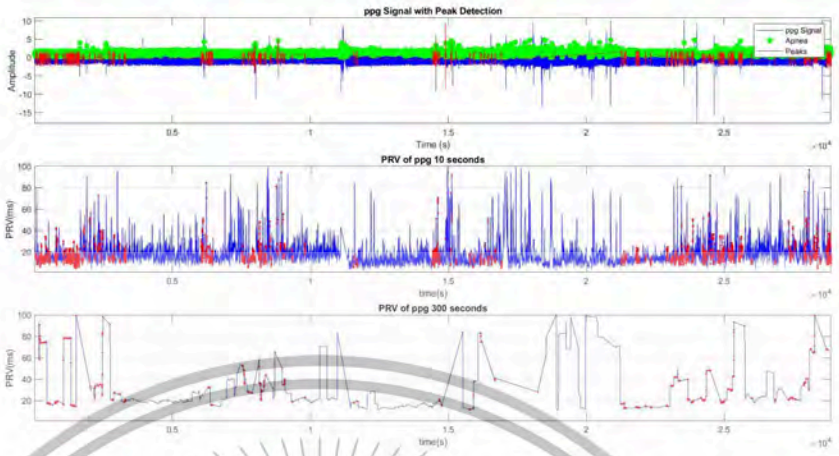


Figure 23 Normal and Apnea Label in PPG

Table 23 below shows the result example from step 7 and 8. The green dot means peak of PPG while the red dots or lines are apnea labels. The first subplot in each figure is a PPG signal with apnea label. And the second and third subplot is PRV with apnea label.

Table 23 PPG and PRV of Three datasets

dataset No.	PPG and PRV with Apnea Label
2	
14	

50	

In AI training, RMSSD with an interval of 10 seconds, which is the subgraph in the middle of every figure in table 23, will be used.

Step 8: From the previous step, the results are an array of 3 times the length of RMSSD per each dataset. The first row is the real time that matches RMSSD in the second row, and the third row is the label that 0 means normal and 1 means apnea. In this step, this array will be segmented into 60 seconds each, and be classified into two classes in two different folders separately as same as in step 8 of ECG in the previous part.

Step 9: Convert the segment arrays of PRV into jpeg images by using a builtin function called ‘image’ and the option of CDataMapping ‘scaled’ same as the ECG images as shown in figure 24. However, there are more images in PPG because the window size when sliding is only 20 seconds, while HRV images use 40 seconds for sliding.

Figure 24 PPG Images in Apnea and Normal Classes

3.3.2 Deep learning Training

Step 1: Open deepNetworkDesigner in MATLAB and load ResNet-50.

Step 2: Edit fully connect that is in the third block from the bottom to be two classes as shown in figure 25.

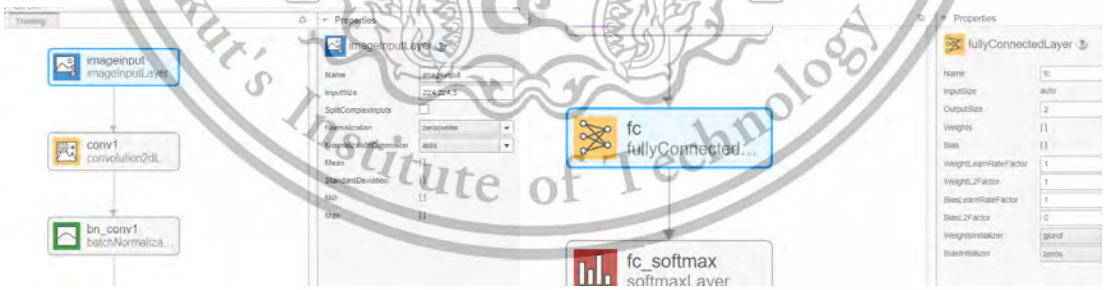


Figure 25 ResNet-50 Layers

Step 3: Import folders that contain two subfolders of images. Set the percentage of validation data to be 10% randomized that is splitted from training data as shown in figure 26.

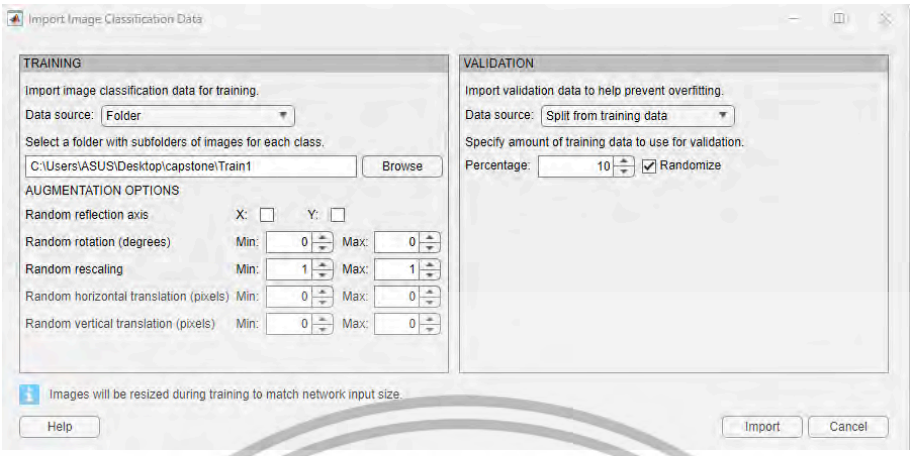


Figure 26 Import Image Classification Data

Images from HRV and PRV will be separately trained for two times each. In the first time of training, only 10,000 images were randomized to be trained. In the second time of HRV training, there were 14,253 images of normal class and 13,709 images of apnea class. In the second training of PRV, there are 28,199 images in normal class and 22,883 images from apnea class.

Table 24 Trained Data in HRV and PRV

HRV	PRV
	

After that, set the trained option as shown in figure 27. In the first-time training, initial rate was 0.001 and max epochs 30, but in the second training initial rate is decreased to 0.0001 and max epochs is 5 because if the max epoch is too high, the model will become overfit.

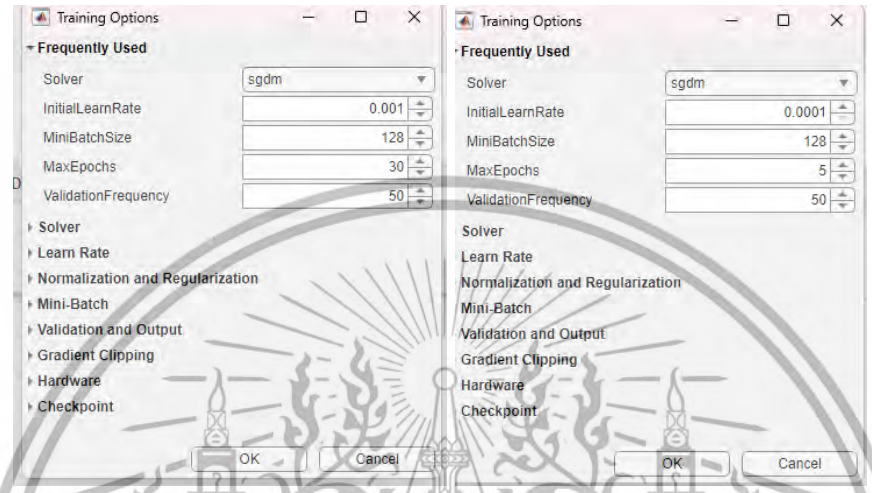
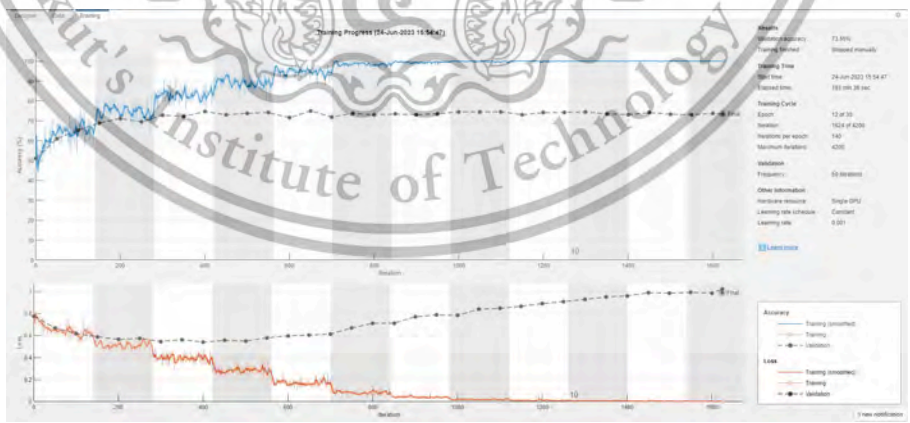
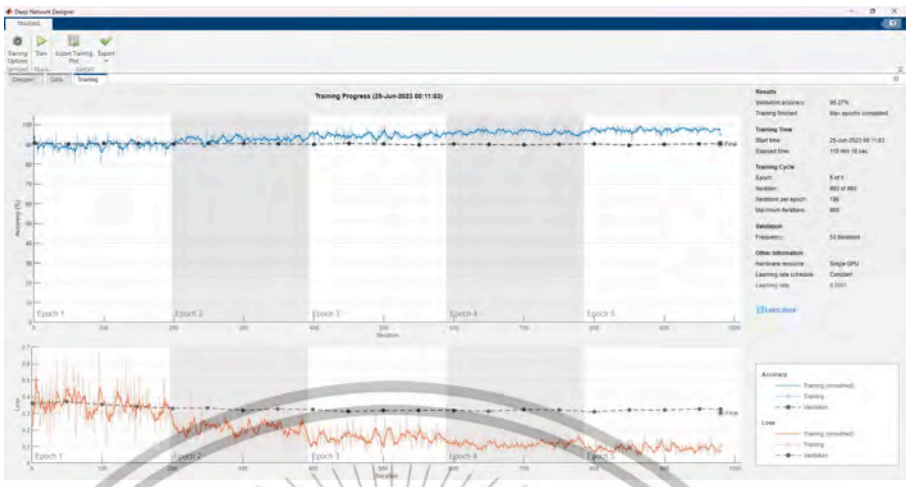


Figure 27 Training Option of The First and the Second Train

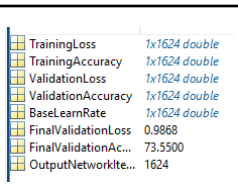
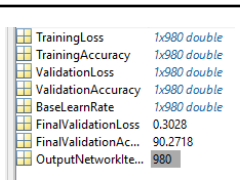
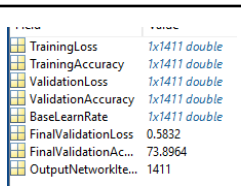
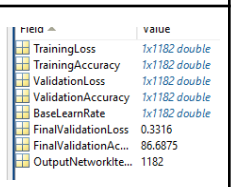
Table 25 Training Progress

	HRV Training Progress
Train 1	

Train 2	

The final validation accuracy and final validation loss of train 1 HRV, train 2 HRV, train 1 PRV, and train 2 PRV is shown below in table 26. The coding that can be used as an alternative of deepNetworkDesigner is in appendix G.

Table 26 TrainInfoStruct of Four Training

	HRV Train 1	**HRV Train 2**	**PRV Train 1**	**PRV Train 2**
Final validation accuracy	73.55%	90.27%	73.90%	86.69%
Final Validation Loss	0.9868 (Overfit)	0.3028	0.5832	0.3316
TrainInfoStruct				

3.3.3 Testing

3.3.3.1 Testing part 1

Prepare test datasets that are similar to the trained images. HRV or PRV of apnea class will be picked up from any position of the array that label apnea, but in normal class, HRV or PRV value must be from the area that label is all equal to zero. Then coding for the test by importing images into the trained model 2. Finally, create the confusion matrix. The array of true class and the predicted class must be used to generate the matrix using a builtin function called 'confusionchart' as in the third row of table 27.

Table 27 Test Trained Network and Plot Confusion Matrix

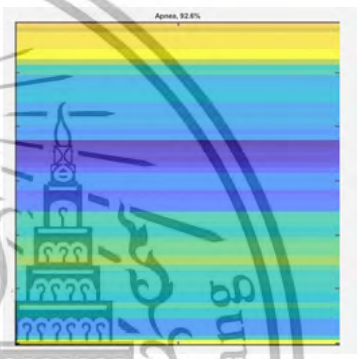
Explanation	**MATLAB Code**

Classify the test image using the trained network. YPred is the predicted class: Normal or Apnea.	f= imread("Test/test"+setnum+"/Apnea"+ i +".jpg");
	[YPred,probs] = classify(trainedNetwork_1,f);
	figure(i),imshow(f);
	label = YPred;
	title(string(label) + ", " + num2str(100*max(probs),3) + "%");

Plot the confusion matrix	cm= confusionchart(truelabel, predictedlabel) saveas(cm,"Test\cm"+setnum+".jpeg");

Table 28 below shows the example of output segments in true positive, true negative, false positive and false negative from the trained neural network of HRV.

Table 28 Example of Output

	True Apnea	**True Normal**
Predicted Apnea	True Positive 	False Positive 
Predicted Normal	False Negative 	True Negative 

3.3.3.2 Testing part 2

Prepare test datasets that apnea and normal condition are the same to test the whole signal. The window size is 60, but the window sliding size will be 30. Therefore, there will be

some areas that are overlapping. There aren't any areas where all labels are apnea in 60 seconds, so only when the sum of apnea labels in that area is over 30, it will be classified as apnea.

Table 29 Changed Code For Apnea Classification for Testing

Change only this part 3
from table 13.
	for r2= 1: 30 :length(rmssde)-59
	if sum(z(r2:r2+59))>=30
	z(r2)=1;
	r_one(r2,:)=rmssde(r2:r2+59);
	end
	end

Table 30 Number of Images

Part 2	HRV Normal	HRV Apnea	PRV Normal	PRV Apnea	Total images
No. 962	688	169	601	27	1485
No. 976	1054	94	783	41	1972
No. 977	547	22	494	24	1087
No. 979	414	49	311	59	833
No. 994	611	151	78	64	904
Total images	3314	485	2267	215	6281

After that, convert HRV and PRV to images as in the training part, but it should be saved in a different folder. After testing, the confusion matrix is generated. However, it is still not the final result. The areas that are overlapped will be used to decrease the sensitivity of the result. The code is shown in table 30 below. When the prediction of a model in two overlapping areas is not 1, it will be rounded down to 0.

Table 31 Result Processing

Explanation	**MATLAB Code**

Part 1: Ground truth matrix will be generated from the data that is saved from 'memat' that contains 3 rows: time, RMSSD, and labels.	load('C:\Users\ASUS\Desktop\capstone\Train3\metest.mat')
	mm = setnum;
	me= memat{mm};
	rmssde= me(2,:);
	ze= me(3,:);
	time=me(1,:);

Part 2: Prediction matrix will be created to remove the overlapping area.
-label matrices have to be converted from 60 x 1 into 60 x 60 so that it will match the time array.
	timeall60=[timeap timenor];
	labelall60=[];
	for lb= 1:60
	labelall60(lb,:)= labelall(1,:);
	end
	
	labelall1= labelall60(:);

-Then both time and label array will be converted back to 3600 times the number of matrices x 1 before being sorted.	timeall1= timeall60(:);
	x= [timeall1 labelall1];
	[~,inx]=sort(x(:,1));
	xsort= x(inx, :);

Part 3: The condition is to check whether the areas are overlapping or not. If they are overlapping and the predicted label is the same, then nothing will change. However, if the label is different, it will be rounded down to zero. One of the overlapping numbers will be recorded in matrix ll before being used to remove the overlapping time and label in the end.	ll=[];
	avelabel=[];
	for num2= 1: length(xsort)-1
	if xsort(num2,1)== xsort(num2+1,1)
	ll(num2+1)=num2+1;
	if xsort(num2,2)== xsort(num2+1,2)
	avelabel(num2)= xsort(num2+1,2);
	%(xsort(num2,2)+xsort(num2+1,2))/2;
	else
	avelabel(num2)= 0;
	end
	else
	avelabel(num2)= xsort(num2,2);
	avelabel(num2+1)= xsort(num2,2);
	end
	end
	xsort(:,2)= avelabel(:,:)';
	ll=nonzeros(ll);

	xsort(ll,:)=[];

Part 4: The matrix of time and label will be plotted along with the groundtruth.
 	
Another figure that zooms into a specific area of the prediction in the signals will be provided in the next chapter.
	figure(5),plot(time, ze)
	hold on;
	plot(xsort(:,1),xsort(:,2),'r')
	hold off;
	axis([xsort(1,1) xsort(end,1) -2 2]);
	legend('Ground Truth','Prediction')
	xlabel('Time (s)'), ylabel('Apnea');
	title('Apnea Prediction');
	saveas(figure(5),"Train6/"+setnum+"-3binary.jpeg");

figure 28 below shows the result of prediction by HRV and PRV of dataset number 962. The confusion matrix and other results will be provided in Chapter 4.

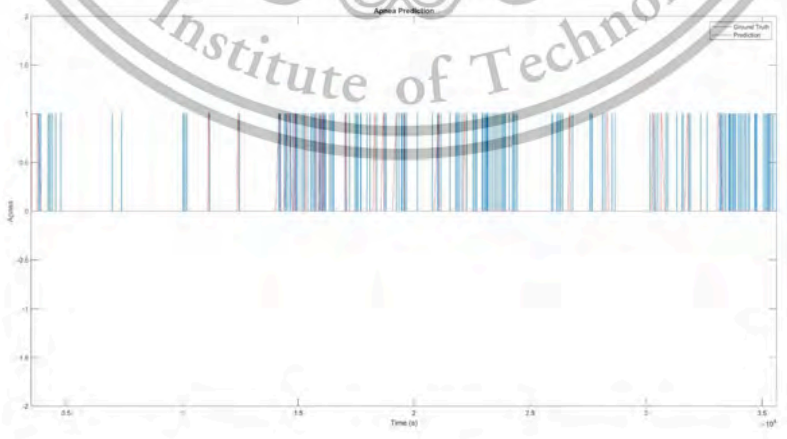


Figure 28 Overall Apnea Prediction Result

Chapter 4

Experimental Result

4.1 Introduction

The result of the confusion matrix will be analyzed and presented by using the following formula in table 32. TP is true positive, which means that the model predicts apnea which is the true result that matches the ground truth. TN is true negative, which means that the prediction is normal and it is true. FN is false negative, which means that the model predicts normal, but the truth is apnea. Finally, FP is false positive, which means that the model predicts apnea, but the true value is normal.

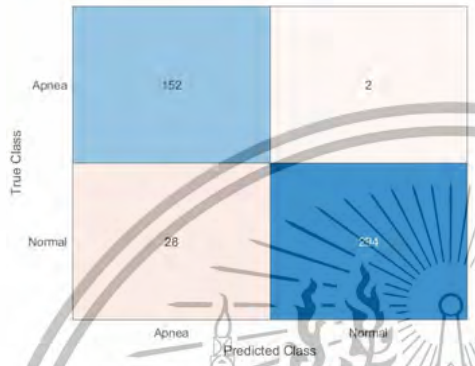
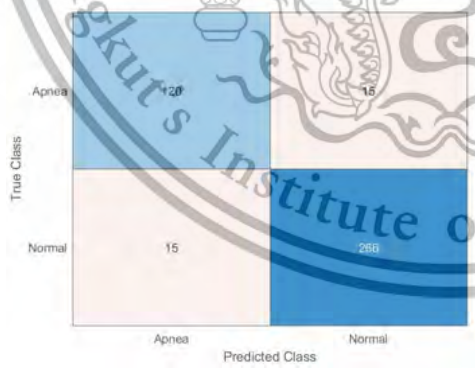
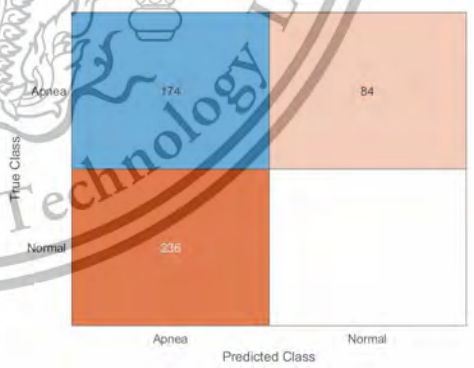
Table 32 Formula of Efficiency Parameter

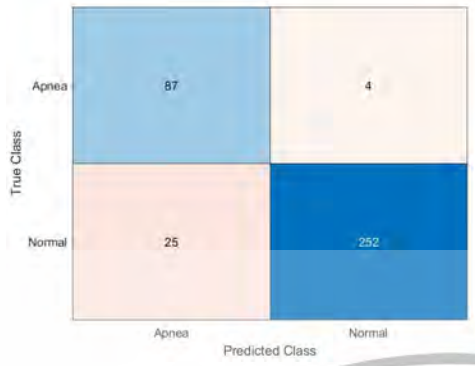
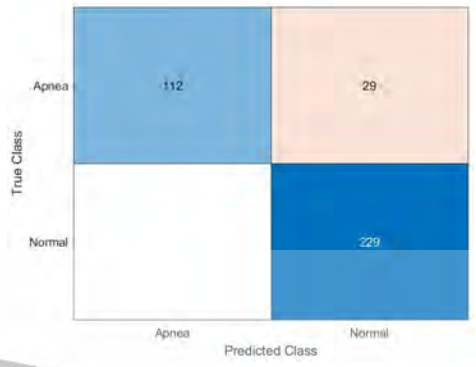
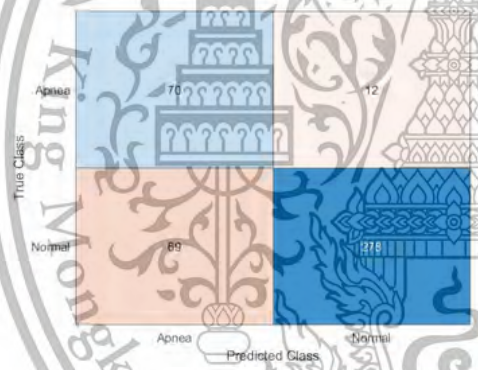
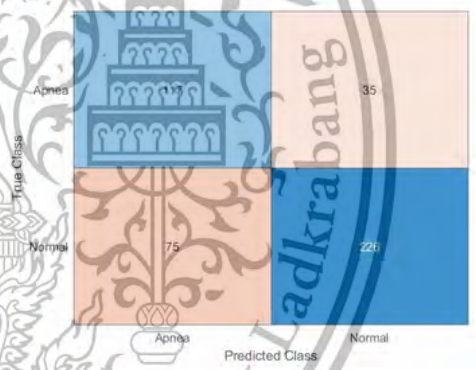
Accuracy = $\frac{TP+TN}{TP+TN+FP+FN}$	Sensitivity= $\frac{TP}{TP+FN}$	Specificity= $\frac{TN}{TN+FP}$
Cohen's Kappa = $\frac{2*(TP*TN-FN*FP)}{(TP+FP)*(FP+TN)+(TP+FN)*(FN+TN)}$		

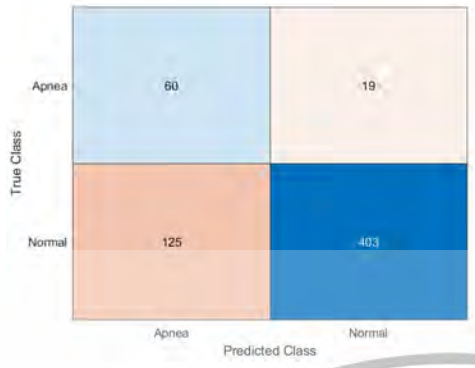
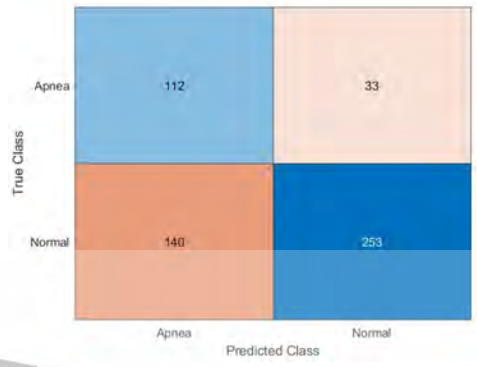
4.2 Result Part 1

The confusion matrix of the result in part 1 is shown in the table 33 below.

Table 33 Confusion Matrix of Five Datasets Part 1

dataset No.	**Prediction from HRV**	**Prediction from PRV**
939	 Accuracy = 93.70% Sensitivity = 98.70% Specificity = 91.30%	 Accuracy = 92.11% Sensitivity = 83.78% Specificity = 100.00%
943	 Accuracy = 92.79% Sensitivity = 88.88% Specificity = 94.66%	 Accuracy = 35.22% Sensitivity = 67.44% Specificity = 0%

951	 Accuracy = 92.12% Sensitivity = 95.60% Specificity = 90.97%	 Accuracy = 92.16% Sensitivity = 79.43% Specificity = 100.00%
962	 Accuracy = 81.11% Sensitivity = 85.37% Specificity = 80.12%	 Accuracy = 75.71% Sensitivity = 76.97% Specificity = 75.08%

967	 Accuracy = 76.28% Sensitivity = 75.95% Specificity = 76.33%	 Accuracy = 67.84% Sensitivity = 77.24% Specificity = 64.38%
Mean	Mean Accuracy = 87.20% Mean Sensitivity = 88.90% Mean Specificity = 86.68%	Mean Accuracy = 72.61% Mean Sensitivity = 76.97% Mean Specificity = 67.89%

The graph in figure 29 concludes the mean accuracy, sensitivity, and specificity of the result from table 33. In part 1, it will mainly focus on the specificity. The reason will be discussed in the next chapter. Taking paired t test, p value is 0.02458 for one-tailed test with 0.05 significance level. It means that there are significant differences between HRV and PRV, and HRV is better for detecting normal signals.

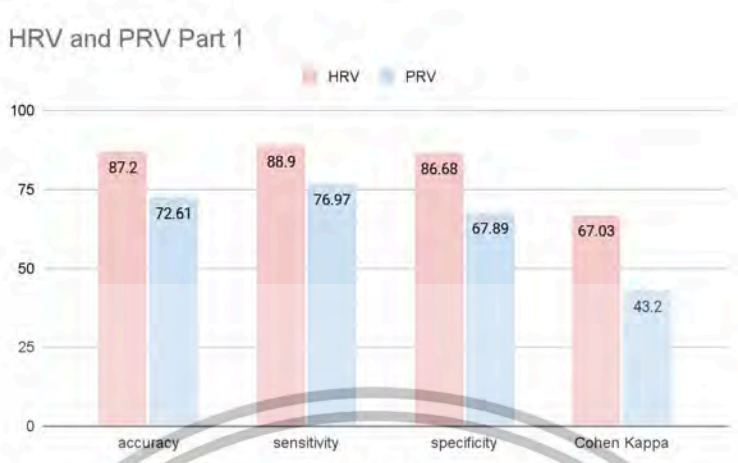
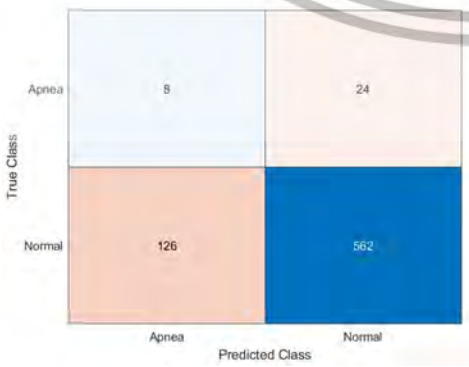


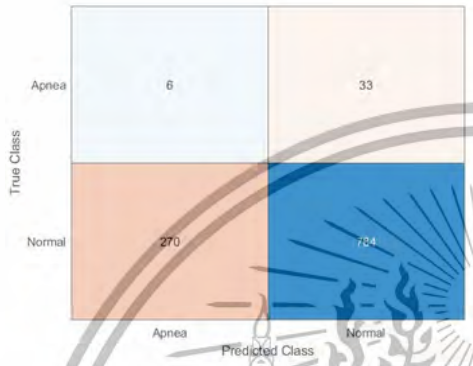
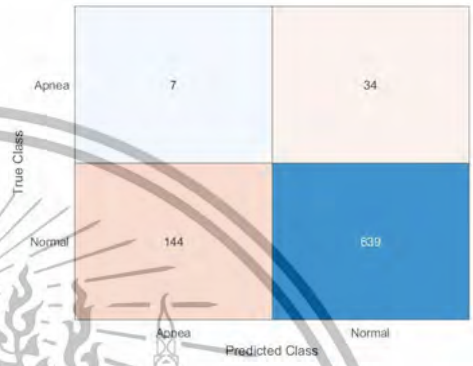
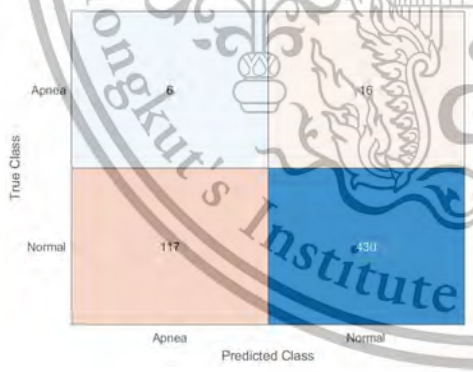
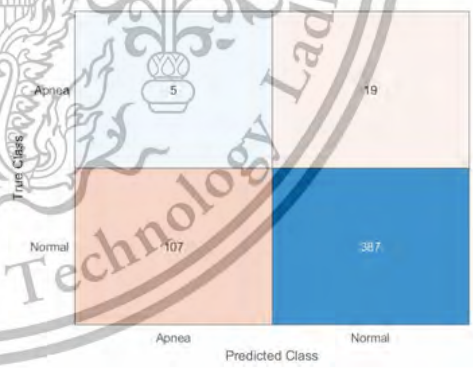
Figure 29 Data Analysis Part 1

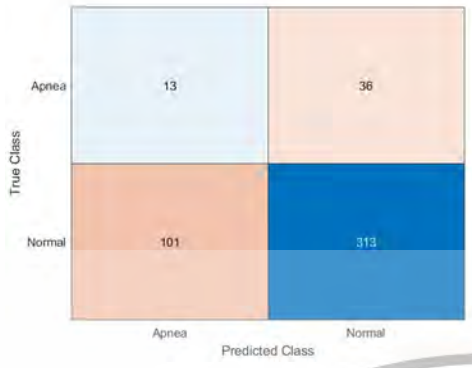
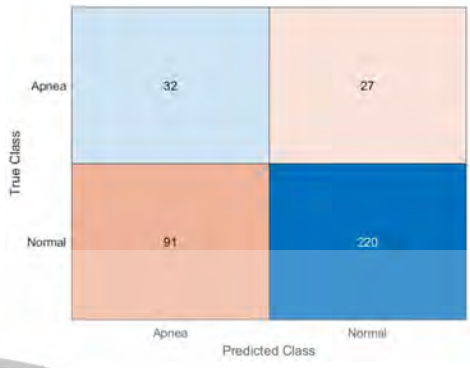
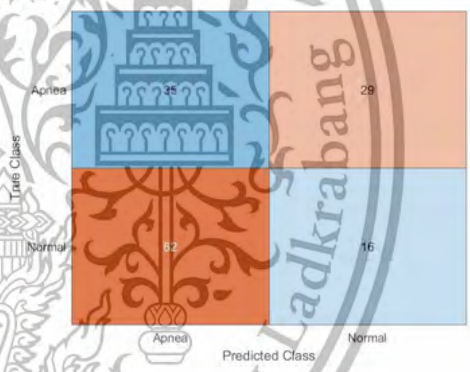
4.2 Result Part 2

The following table 34 shows the confusion matrix of the result in part 2 that the images of the input result are generated by different methods as mentioned in the testing part of the previous chapter.

Table 34 Confusion Matrix of Two Datasets Part 2

dataset No.	**Prediction from HRV**	**Prediction from PRV**
962		

	Accuracy = 79.17% Sensitivity = 25.00% Specificity = 81.69%	Accuracy = 84.55% Sensitivity = 11.11% Specificity = 87.85%
967	 Accuracy = 72.28% Sensitivity = 15.38% Specificity = 74.38%	 Accuracy = 75.23% Sensitivity = 17.07% Specificity = 79.20%
977	 Accuracy = 76.63% Sensitivity = 27.27% Specificity = 78.61%	 Accuracy = 75.68% Sensitivity = 20.83% Specificity = 78.34%

979	 Accuracy = 70.41% Sensitivity = 26.53% Specificity = 75.60%	 Accuracy = 68.11% Sensitivity = 54.24% Specificity = 70.74%
994	 Accuracy = 54.99% Sensitivity = 34.44% Specificity = 60.07%	 Accuracy = 35.92% Sensitivity = 54.69% Specificity = 20.51%
Mean	Mean Accuracy = 70.69% Mean Sensitivity = 25.73% Mean Specificity = 74.07%	Mean Accuracy = 67.90% Mean Sensitivity = 31.59% Mean Specificity = 67.33%

The graph in figure 30 concludes the mean accuracy, sensitivity, and specificity of the result from table 34. Paired T Test is used to determine significance difference between each parameter. Results of the paired-t test indicated that there is no significant difference between the accuracy, sensitivity, and specificity. The p-value is equal to 0.275, 0.249, and 0.234, respectively for one-tailed tests with 0.05 significance level.

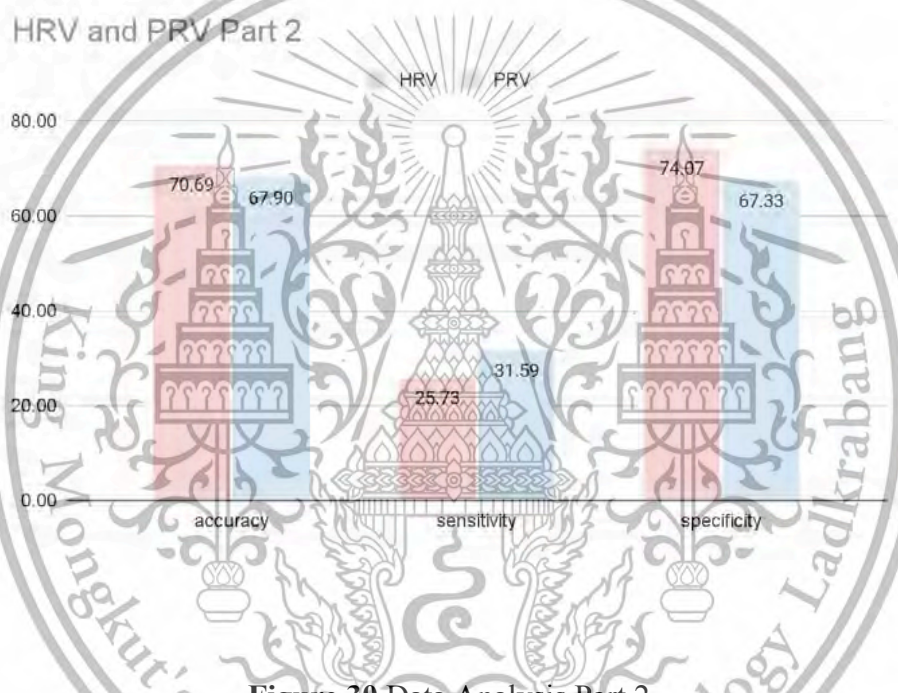
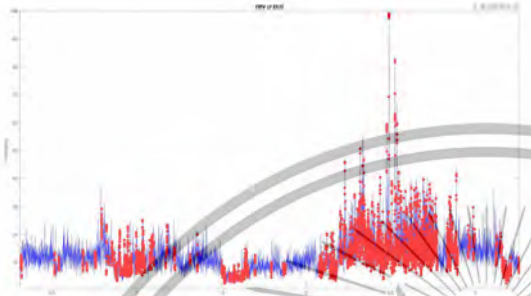
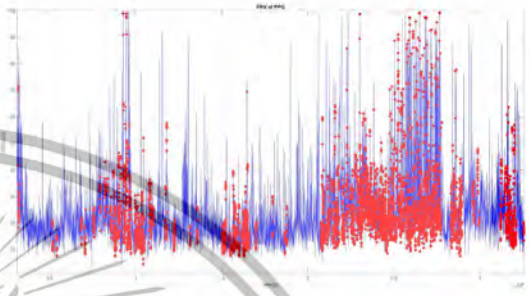
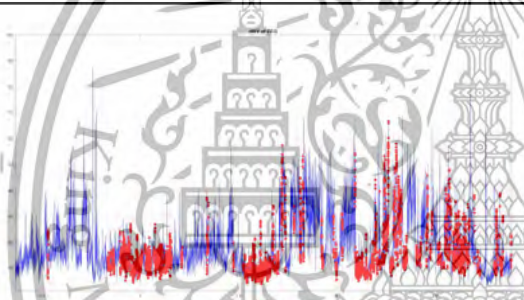
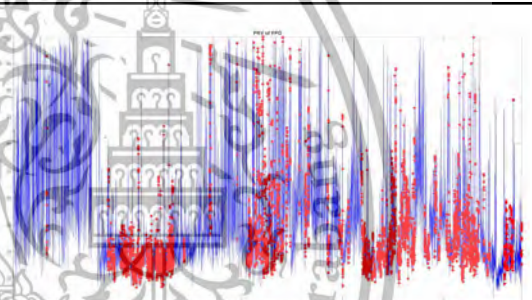
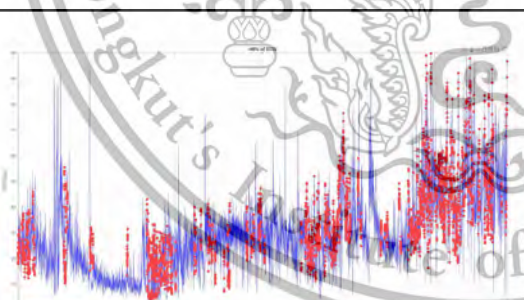
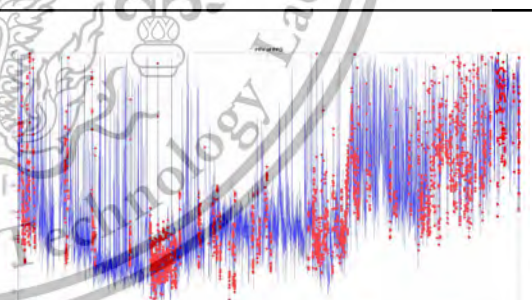
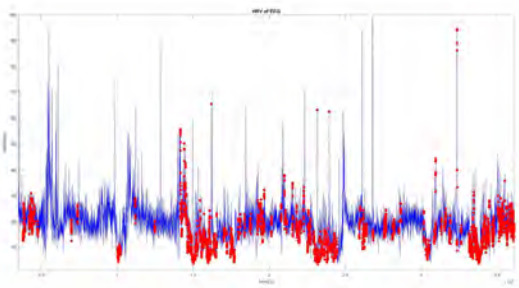
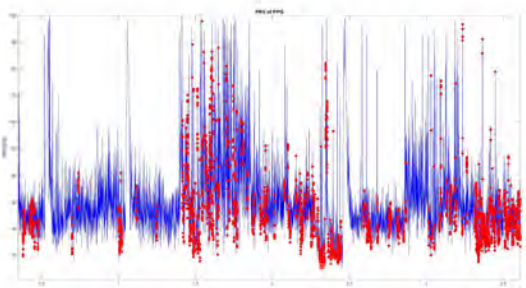
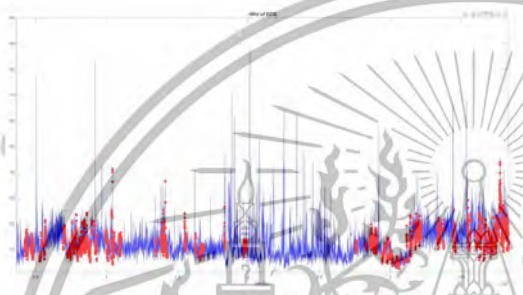
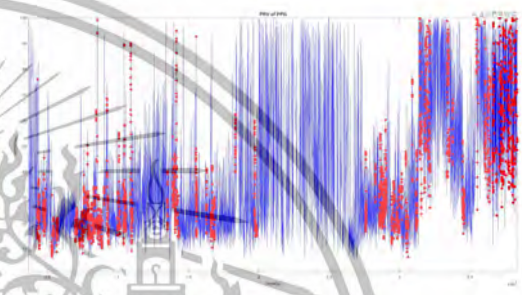
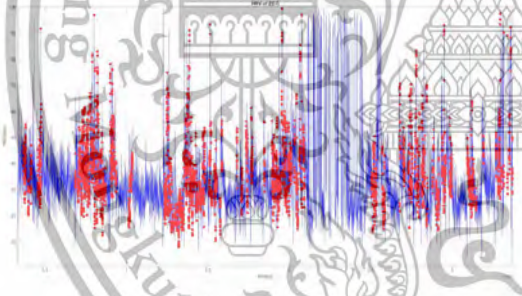
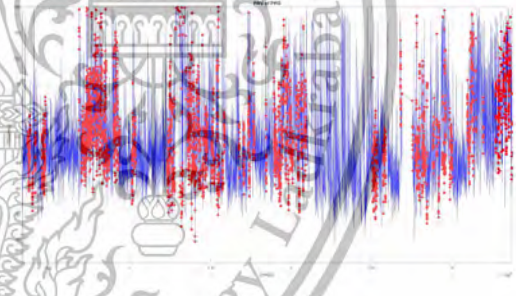
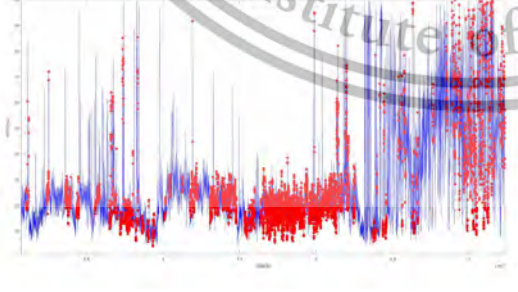
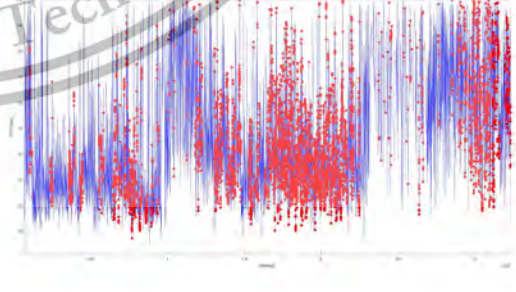


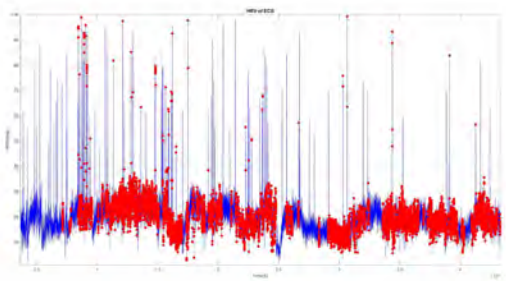
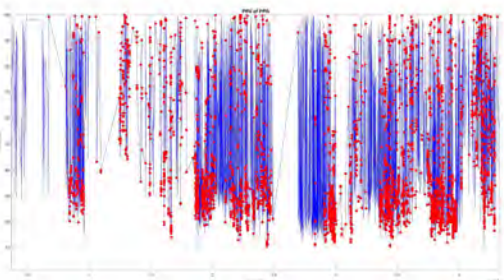
Figure 30 Data Analysis Part 2

The confusion matrix results in table 33 and table 34 will be compared with the following HRV and PRV graphs in the following table 35 in the discussion of the next chapter. Both table 33 and table 34 can use the same HRV and PRV graphs because they are generated before the segmentation and classification process. HRV and PRV below is the plot after removing the unusual value.

Table 35 HRV and PRV After Noise Removal

dataset No.	**ECG and HRV**	**PPG and PRV**
939		
943		
951		

962		
967		
977		
979		

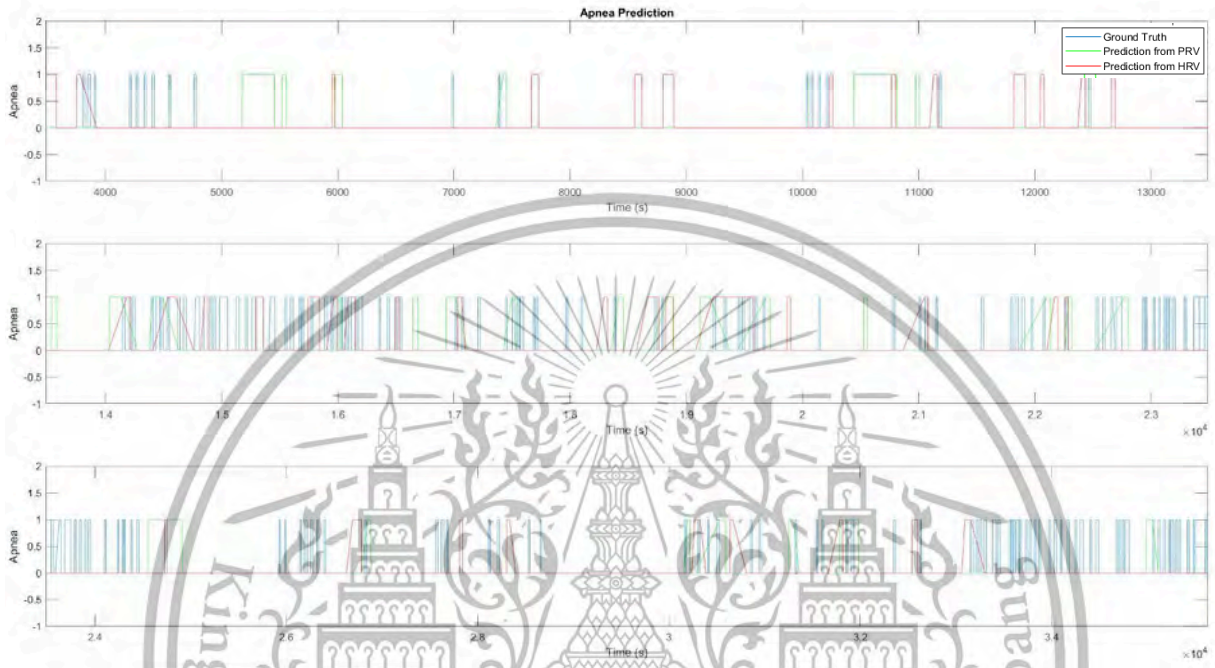
994		

Result Processing

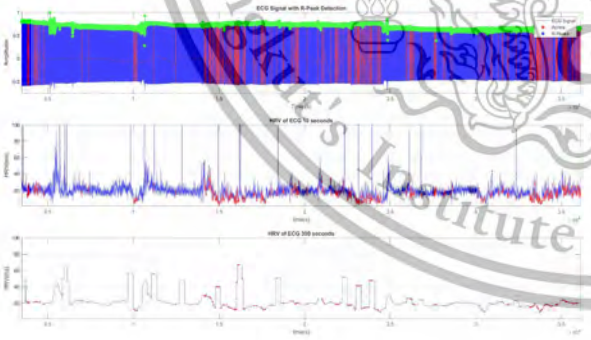
The following table 36, 37, 38, 39, and 40 shows ECG with HRV and PPG with PRV along with the signal after removing the overlapping sequences of 5 datasets: 962, 967, 977, 979, and 994 respectively. The result will be discussed in the next section.

Table 36 Prediction in Overall Signals of Dataset No.962

Dataset No.962- Signal with 9.0644 hours = 543.8646 minutes = 32632 seconds



ECG and HRV

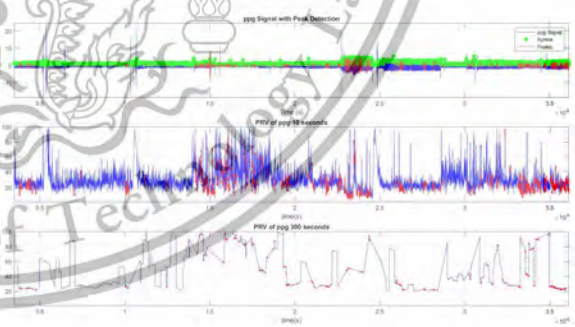


Number of HRV = 32053 values

Mean HRV = 19.9597

SD of HRV = 9.3860

PPG and PRV



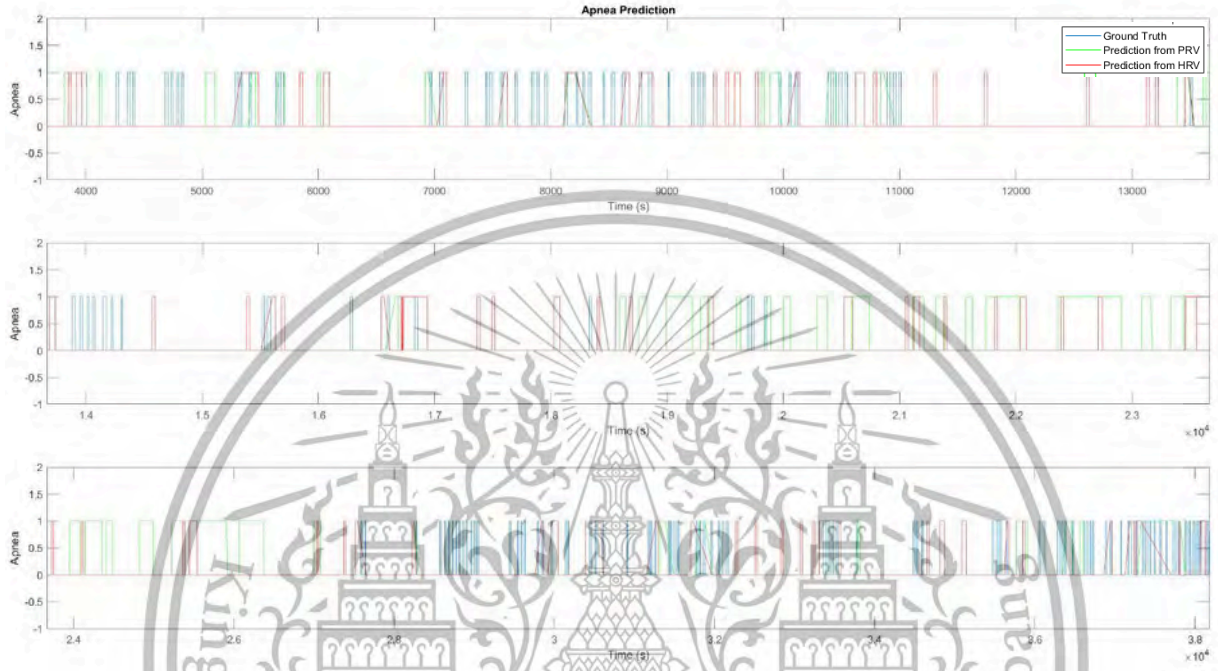
Number of PRV = 29153 values

Mean PRV = 29.4265

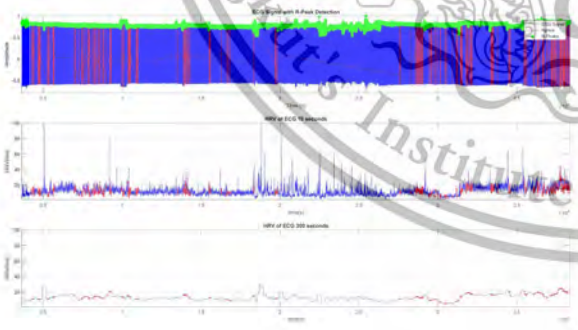
SD of PRV = 12.2831

Table 37 Prediction in Overall Signals of Dataset No.967

Dataset No.967



ECG and HRV



PPG and PRV

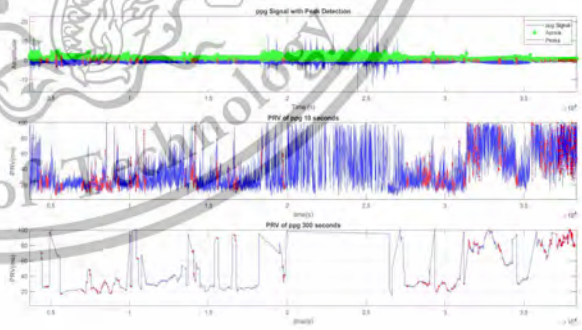
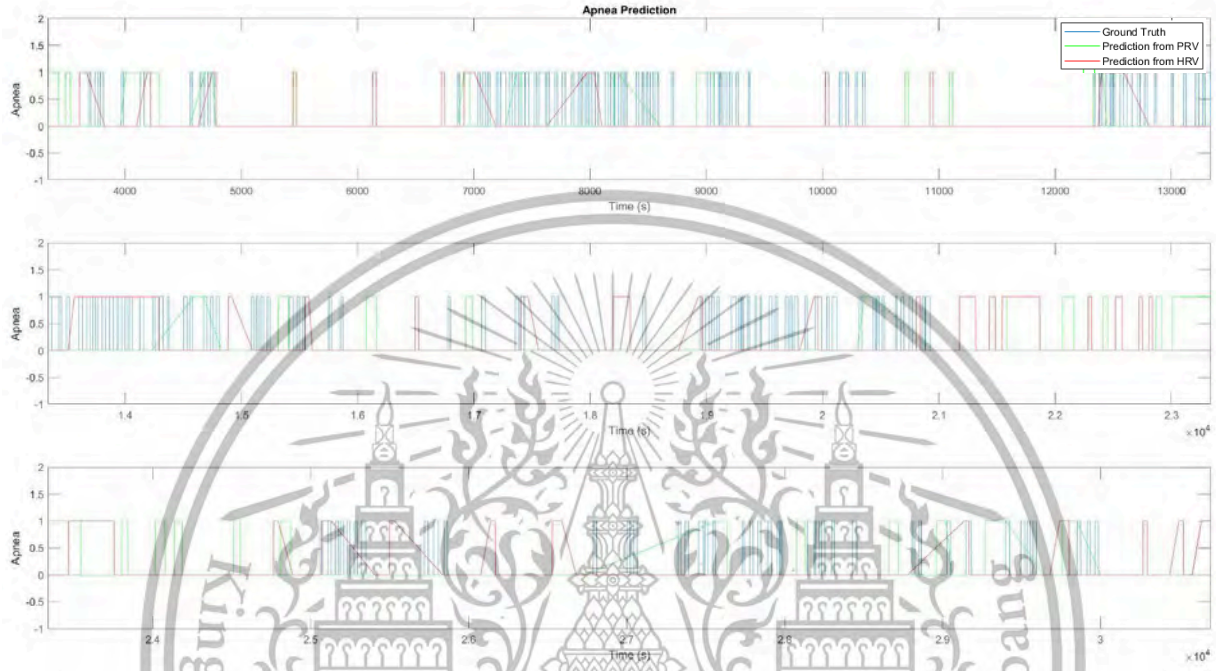
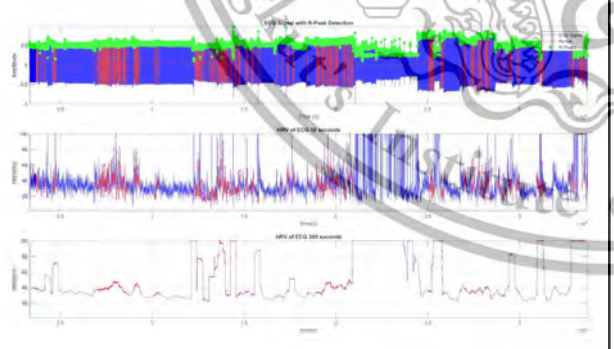


Table 38 Prediction in Overall Signals of Dataset No.977

Dataset No.977



ECG and HRV



PPG and PRV

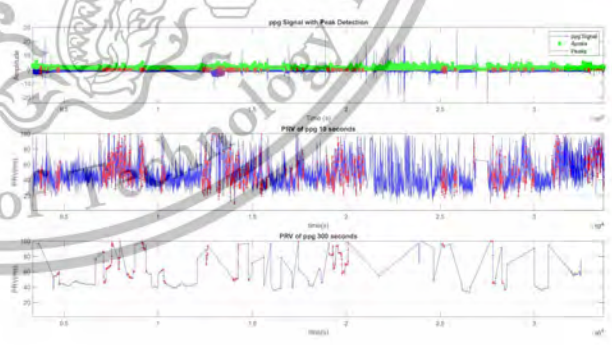
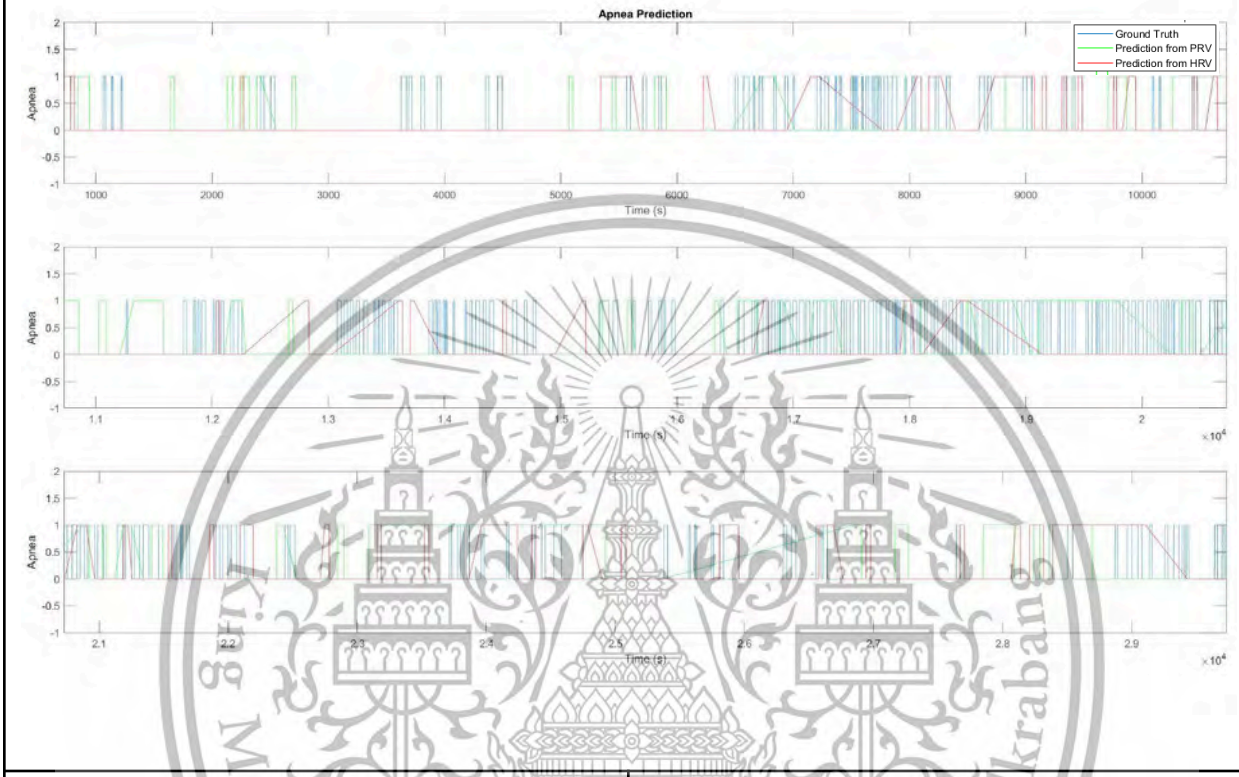
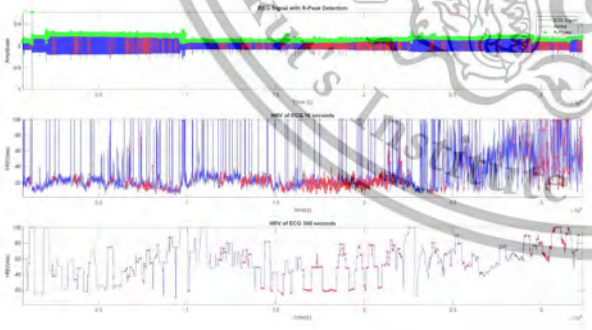


Table 39 Prediction in Overall Signals of Dataset No.979

Dataset No.979



ECG and HRV



PPG and PRV

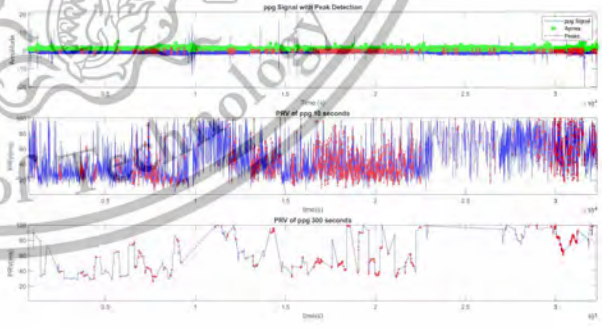
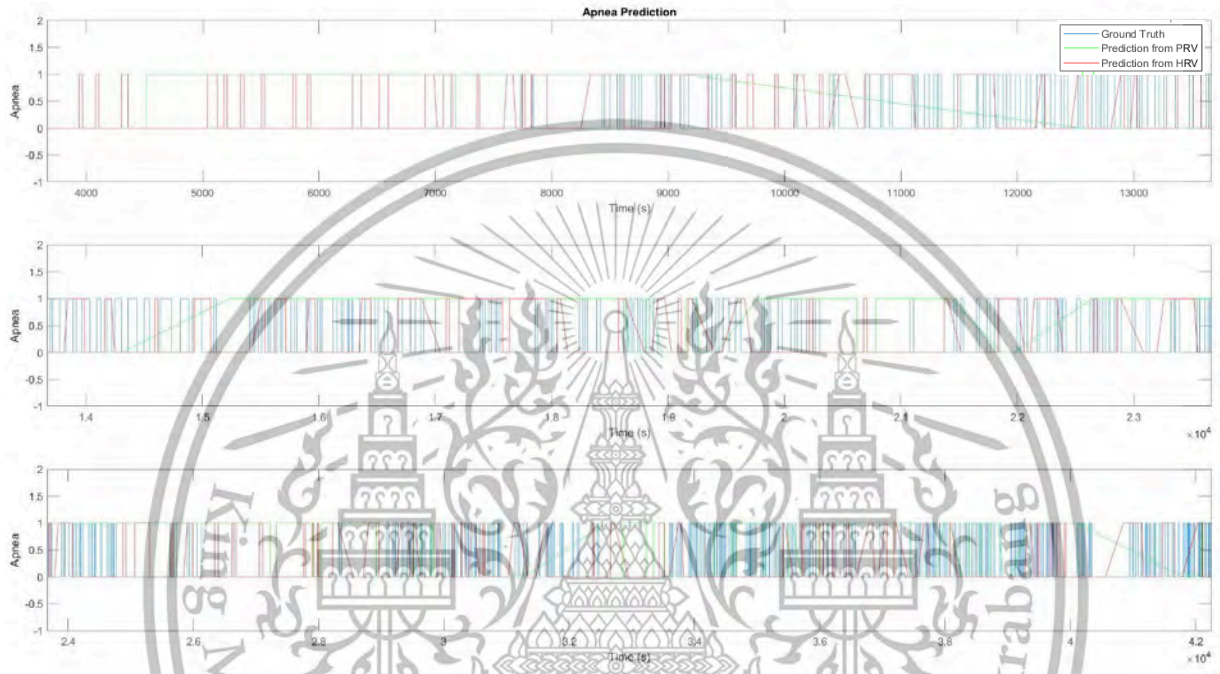
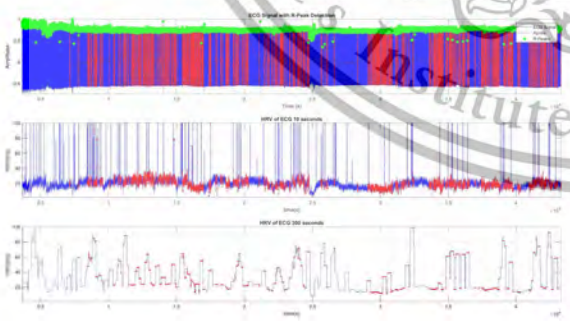


Table 40 Prediction in Overall Signals of Dataset No.994

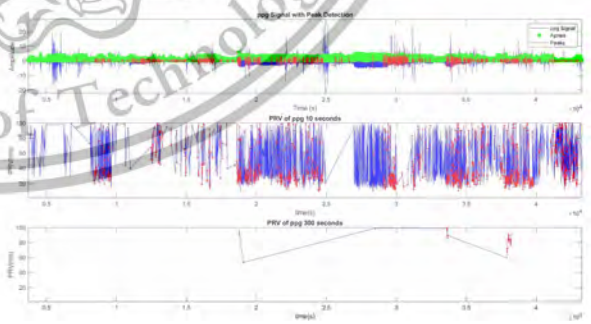
Dataset No.994



ECG and HRV



PPG and PRV



Chapter 5

Conclusion

5.1 Introduction

This chapter presents discussion and conclusion of the result in chapter 4 along with all processes in chapter 3 methodology and material. One point to be discussed is the effectiveness of using HRV or PRV to classify normal and apnea part of the signals, and the reason for the occurrence. And the main discussion point is to compare the efficiency of apnea detection by using HRV and PRV, so the graph of those two parameters will be compared as well.

5.2 Discussion and conclusion

According to part 1 of the result in table 32, the accuracy, sensitivity, and specificity of five datasets each from apnea classification by HRV and PRV are calculated. The efficiency of classification will depend on the patterns of HRV or PRV value. The main objective of test part 1 is to see differences between the tested sets that have been used for training and have never been used for training. The first three datasets in table 32 have been used in the second time of network training, so the accuracy, sensitivity, and specificity is much higher than the rest. This shows the high validation accuracy of the images that fit the models. However, other two datasets that have never been used for training have only about 80% accuracy although the

preparation method is the same. Therefore, there are only some patterns recognised by the CNN, but the model is still not good enough for apnea classification. The solution is to increase the number of datasets and training images by decreasing the window sliding size of apnea and normal class in the future work.

Moreover, comparing between the same datasets using different signals (ECG and PPG) for classification, datasets number 939 and 951 have similar accuracy. The specificity of those two datasets are also 100%. However, datasets number 943 of PRV have very low accuracy, sensitivity, and specificity. It might be because of overfitting in the trained network model of PRV, and dataset number 943 has a different normal pattern of PRV compared to most images that were used for training. The solution is to filter out more noises from all PRV data.

From figure 29, all efficiency parameters of HRV are much higher than that of PRV. In part 1, the most important parameter is specificity that focuses on true negative. The results from the previous chapter means that HRV is more significant when it is used to classify normal signals. The Cohen Kappa of HRV is 67.03% that means substantial agreement while that of PRV is 43.2% that means moderate agreement. So, HRV is better. One of the important reasons is that more noises can be found in PPG which make the peak detection more difficult and lead to error in PRV calculation. Although the Cohen Kappa value in both signals is acceptable, the result in part one is not practical for detecting apnea because the apnea period is unknown.

The part 2 of testing is more practical for testing the real signal, but the accuracy and sensitivity of apnea class will decrease. The sensitivity which is the true positive value over the

sum of true positive and false negative is very low. The sensitivity from HRV is 25.73% while that of PRV is 31.59%. It means that less than one-third of the apnea data can be detected to be apnea correctly. However, the efficiency of detecting normal periods should be similar to the result of part 1. The specificity of HRV is 74.07%, which is a moderately acceptable result while the specificity of PRV is 67.33% that is lower as expected. When the sensitivity increases, the specificity is normally decreased. The model still should be improved by increasing the number of datasets or changing to other datasets that have less noise and more significant difference between apnea and normal parts of ECG or PPG signal.

According to the result in part 2 when comparing HRV and PRV, the efficiency value of both signals do not have any significant difference although accuracy and specificity of HRV is a little bit higher. At first, it is expected that specificity of HRV will be significantly higher than that of PRV as well as part 1 because more peak in ECG is detected as mentioned in table 36. The HRV graph is also more reliable due to lower noise and lower standard deviation of HRV compared to that of PRV as shown in table 35 and 36. However, it also depends on each dataset that has an unusual pattern of HRV and PRV.

Another important point to be discussed is about the datasets used in this study. The difference between apnea and normal part of signal in both ECG and PPG is unclear. From the literature review in chapter 2, there are small differences in the ECG signal between the normal and apnea part that make the HRV of apnea part lower than the normal part. However, no difference between apnea and normal signal can be observed in a labeled ECG or PPG signal as shown in figure 17 and 25 in chapter 3. The pattern in labeled HRV and PRV graphs also cannot

be observed by naked eyes. Therefore, other datasets from other open sources should be used to confirm whether the result of this study is affected due to the apnea label or not.

In table 36 to 40, every position in ECG and PPG signal is predicted whether it is normal or apnea period. The outcome is not satisfactory. The red line of prediction from HRV and the green line from PRV should match the ground truth. Only in some datasets such as dataset number 962 have high specificity that rarely have false positives. The highest accuracy of HRV is 79.17% and the specificity is 81.69%. And the highest accuracy of PRV from the test result is 84.55% and the specificity is 87.85%. However, the sensitivity is too low. If the sensitivity of the same model is increased, the specificity will be decreased. Therefore, the most possible solution is to add other effective features for apnea detection along with HRV or PRV.

There are many limitations in this study. First, this study is decided to use only MATLAB which is more recommended for image classification. To generate images takes too much time so if the input can be the array of numbers, it will be better for deep learning to be trained. But there are only a few types of input in MATLAB, and array of numbers cannot be used as input data. Second, generating images by using scale is not effective for HRV or PRV. The scale of color will change depending on the range of HRV or PRV in each segment. The range of HRV or PRV is all real numbers between 0 to 100, but the color data is only integers ranging from 1 to 256. If the scale is too wide, the low value will be converted into a similar blue color that is difficult to classify. For example, 10.1 and 10.2 will have the same color. Third, the other types of CNN and LSTM are not working for detected sequences of HRV or PRV signals in this study. It is

necessary to design a new CNN-LSTM model. Therefore, other types of network and program should be used instead of ResNet-50 and MATLAB.

In the future work, the most important thing is to increase the number of datasets. Due to the time limitation, it cannot be done in this study. Second, to make the model more practical, the length of each segment of HRV or PRV should be decreased so that each segment of the apnea class will contain all apnea labels. However, the number and complexity of features will also be lowered. This is the problem that should be considered. Most importantly, other deep networks should be used instead of ResNet-50 that can only take images input. New network should be designed to receive 1D signal input in MATLAB. Or other programs rather than MATLAB should be used to take the array input that is separately classified into two or more classes without having to convert them into images.

REFERENCES

1.What is sleep apnea? [Internet]. National Heart Lung and Blood Institute. U.S. Department of Health and Human Services; [cited 2023Apr29]. Available from: https://www.nhlbi.nih.gov/health/sleep-apnea

2.Javaheri S;Barbe F;Campos-Rodriguez F;Dempsey JA;Khayat R;Javaheri S;Malhotra A;Martinez-Garcia MA;Mehra R;Pack AI;Polotsky VY;Redline S;Somers VK; Sleep apnea: Types, mechanisms, and clinical cardiovascular consequences [Internet]. Journal of the American College of Cardiology. U.S. National Library of Medicine; [cited 2023Apr29]. Available from: https://pubmed.ncbi.nlm.nih.gov/28209226/

3.Finkel KJ, Searleman AC, Tymkew H, Tanaka CY, Saager L, Safer-Zadeh E, et al. Prevalence of undiagnosed obstructive sleep apnea among adult surgical patients in an academic medical center. Sleep Medicine [Internet]. 2009 Aug 1 [cited 2020 Nov 20];10(7):753–8. Available from: https://pubmed.ncbi.nlm.nih.gov/19186102/

4.Punjabi NM. The Epidemiology of Adult Obstructive Sleep Apnea. Proceedings of the American Thoracic Society [Internet]. 2008 Feb 15;5(2):136–43. Available from: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2645248/

5.Benjafield AV, Ayas NT, Eastwood PR, Heinzer R, Ip MSM, Morrell MJ, et al. Estimation of the global prevalence and burden of obstructive sleep apnoea: a literature-based analysis. The

Lancet Respiratory Medicine [Internet]. 2019 Aug;7(8):687–98. Available from: https://www.thelancet.com/journals/lanres/article/PIIS2213-2600(19)30198-5/fulltext

6.Mayo Clinic. Polysomnography (sleep study) - Mayo Clinic [Internet]. Mayoclinic.org. 2018. Available from: https://www.mayoclinic.org/tests-procedures/polysomnography/about/pac-20394877

7.Mohammadieh A, Sutherland K, Cistulli PA. Sleep disordered breathing: management update. Internal Medicine Journal. 2017 Nov;47(11):1241–7.

8.Mostafa, Mendonça, Ravelo-García, Morgado-Dias. A Systematic Review of Detecting Sleep Apnea Using Deep Learning. Sensors. 2019 Nov 12;19(22):4934.

9.Tyagi PK. Automatic detection of sleep apnea from single-lead ECG signal using enhanced-deep belief network model [Internet]. Biomedical Signal Processing and Control. Elsevier; 2022 [cited 2023Apr30]. Available from: https://www.sciencedirect.com/science/article/abs/pii/S1746809422008552

10.Penzel T. Is heart rate variability the simple solution to diagnose sleep apnoea? [Internet]. European Respiratory Society. European Respiratory Society; 2003 [cited 2023Apr30]. Available from: https://erj.ersjournals.com/content/22/6/870

11.Staff HHP. Heart rate variability: How it might indicate well-being [Internet]. Harvard Health. 2021 [cited 2023Apr30]. Available from: https://www.health.harvard.edu/blog/heart-rate-variability-new-way-track-well-2017112212789

12.Orbell S, Schneider H, Esbitt S, Gonzalez JS, Gonzalez JS, Shreck E, et al. Heart Rate Variability (HRV). In: Encyclopedia of Behavioral Medicine. New York, NY: Springer New York; 2013. p. 953–953.

13.Obstructive sleep apnea and heart disease - american thoracic society [Internet]. [cited 2023Apr30]. Available from: https://www.thoracic.org/patients/patient-resources/resources/obstructive-sleep-apnea-and-heart.pdf

14.Castaneda D, Esparza A, Ghamari M, Soltanpur C, Nazeran H. A review on wearable photoplethysmography sensors and their potential future applications in health care [Internet]. International journal of biosensors & bioelectronics. U.S. National Library of Medicine; 2018 [cited 2023Apr30]. Available from: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6426305/

15.Mejía-Mejía E, Budidha K, Abay TY, May JM, Kyriacou PA. Heart rate variability (HRV) and Pulse Rate Variability (PRV) for the assessment of autonomic responses [Internet]. Frontiers in physiology. U.S. National Library of Medicine; 2020 [cited 2023Apr30]. Available from: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7390908/

16.Lazazzera R, Deviaene M, Varon C, Buyse B, Testelmans D, Carrault G, et al. Detection and classification of sleep apnea and hypopnea using PPG and SPO 2 signals [Internet]. IEEE transactions on bio-medical engineering. U.S. National Library of Medicine; [cited 2023Apr30]. Available from: https://pubmed.ncbi.nlm.nih.gov/32997622/

17.Goyal M, Johnson J. Obstructive sleep apnea diagnosis and management. Mo Med [Internet]. 2017 [cited 2023 Jun 29];114(2):120–4. Available from: http://dx.doi.org/10.2310/im.1365

18.Bhutta BS, Alghoula F, Berim I. Hypoxia. StatPearls Publishing [Internet]. 2022 [cited 2023 Jun 29]. Available from: https://www.ncbi.nlm.nih.gov/books/NBK482316/

19.Causes of sleep apnea [Internet]. WebMD. [cited 2023 Jun 29]. Available from: https://www.webmd.com/sleep-disorders/sleep-apnea/obstructive-sleep-apnea-causes

20.Central sleep apnea [Internet]. Mayo Clinic. 2021 [cited 2023 Jun 29]. Available from: https://www.mayoclinic.org/diseases-conditions/central-sleep-apnea/symptoms-causes/syc-20352109

21.Khan MT, Franco RA. Complex sleep apnea syndrome. Sleep Disord [Internet]. 2014 [cited 2023 Jun 29];2014:798487. Available from: http://dx.doi.org/10.1155/2014/798487

22.Asghari A, Mohammadi F. Is Apnea-Hypopnea Index a proper measure for Obstructive Sleep Apnea severity? Med J Islam Repub Iran. 2013;27(3):161–2.

23.Senaratna CV, Perret JL, Lodge CJ, Lowe AJ, Campbell BE, Matheson MC, et al. Prevalence of obstructive sleep apnea in the general population: A systematic review. Sleep Med Rev [Internet]. 2017 [cited 2023 Jun 29];34:70–81. Available from: https://pubmed.ncbi.nlm.nih.gov/27568340/

24.Neruntarat C, Chantapant S. Prevalence of sleep apnea in HRH Princess Maha Chakri Srinthorn Medical Center, Thailand. Sleep Breath [Internet]. 2011 [cited 2023 Jun 29];15(4):641–8. Available from: https://pubmed.ncbi.nlm.nih.gov/20848319/

25.Berry RB, Abreu AR, Krishnan V, Quan SF, Strollo PJ, Malhotra RK. A transition to the American Academy of Sleep Medicine-recommended hypopnea definition in adults: initiatives of the Hypopnea Scoring Rule Task Force. J Clin Sleep Med [Internet]. 2022;18(5):1419–25. Available from: http://dx.doi.org/10.5664/jcsm.9952

26.Mbata G, Chukwuka J. Obstructive sleep apnea hypopnea syndrome. Ann Med Health Sci Res [Internet]. 2012 [cited 2023 Jun 29];2(1):74–7. Available from: http://dx.doi.org/10.4103/2141-9248.96943

27.Wells D. Hypopnea: Symptoms, treatments, and more [Internet]. Healthline. 2017 [cited 2023 Jun 29]. Available from: https://www.healthline.com/health/hypopnea

28.Rundo JV, Downey R 3rd. Polysomnography. Handb Clin Neurol [Internet]. 2019 [cited 2023 Jun 29];160:381–92. Available from: https://pubmed.ncbi.nlm.nih.gov/31277862/

29.Gerstenslager B, Slowik JM. Sleep Study. StatPearls Publishing; 2022.

30.Markun LC, Sampat A. Clinician-focused overview and developments in polysomnography. Curr Sleep Med Rep [Internet]. 2020 [cited 2023 Jun 29];6(4):309–21. Available from: http://dx.doi.org/10.1007/s40675-020-00197-5

31.Lucey BP, Mcleland JS, Toedebusch CD, Boyd J, Morris JC, Landsness EC, et al. Comparison of a single-channel EEG sleep study to polysomnography. J Sleep Res [Internet]. 2016 [cited 2023 Jun 29];25(6):625–35. Available from: http://dx.doi.org/10.1111/jsr.12417

32.Hoffman T. What is Heart Rate Variability (HRV) & why does it matter? [Internet]. Firstbeat. 2018 [cited 2023 Jun 29]. Available from: https://www.firstbeat.com/en/blog/what-is-heart-rate-variability-hrv/

33.What is Polysomnography? [Internet]. Healthline. 2020 [cited 2023 Apr 29]. Available from: https://www.healthline.com/health/polysomnography#preparation

34.Electrocardiogram [Internet]. Hopkinsmedicine.org. 2021 [cited 2023 Jun 29]. Available from: https://www.hopkinsmedicine.org/health/treatment-tests-and-therapies/electrocardiogram

35.Cadogan M. ECG lead positioning [Internet]. Life in the Fast Lane • LITFL. 2021 [cited 2023 Jun 29]. Available from: https://litfl.com/ecg-lead-positioning/

36.Characteristics of the normal electrocardiogram [Internet]. BrainKart. [cited 2023 Jun 29]. Available from: https://www.brainkart.com/article/Characteristics-of-the-Normal-Electrocardiogram_19237/

37.Researchgate.net. [cited 2023 Jun 29]. Available from: https://www.researchgate.net/figure/A-single-cycle-of-a-typical-ECG-with-the-important-points-labeled-ie-P-Q-R-S-and-T_fig1_289313797

38.Normal electrocardiography (ECG) intervals [Internet]. Medscape.com. 2021 [cited 2023 Jun 29]. Available from: https://emedicine.medscape.com/article/2172196-overview

39.Heart rate variability (HRV): What it is and how you can track it [Internet]. Cleveland Clinic. [cited 2023Apr30]. Available from: https://my.clevelandclinic.org/health/symptoms/21773-heart-rate-variability-hrv

40.Sports Medicine. Heart Rate [Internet]. Ucdavis.edu. [cited 2023 Jun 29]. Available from: https://health.ucdavis.edu/sports-medicine/resources/heart-rate

41.Geovanini GR, Vasques ER, de Oliveira Alvim R, Mill JG, Andreão RV, Vasques BK, et al. Age and sex differences in heart rate variability and vagal specific patterns - Baependi Heart Study. Glob Heart [Internet]. 2020;15(1):71. Available from: http://dx.doi.org/10.5334/gh.873

42.Held J, Vîslă A, Wolfer C, Messerli-Bürgy N, Flückiger C. Heart rate variability change during a stressful cognitive task in individuals with anxiety and control participants. BMC Psychol [Internet]. 2021;9(1):44. Available from: http://dx.doi.org/10.1186/s40359-021-00551-4

43.Shaffer F, Ginsberg JP. An overview of heart rate variability metrics and norms [Internet]. Frontiers in public health. U.S. National Library of Medicine; 2017 [cited 2023Apr30]. Available from: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5624990/

44.Fast Fourier transform - MATLAB fft [Internet]. Mathworks.com. [cited 2023 Jun 29]. Available from: https://www.mathworks.com/help/matlab/ref/fft.html

45.What is a spectrogram? [Internet]. Vibration Research. 2020 [cited 2023 Jun 29]. Available from: https://vibrationresearch.com/blog/what-is-a-spectrogram/

46.Castaneda D, Esparza A, Ghamari M, Soltanpur C, Nazeran H. A review on wearable photoplethysmography sensors and their potential future applications in health care. Int J Biosens Bioelectron [Internet]. 2018 [cited 2023 Jun 29];4(4):195–202. Available from: http://dx.doi.org/10.15406/ijbsbe.2018.04.00125

47.Researchgate.net. [cited 2023 Jun 29]. Available from: https://www.researchgate.net/figure/a-PPG-sensors-including-two-light-sources-and-two-light-sensors-b-Light-absorption_fig3_342787437

48.Pulse Oximetry - NanoViova [Internet]. Google.com. [cited 2023 Jun 29]. Available from: https://sites.google.com/a/nanoviova.com/www1/technology-expertise/pulse-oximetry

49.Pitson DJ, Sandell A, van den Hout R, Stradling JR. Use of pulse transit time as a measure of inspiratory effort in patients with obstructive sleep apnoea [Internet]. The European respiratory journal. U.S. National Library of Medicine; [cited 2023Apr30]. Available from: https://pubmed.ncbi.nlm.nih.gov/8586120/

50.What is MATLAB? [Internet]. Mathworks.com. [cited 2023 Jun 29]. Available from: https://www.mathworks.com/discovery/what-is-matlab.html

51.Bisong E. What is deep learning? In: Building Machine Learning and Deep Learning Models on Google Cloud Platform. Berkeley, CA: Apress; 2019. p. 327–9.

52.Unzueta D. Fully connected layer vs. Convolutional layer: Explained [Internet]. Built In. 2022 [cited 2023 Jun 29]. Available from: https://builtin.com/machine-learning/fully-connected-layer

53.Kulshrestha S. What is A convolutional neural network? In: Developing an Image Classifier Using TensorFlow. Berkeley, CA: Apress; 2019.

54.Kher R, Patel GH. Signal processing techniques for removing noise from ECG signals [Internet]. Jscholaronline.org. [cited 2023 Jun 29]. Available from: http://www.jscholaronline.org/articles/JBER/Signal-Processing.pdf

55.Sadhukhan D, Mitra M. R-peak detection algorithm for ECG using double difference and RR interval processing. Procedia Technol [Internet]. 2012;4:873–7. Available from: https://www.sciencedirect.com/science/article/pii/S2212017312004227

56.Simplilearn. What is a Confusion Matrix in Machine Learning? [Internet]. Simplilearn.com. Simplilearn; 2020 [cited 2023 Jun 29]. Available from: https://www.simplilearn.com/tutorials/machine-learning-tutorial/confusion-matrix-machine-learning

57.Khan MB. Cohen's Kappa Score [Internet]. Bootcamp. 2022 [cited 2023 Jun 29]. Available from: https://bootcamp.uxdesign.cc/cohens-kappa-score-33a0710b2fe0

58.Lin C-Y, Wang Y-W, Setiawan F, Trang NTH, Lin C-W. Sleep apnea classification algorithm development using a machine-learning framework and bag-of-features derived from electrocardiogram spectrograms. J Clin Med [Internet]. 2021 [cited 2023 Jun 29];11(1):192. Available from: https://pubmed.ncbi.nlm.nih.gov/35011934/

59.Dehkordi P. Pulse rate variability compared with heart rate variability in children. [Internet]. [cited 2023Apr30]. Available from: https://ieeexplore.ieee.org/document/6611059

60.Yuda E, Shibata M, Ogata Y, Ueda N, Yambe T, Yoshizawa M, et al. Pulse rate variability: a new biomarker, not a surrogate for heart rate variability. Journal of Physiological Anthropology [Internet]. 2020 Aug 18 [cited 2023Apr30];39(1). Available from: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7437069/

61.Zarei A, Beheshti H, Asl BM. Detection of sleep apnea using deep neural networks and single-lead ECG signals. Biomed Signal Process Control [Internet]. 2022;71(103125):103125. Available from: https://www.sciencedirect.com/science/article/pii/S1746809421007229

Appendix A

```
clear
close
load inclusion.mat
load exclusion.mat
inclusion(nonzeros(exclusion),:)=[];
fileno= inclusion(1:122,1);
starttime= inclusion(:,4);
endtime= inclusion(:,5);
fs=256;
amount= length(fileno);
set(groot, 'defaultFigureWindowState', 'maximized');
for i= 1:121
i
if fileno(i) <= 9
file= append('mesasleep\mesa-sleep-000',num2str(fileno(i)),'.edf');
elseif fileno(i) <=99
file= append('mesasleep\mesa-sleep-00',num2str(fileno(i)),'.edf');
else
file= append('mesasleep\mesa-sleep-0',num2str(fileno(i)),'.edf');
end
data= edfread(file);
```

```
nnn=size(data.EKG,1); %number of R that we want to see
inv=10; %number of interval to calculate rmssd
xmlfile= erase(file, '.edf');
xmlfile= append(xmlfile,'-nsrr.xml');
d=readstruct(xmlfile, 'FileType', 'xml');
d1= struct2cell(d);
d2 = d1{3, 1}.ScoredEvent;
Events = "";
Start = 0;
Duration = 0;
for s1= 1:length(d2)
%if  (d2(s1).EventConcept  ==  "Hypopnea|Hypopnea"  ||  d2(s1).EventConcept  ==  "Obstructive
apnea|Obstructive Apnea" || d2(s1).EventConcept == "Unsure|Unsure")
%if (d2(s1).EventConcept == "Hypopnea|Hypopnea")%"Obstructive apnea|Obstructive Apnea")
if (d2(s1).SignalLocation == "Flow")
Events(end+1)=d2(s1).EventConcept;
Start(end+1) = d2(s1).Start;
Duration(end+1) = d2(s1).Duration;
end
end
Events(1) = [];
Start(1) = [];
Duration(1) = [];
```

```
End =(Start) + (Duration);

EventsNo(i)=size(Events,2);

tab = table(Events',Start',Duration',End');

tab = renamevars(tab,["Var1","Var2","Var3","Var4"],["Events","Start","Duration","End"]);

inclusiontime= double(starttime(i)+1:(tab.End(end)-(inv/2)+1));

ECG= cell2mat(data.EKG);

ECG=-ECG;

ECGfil= lowpass(ECG,10,500);

% ECGfil= lowpass(ECG,15,256);

% ECGfil= highpass(ECGfil,5,256);

if length(inclusiontime(1)*fs : tab.End(end)*fs)> length(ECGfil)

ECGfil=ECGfil(inclusiontime(1)*fs : inclusion(end)*fs);

ECGfiltime= [inclusiontime(1): 1/fs : inclusion(end)]';

else

ECGfil=ECGfil(inclusiontime(1)*fs : tab.End(end)*fs);

ECGfiltime= [inclusiontime(1): 1/fs : tab.End(end)]';

end

ECGfilz= (ECGfil- mean(ECGfil))./std(ECGfil); %z-score

[ecg_h, ecg_b] = rmnoise(ECGfilz, fs);

ECGandtime= [ECGfiltime ECGfil];

ecg_signal= ecg_b;

YY= ecg_b;

XX= ECGfiltime;
```

```
Yt = YY;

Xt = XX; % 256 per sec

Yt = YY;

Xt = XX; % 256 per sec

% k = 10;

wsize = 800;

% Find second derivative

m = zeros(1, length(Yt)-1);

for inj = 1:length(Yt)-1

m(inj) = Yt(inj+1) - Yt(inj);

end

% Find all peaks

peakbool = zeros(1, length(m)-1);

for inj = 1:length(m)-1

if (m(inj) > 0) && (m(inj+1) < 0) % Pos to Neg

peakbool(inj+1) = Yt(inj); % Top peak

elseif (m(inj) < 0) && (m(inj+1) > 0) % Neg to Pos

peakbool(inj+1) = Yt(inj); % Bottom peak

else

peakbool(inj+1) = 0;

end

end

prev = 0;
```

```
finalpeak = zeros(1, length(peakbool));
for inj = 1:wsize:length(peakbool)
% each time i to i+wsize
counter = 0;
val = 0;
bee = inj+wsize;
if bee > length(peakbool)
bee = length(peakbool);
end
for j = inj:bee
if peakbool(j) ~= 0
counter = counter + 1;
val = val + abs(peakbool(j));
end
end
val = val/counter;
%disp(i)
for j = inj:bee
if peakbool(j) ~= 0
if (peakbool(j) > val) && (abs(prev) > val) && (prev < 0) % Current big pos, prev big neg
for jay = j+1:length(peakbool)
if (abs(peakbool(jay)) > val) && (peakbool(jay) < 0) % Next is big neg
finalpeak(j) = peakbool(j);
```

```
break
elseif peakbool(jay) ~= 0 % Not big neg peak
break % ignore it
end
end
end
prev = peakbool(j);
end
end
end
% Getting Final results
nX = [];
nY = [];
counter = 0;
for inj = 1:length(finalpeak)
if finalpeak(inj) ~= 0
counter = counter + 1;
nX(counter) = Xt(inj);
nY(counter) = Yt(inj);
ind(counter) = inj;
end
end
nXdiff = zeros(1, length(nX));
```

```
for inj = 1:length(nX)-1
nXdiff(inj) = nX(inj+1) - nX(inj);
end
% Filter out close proximity ones
n2X = nX;
n2Y = nY;
val = (mean(nXdiff)/3);
boolnxdiff = nXdiff < val;
j = 0;
for inj = 1:length(boolnxdiff)
if (boolnxdiff(inj) == 1) && (inj > j)
for j = inj+1:length(boolnxdiff)
if boolnxdiff(j) == 0
[maxval, inde] = max(n2Y(inj:j));
% disp([i j inde])
for jay = inj:j
if jay ~= (inj - 1 + inde)
n2X(jay) = 0;
n2Y(jay) = 0;
end
end
end
break
end
```

```
end
end
end
xout = nonzeros(n2X);
yout = nonzeros(n2Y);
xyout= [xout yout];
xyoutmat{i}= xyout;
hold off
plot(Xt, Yt)
% plot(Yt)
hold on
% scatter(ind, nY, 'filled', 'r')
scatter(nX, nY, 'filled', 'r')
scatter(xout, yout, 'filled', 'b')
miss=[];
diff= xout(2:length(xout))-xout(1:length(xout)-1);
count=0;
for loop=1:length(diff)-1
if diff(loop) > 1.4
count= count+1;
diffn(loop)=diff(loop);
miss=[miss; xout(loop)];
end
```

```
end
countmat(i)=count;
%%
t= ECGfiltime;
%ecg_signal= ecg_b;
figure(1);
plot(t, ecg_signal, 'b');
axis tight;
hold on;
scatter(xout, yout, 'filled','g');
%scatter(xoutmiss, youtmiss, 'filled','r');
%scatter(xoutmat, youtmat,'filled','r');
%scatter(xyout(:,1), xyout(:,2), 'filled','g');
xlabel('Time (s)');
ylabel('Amplitude');
legend('ECG Signal', 'R-Peaks');
title('ECG Signal with R-Peak Detection');
%
% Auto zoom and slide at unusual peak interval
% for mi=1: length(miss)
% mi
% range= miss(mi,1);
% figure(1), axis([range-50 range+50 -1 1]);
```

```
% pause(2)
% end
figure(1), axis([inclusiontime(1) tab.End(end) -1 1]);
saveas(figure(1),"figure4\"+inclusion(i,1)+".jpg");
figure(1), axis([inclusiontime(1) inclusiontime(1)+100 -1 1]);
saveas(figure(1),"figure4\"+inclusion(i,1)+"rpeak-1.jpg");
figure(1), axis([inclusiontime(1)+1000 inclusiontime(1)+1050 -1 1]);
saveas(figure(1),"figure4\"+inclusion(i,1)+"rpeak-2.jpg");
figure(1), axis([inclusiontime(1)+17600 inclusiontime(1)+17650 -1 1]);
saveas(figure(1),"figure4\"+inclusion(i,1)+"rpeak-3.jpg");
close(figure(1))
load('matme\matxyout115.mat')
%xyout= xyoutmat{i};
%rpeaktime= xyout(:,1);
rpeaktime= xout;
for inv= [300, 10]
HRV1e= rpeaktime*1000; %unit= ms
HRV2e=HRV1e(2)-HRV1e(1);
for n=2:length(HRV1e)-1
HRV2e(end+1)= HRV1e(n+1)-HRV1e(n);
end
HRV3e=HRV2e(2)-HRV2e(1);
for m=2:length(HRV2e)-1
```

```
HRV3e(end+1)= HRV2e(m+1)-HRV2e(m);
end
HRV3e=HRV3e.^2;
HRV4e=[];
for le= 1: length(HRV3e)-inv+1
HRV4e(le)= sum(HRV3e(le:le+inv-1));
end
rmssde= sqrt(HRV4e./(inv-1));
for range = 1: length(rmssde)
if rmssde(range) >=100
rmssde(range) =100;
end
end
Time = HRV1e((inv/2)+1 : length(HRV1e)-(inv/2)-1);
Time = Time/1000;
Time = Time'; %unit:sec
if inv==300
rmssde_300 = rmssde;
Time_300= Time;
end
end
zz= zeros(length(Time),1)';
for z1= 1:length(Start)
```

```
for z2 = 2: length(zz)-1
if (Time(z2-1) <= End(z1) && Time(z2+1) >= Start(z1))
zz(z2) = 1;
end
end
end
ze= zz.*rmssde;
zz_300= zeros(length(Time_300),1)';
for z1= 1:length(Start)
for z2 = 2: length(zz_300)-1
if (Time_300(z2-1) <= End(z1) && Time_300(z2+1) >= Start(z1))
zz_300(z2) = 1;
end
end
end
ze_300= zz_300.*rmssde_300;
me= [Time ;rmssde; zz];
me_300= [Time_300 ;rmssde_300; zz_300];
memat{i}=me;
me_300mat{i}=me_300;
rmssdecut= rmssde(146:length(rmssde)-145);
rmssd=[rmssdecut; rmssde_300; Time_300];
rmssdmat{i}=rmssd;
```

```
zzecg= zeros(length(ECGfiltime),1)';
%zecgtime= zeros(length(ECGfiltime),1)';
for z1= 1:length(Start)
for z2 = 2: length(zzecg)-1
if (ECGfiltime(z2-1) <= End(z1) && ECGfiltime(z2+1) >= Start(z1))
zecgtime(z2)=ECGfiltime(z2);
zzecg(z2) = 1;
end
end
end
zecg= zzecg'.*ecg_signal;
zecgtime= zzecg'.*ECGfiltime;
zecg=nonzeros(zecg);
zecgtime=nonzeros(zecgtime);
t= ECGfiltime;
%ecg_signal= ecg_b;
figure(2); subplot(311)
plot(t, ecg_signal, 'b');
%axis ([inclusiontime(1)+20000 inclusiontime(1)+20100 -1 1]);
axis tight;
hold on;
plot(zecgtime,zecg,'-red')
scatter(xyout(:,1), xyout(:,2), 'filled','g');
```

```
xlabel('Time (s)');
ylabel('Amplitude');
legend('ECG Signal', 'Apnea','R-Peaks');
title('ECG Signal with R-Peak Detection');
%figure('units','normalized','outerposition',[0 0 1 1])
figure(2), subplot(312), plot(Time, rmssde,'blue')
axis([inclusiontime(1) Time(end) 1 100]);
grid on, xlabel('time(s)'), ylabel('HRV(ms)'),title('HRV of ECG 10 seconds')
hold on;
plot(Time,ze,'.red')
figure(2), subplot(313), plot(Time_300, rmssde_300,'blue')
axis([inclusiontime(1) Time(end) 1 100]);
grid on, xlabel('time(s)'), ylabel('HRV(ms)'),title('HRV of ECG 300 seconds')
hold on;
%subplot(211)
plot(Time_300,ze_300,'.red')
hold off;
saveas(figure(2),"figure4\"+inclusion(i,1)+"rmssde.jpg");
close(figure(2))
end
```

Appendix B

Noise removal by using filters

Complete pan tompkins implementation ECG QRS Detector [Internet]. [cited 2023 Jul 1]. Available from: https://www.mathworks.com/matlabcentral/fileexchange/45840-complete-pan-tompkins-implementation-ecg-qrs-detector

```
function [ecg_h, ecg_b] = rmnoise(ecg_signal, fs)
ecg_signal = ecg_signal - mean(ecg_signal);
%Low Pass Filter
wd = 12*2/fs;
norder = 3;
[a,b] = butter(norder,wd,'low');
ecg_l = filtfilt(a,b,ecg_signal);
ecg_l = ecg_l/ max(abs(ecg_l));
%High Pass filter
wd = 5*2/fs;
norder = 3;
[a,b] = butter(norder,wd,'high');
ecg_h = filtfilt(a,b,ecg_l);
ecg_h = ecg_h/ max(abs(ecg_h));
```

```
% bandpass filter
f1=5;
f2=15;
wd=[f1 f2]*2/fs;
norder = 3;
[a,b] = butter(norder,wd);
ecg_b = filtfilt(a,b,ecg_h);
ecg_b = ecg_b/ max( abs(ecg_b));
```

Appendix C

Array of HRV Segmentation and Classification

```
for mm = 1:115 % 1: length(memat)
me= memat{mm};
rmssde= me(2,:);
ze= me(3,:);
r0=0;
for r00=1:length(rmssde)
if rmssde(r00)==100
r0(r00)= r00;
end
end
r0=nonzeros(r0);
rmssde(:,r0)=[];
ze(:,r0)=[];
%r1,r2 is in the same position as Time= (inv/2)+1: size(rmssde,2)+(inv/2);
r_one= zeros(length(rmssde),60);
r_zero= zeros(length(rmssde),60);
z=ze;
for r2= 1: 60: length(rmssde)-59
if sum(z(r2:r2+59))==0
```

```
r_zero(r2,:)=rmssde(r2:r2+59);
end
end
% for r2= 1: 10:length(rmssde)-59
% if sum(z(r2:r2+59))>=50
% z(r2)=1;
% r_one(r2,:)=rmssde(r2:r2+59);
% end
% end
r_ap= zeros(length(z),1);
for r2= 1:length(rmssde)-59
if z(r2)==1
r_ap(r2)=rmssde(r2);
end
end
r_ap= nonzeros(r_ap);
for r2= 1: 40:length(r_ap)-59
r_one(r2,:)=r_ap(r2:r2+59);
end
r4=0;
r5=0;
for r3=1:length(rmssde)
if r_zero(r3,:)==zeros(1,60)
```

```
r4(end+1)= r3;
end
if r_one(r3,:) == zeros(1,60)
r5(end+1)= r3;
end
end
r4(1)=[];
r5(1)=[];
r_zero(r4,:)=[];
r_one(r5,:)=[];
tsla=[];
nor=[];
for r6=1:size(r_one,1)
tsla{r6}=r_one(r6,:);
end
for r7=1:size(r_zero,1)
nor{r7}=r_zero(r7,:);
end
TApnea{mm}= tsla;
TNormal{mm}= nor;
end
```

Appendix D

Convert Apnea and Normal Segments to Images

```
%tA = [TApnea{:,1:112}];
load('C:\Users\ASUS\Desktop\capstone\Train3\tA.mat')
tA= cell2mat(tA);
%SLA= SLA(1,(1:874000));
tsla= reshape(tA,[60 13373]);
%rands= randperm(length(tsla));
%rands= sort(rands(1:10000)); %length(SS)));
load('C:\Users\ASUS\Desktop\capstone\Train5\rand.mat')
%for i=1:length(tsla)
for i= rand(2528:length(rand))
z=tsla(:,i);
figure("Visible","off","Units","pixels","Position",[100 100 143.6 143.6]);
axes("Position",[0 0 1 1]);
image = image(z,'CDataMapping','scaled');
saveas(image,"Train5/newap/"+ i +".jpg");
clear image;
end

%%
load('C:\Users\ASUS\Desktop\capstone\Train3\Normal.mat')
```

```
%load('D:\work\capstone\Train1\\Normal.mat', 'TNormal')

tN = [TNormal{:,1:112}];

tN= cell2mat(tN);

%NOR= NOR(1,(1:5612300));

%NOR= NOR(1,(1:length(SLA)));

tnor= reshape(tN,[60 35506]);

%rand= randperm(length(tN));

%rand= sort(rand(1:10000)); %length(SS)));

load('C:\Users\ASUS\Desktop\capstone\Train5\randnor.mat')

for j= rands(2733:length(rands))

%for j= 1:length(tnor)

z=tnor(:,j);

figure("Visible","off","Units","pixels","Position",[100 100 143.6 143.6]); %143.5 143.5]);

axes("Position",[0 0 1 1]);

image = image(z,'CDataMapping','scaled');

saveas(image,"Train5/newnor/"+j+".jpg");

clear image;

end
```

Appendix E

Output test for images and the whole signal

```
%test output
clear
%load trainedNetwork_1
%load trainInfoStruct_1
load('C:\Users\ASUS\Desktop\capstone\Train6\trainedNetwork_2.mat')
load('C:\Users\ASUS\Desktop\capstone\Train6\trainInfoStruct_2.mat')
setnum=116;
%%
for i=1:32
f= imread("Train4/test"+setnum+"/Apnea"+ i +".jpg"); %("test\"+i+"21sla.jpg");
%Classify the test image using the trained network.
[YPred,probs] = classify(trainedNetwork_2,f);
%figure(i),imshow(f);
label = YPred;
title(string(label) + ", " + num2str(100*max(probs),3) + "%");
slalabel(i)= [YPred];
slatrue(i)="Apnea";
ProbSLA(i)=100*max(probs);
end
```

```
%%
for j=1:688 %rand(1:length(rand)) %1:86
f= imread("Train4/test"+setnum+"/Normal"+j+".jpg");%("Test\"+i+"21nor.jpg");
%f= imresize(g, [875,656]);
%f= imresize(g, [656,875]);
%Classify the test image using the trained network.
[YPred,probs] = classify(trainedNetwork_2,f);
%figure(i),imshow(f);
label = YPred;
title(string(label) + ", " + num2str(100*max(probs),3) + "%");
norlabel(j)= YPred;
nortrue(j)="Normal";
ProbNOR(j)=100*max(probs);
end
predictedlabel= [slalabel, norlabel];
predictedlabel= string(predictedlabel);
truelabel= [slatrue, nortrue];
figure("Visible","on")
cm= confusionchart(truelabel, predictedlabel)
%saveas(cm,"Train8tryhalf\cm"+setnum+".jpeg");
%%
aplabel = renamecats(slalabel,"Apnea","1");
nlabel = renamecats(aplabel,"Normal","0");
```

```
numlabel= str2num(char(nlabel))';
load('C:\Users\ASUS\Desktop\capstone\Train4\TimeApnea.mat')
timeA = [TimeApnea{:,setnum}];
timeA= cell2mat(timeA);
timeap= reshape(timeA,[60 32]);
%create time for apnea (True Positive)
aplabel=zeros(size(timeap));
for num1=1:length(numlabel)
aplabel(num1,:)=numlabel(1,:);
end
plotap=numlabel.*timeap;
%%
aplabel = renamecats(norlabel,"Apnea","1");
nlabel = renamecats(aplabel,"Normal","0");
numlabel= str2num(char(nlabel))';
load('C:\Users\ASUS\Desktop\capstone\Train4\TimeNormal.mat')
timeN = [TimeNormal{:,setnum}];
timeN= cell2mat(timeN);
timenor= reshape(timeN,[60 688]);
%create time plot for false positive (normal but detect apnea)
nlabel=zeros(size(timenor));
for num1=1:length(numlabel)
nlabel(:,num1)=numlabel(:,num1);
```

```
end
plotnorap=numlabel.*timenor;
%%
%plot true negative
aplabel = renamecats(norlabel,"Apnea","0");
nlabel = renamecats(aplabel,"Normal","1");
numlabel= str2num(char(nlabel))';
load('C:\Users\ASUS\Desktop\capstone\Train4\TimeNormal.mat')
timeN = [TimeNormal{:,setnum}];
timeN= cell2mat(timeN);
timenor= reshape(timeN,[60 688]);
%create time plot for false positive (normal but detect apnea)
nlabel=zeros(size(timenor));
for num1=1:length(numlabel)
nlabel(:,num1)=numlabel(:,num1);
end
plotnor=nlabel.*timenor;
%%
load('C:\Users\ASUS\Desktop\capstone\Train3\metest.mat')
mm = setnum;
me= memat{mm};
ll=[];
for mi = 1: length(me)
```

```
if me(2,mi)==100 %(max(me(2,:)) + min(me(2,:)))/2 %mean(me(2,:))
ll(mi)=mi;
end
end
ll=nonzeros(ll);
me(:,ll)=[];
meremove{mm}=me;
set(groot, 'defaultFigureWindowState', 'maximized');
figure(4),subplot(211),plot(me(1,:), me(2,:), 'b')
axis([me(1,1) me(1,end) 1 100]);
hold on;
scatter(me(1,:), me(3,:).*me(2,:), 'filled','black')
hold off;
xlabel('Time (s)');, ylabel('HRV (ms)');
legend('HRV', 'Apnea');
title('HRV and True Label');
figure(4),subplot(212),plot(me(1,:), me(2,:), 'b')
axis([me(1,1) me(1,end) 1 100]);
%axis([21000 22000 1 100]);
hold on;
plot(plotnorap, ones(size(plotnorap)).*36,'*r')
scatter(plotap, ones(size(plotap)).*40, 'filled','black')
plot(plotnor, ones(size(plotnor)).*40,'*g')
```

```
xlabel('Time (s)');, ylabel('HRV (ms)');
legend('HRV', 'False Positive','True Positive','True Negative');
title('HRV and Apnea Prediction');
hold off;
%saveas(figure(4),"Train6\"+setnum+".jpeg");
%close(figure(4));
%Plot Binary
labelall= [slalabel, norlabel];
labelall = renamecats(labelall,"Apnea","1");
labelall = renamecats(labelall,"Normal","0");
labelall= str2num(char(labelall))';
%Groundtruth
load('C:\Users\ASUS\Desktop\capstone\Train3\metest.mat')
mm = setnum;
me= memat{mm};
rmssde= me(2,:);
ze= me(3,:);
time=me(1,:);
%Prediction
timeall60=[timeap timenor];
labelall60=[];
for lb= 1:60
labelall60(lb,:)= labelall(1,:);
```

```
end
labelall1= labelall60(:);
timeall1= timeall60(:);
x= [timeall1 labelall1];
[~,inx]=sort(x(:,1));
xsort= x(inx, :);
ll=[];
avelabel=[];
for num2= 1: length(xsort)-1
if xsort(num2,1)== xsort(num2+1,1)
ll(num2+1)=num2+1;
avelabel(num2)= 0; %(xsort(num2,2)+xsort(num2+1,2))/2;
else
avelabel(num2)= xsort(num2,2);
avelabel(num2+1)= xsort(num2,2);
end
end
xsort(:,2)= avelabel(:,:)';
ll=nonzeros(ll);
xsort(ll,:)=[];
figure(5),plot(time, ze)
hold on;
plot(xsort(:,1),xsort(:,2),'r')
```

```
hold off;
axis([xsort(1,1) xsort(end,1) -2 2]);
legend('Ground Truth','Prediction')
xlabel('Time (s)'), ylabel('Apnea');
title('Apnea Prediction');
saveas(figure(5),"Train6/"+setnum+"binary.jpeg");
figure(6),subplot(311),plot(time, ze)
hold on;
plot(xsort(:,1),xsort(:,2),'r')
hold off;
axis([xsort(1,1) xsort(1,1)+10000 -2 2]);
legend('Ground Truth','Prediction from HRV')
xlabel('Time (s)'), ylabel('Apnea');
title('Apnea Prediction');
subplot(312),plot(time, ze)
hold on;
plot(xsort(:,1),xsort(:,2),'r')
hold off;
axis([xsort(1,1)+10001 xsort(1,1)+20000 -2 2]);
legend('Ground Truth','Prediction from HRV')
xlabel('Time (s)'), ylabel('Apnea');
subplot(313),plot(time, ze)
hold on;
```

```
plot(xsort(:,1),xsort(:,2),'r')
hold off;
axis([xsort(1,1)+20001 xsort(end,1) -2 2]);
legend('Ground Truth','Prediction from HRV')
xlabel('Time (s)'), ylabel('Apnea');
saveas(figure(6),"Train6/"+setnum+"binarysep.jpeg");
apnea=0;
for count=1:length(ze)
if ze(count)==1
apnea=apnea+1;
end
end
apneatest=0;
for count=1:length(xsort)
if xsort(count,2)~=0
apneatest=apneatest+1;
end
end
```

Appendix F

Data preparation for PPG

```
% clear
% close
load inclusion.mat
load exclusion.mat
inclusion(nonzeros(exclusion),:)=[];
fileno= inclusion(1:116,1);
starttime= inclusion(:,4);
endtime= inclusion(:,5);
fs=256;
amount= length(fileno);
set(groot, 'defaultFigureWindowState', 'maximized');
for i= 1%:115
i
if fileno(i) <= 9
file= append('mesasleep\mesa-sleep-000',num2str(fileno(i)),'.edf');
elseif fileno(i) <=99
file= append('mesasleep\mesa-sleep-00',num2str(fileno(i)),'.edf');
else
file= append('mesasleep\mesa-sleep-0',num2str(fileno(i)),'.edf');
end
```

```
data= edfread(file);


inv=10; %number of interval to calculate rmssd
xmlfile= erase(file, '.edf');
xmlfile= append(xmlfile,'-nsrr.xml');
d=readstruct(xmlfile, 'FileType', 'xml');
d1= struct2cell(d);
d2 = d1{3, 1}.ScoredEvent;
Events = "";
Start = 0;
Duration = 0;
for s1= 1:length(d2)
if (d2(s1).SignalLocation == "Flow")
Events(end+1)=d2(s1).EventConcept;
Start(end+1) = d2(s1).Start;
Duration(end+1) = d2(s1).Duration;
end
end
Events(1) = [];
Start(1) = [];
Duration(1) = [];
End =(Start) + (Duration);
EventsNo(i)=size(Events,2);
```

```
tab = table(Events',Start',Duration',End');

tab = renamevars(tab,["Var1","Var2","Var3","Var4"],["Events","Start","Duration","End"]);

inclusiontime= double(starttime(i)+1:(tab.End(end)-(inv/2)+1));

PPG= cell2mat(data.Pleth);

PPGfil= lowpass(PPG,5,256);

inclusiontime= double(starttime(i)+1:(tab.End(end)-(inv/2)+1));

[ppg_h, ppg_b] = rmnoise(PPGfil, fs);

if length(inclusiontime(1)*fs : tab.End(end)*fs)> length(PPGfil)

PPGfil=PPGfil(inclusiontime(1)*fs : inclusion(end)*fs);

PPGfiltime= [inclusiontime(1): 1/fs : inclusion(end)]';

else

PPGfil=PPGfil(inclusiontime(1)*fs : tab.End(end)*fs);

PPGfiltime= [inclusiontime(1): 1/fs : tab.End(end)]';

end

PPGfilz= (PPGfil- mean(PPGfil))/std(PPGfil); %z-score

PPGfilz = imgaussfilt(PPGfilz);

PPGfilz = medfilt1(PPGfilz);

subplot(211),plot(PPGfiltime, PPG)

axis([20000 20050 -0.2 0.2])

xlabel('Time (s)');

ylabel('Amplitude');

title('PPG Signal Before Filter');

subplot(212),plot(PPGfiltime, PPGfilz)
```

```
axis([20000 20050 -2 2])
xlabel('Time (s)');
ylabel('Amplitude');
title('PPG Signal After Filter');
[~, cycle_peak_indices] = findpeaks(PPGfilz, 'MinPeakHeight', 0);
xout= PPGfiltime(cycle_peak_indices);
yout= PPGfilz(cycle_peak_indices);
ll=[];
lrdiff= xout(2:length(xout))-xout(1:length(xout)-1);
for l= 1:length(xout) -1
if xout(l+1)- xout(l) <= 0.5
if yout(l+1) > yout(l)
ll(l)= l;
else
ll(l)= l+1;
end
end
end
ll= nonzeros(ll);
xout(ll)=[];
yout(ll)=[];
ll=[];
lrdiff= xout(2:length(xout))-xout(1:length(xout)-1);
```

```
for l= 1:length(xout) -1
if xout(l+1)- xout(l) <= 0.5
if yout(l+1) > yout(l)
ll(l)= l;
else
ll(l)= l+1;
end
end
end
ll= nonzeros(ll);
xout(ll)=[];
yout(ll)=[];
ll=[];
lrdiff= xout(2:length(xout))-xout(1:length(xout)-1);
for l= 1:length(xout) -1
if xout(l+1)- xout(l) <= 0.5
if yout(l+1) > yout(l)
ll(l)= l;
else
ll(l)= l+1;
end
end
end
```

```
ll= nonzeros(ll);
xout(ll)=[];
yout(ll)=[];
wXsize = 6;
stdrdev = zeros(1,length(yout));
counter = 0;
for window = 1:length(yout)-wXsize
counter = counter +1;
stdrdev(counter) = std(yout(window:window+wXsize));
if std(yout(window:window+wXsize)) > 1.5
xout(window+wXsize/2) = 0;
yout(window+wXsize/2) = 0;
end
end
xout = nonzeros(xout);
yout = nonzeros(yout);
xyout= [xout yout];
figure(1),plot(PPGfiltime, PPGfilz, 'b');
hold on;
scatter(xout, yout, 'filled','g');
axis tight;
xlabel('Time (s)');
ylabel('Amplitude');
```

```
legend('ppg Signal', 'Peaks');
title('ppg Signal with Peak Detection');
% saveas(figure(5),"figure1\"+inclusion(i,1)+"-filter.jpg");
% close(figure(5));
miss=[];
diff= xout(2:length(xout))-xout(1:length(xout)-1);
count=0;
for loop=1:length(diff)-1
if diff(loop) > 1.4
count= count+1;
diffn(loop)=diff(loop);
miss=[miss; xout(loop)];
end
end
countmat(i)=count;
diffmat{i}= diffn;
%%
% run loop to check peak plot when peak-peak is too high.
% for mi=1: length(miss)
% mi
% range= miss(mi,1);
% figure(1), axis([range-50 range+50 -1 1]);
% pause(2)
```

```
% end
%%
figure(1), axis([inclusiontime(1) tab.End(end) -4 4]);
saveas(figure(1),"figure4p\"+inclusion(i,1)+".jpg");
figure(1), axis([inclusiontime(1) inclusiontime(1)+100 -4 4]);
saveas(figure(1),"figure4p\"+inclusion(i,1)+"peak-1.jpg");
figure(1), axis([inclusiontime(1)+1000 inclusiontime(1)+1050 -4 4]);
saveas(figure(1),"figure4p\"+inclusion(i,1)+"peak-2.jpg");
figure(1), axis([inclusiontime(1)+17600 inclusiontime(1)+17650 -4 4]);
saveas(figure(1),"figure4p\"+inclusion(i,1)+"peak-3.jpg");
close(figure(1))
%load('matme\matxyout115.mat')
%xyout= xyoutmat{i};
rpeaktime= xyout(:,1);
%rpeaktime= xout;
for inv= [300, 10]
PRV1p= rpeaktime*1000; %unit= ms
PRV2p=PRV1p(2)-PRV1p(1);
for n=2:length(PRV1p)-1
PRV2p(end+1)= PRV1p(n+1)-PRV1p(n);
end
PRV3p=PRV2p(2)-PRV2p(1);
for m=2:length(PRV2p)-1
```

```
PRV3p(end+1)= PRV2p(m+1)-PRV2p(m);
end
PRV3p=PRV3p.^2;
PRV4p=[];
for le= 1: length(PRV3p)-inv+1
PRV4p(le)= sum(PRV3p(le:le+inv-1));
end
Time = PRV1p((inv/2)+1 : length(PRV1p)-(inv/2)-1);
Time = Time/1000;
Time = Time'; %unit:sec
rmssdp= sqrt(PRV4p./(inv-1));
for range = 1: length(rmssdp)
if rmssdp(range) >=100
rmssdp(range) =0;
Time(range)=0;
end
end
rmssdp =nonzeros(rmssdp);
Time= nonzeros(Time);
if inv==300
rmssdp_300 = rmssdp;
Time_300= Time;
end
```

```
end
zz= zeros(length(Time),1);
for z1= 1:length(Start)
for z2 = 2: length(zz)-1
if (Time(z2-1) <= End(z1) && Time(z2+1) >= Start(z1))
zz(z2) = 1;
end
end
end
ze= zz.*rmssdp;
zz_300= zeros(length(Time_300),1);
for z1= 1:length(Start)
for z2 = 2: length(zz_300)-1
if (Time_300(z2-1) <= End(z1) && Time_300(z2+1) >= Start(z1))
zz_300(z2) = 1;
end
end
end
ze_300= zz_300.*rmssdp_300;
me= [Time ;rmssdp; zz];
me_300= [Time_300 ;rmssdp_300; zz_300];
memat{i}=me;
me_300mat{i}=me_300;
```

```
rmssdpcut= rmssdp(146:length(rmssdp)-145);

rmssd=[rmssdpcut; rmssdp_300; Time_300];

rmssdmat{i}=rmssd;

zzppg= zeros(length(PPGfiltime),1)';

%zppgtime= zeros(length(PPGfiltime),1)';

for z1= 1:length(Start)

for z2 = 2: length(zzppg)-1

if (PPGfiltime(z2-1) <= End(z1) && PPGfiltime(z2+1) >= Start(z1))

zppgtime(z2)=PPGfiltime(z2);

zzppg(z2) = 1;

end

end

end

zppg= zzppg'.*PPGfilz;

zppgtime= zzppg'.*PPGfiltime;

zppg=nonzeros(zppg);

zppgtime=nonzeros(zppgtime);

t= PPGfiltime;

%PPGfilz= ppg_b;

figure(2); subplot(311)

plot(t, PPGfilz, 'b');

%axis ([inclusiontime(1)+20000 inclusiontime(1)+20100 -1 1]);%tight;

axis tight;
```

```
hold on;
scatter(xout, yout, 'filled','g');
plot(zppgtime,zppg,'-red')
xlabel('Time (s)');
ylabel('Amplitude');
legend('ppg Signal', 'Apnea','Peaks');
title('ppg Signal with Peak Detection');
%figure('units','normalized','outerposition',[0 0 1 1])
figure(2), subplot(312), plot(Time, rmssdp,'blue')
axis([inclusiontime(1) Time(end) 1 100]);
grid on, xlabel('time(s)'), ylabel('PRV(ms)'),title('PRV of ppg 10 seconds')
hold on;
plot(Time,ze,'.red')
figure(2), subplot(313), plot(Time_300, rmssdp_300,'blue')
axis([inclusiontime(1) Time(end) 1 100]);
grid on, xlabel('time(s)'), ylabel('PRV(ms)'),title('PRV of ppg 300 seconds')
hold on;
%subplot(211)
plot(Time_300,ze_300,'.red')
hold off;
saveas(figure(2),"figure4p\"+inclusion(i,1)+"rmssdp.jpg");
close(figure(2))
end
```

Appendix G

ResNet 50 Layers

```
lgraph = layerGraph();
tempLayers = [
imageInputLayer([224 224 3],"Name","imageinput")
convolution2dLayer([7 7],64,"Name","conv1","Padding",[3 3 3 3],"Stride",[2 2])
batchNormalizationLayer("Name","bn_conv1","Epsilon",0.001)
reluLayer("Name","activation_1_relu")
maxPooling2dLayer([3 3],"Name","max_pooling2d_1","Padding",[1 1 1 1],"Stride",[2 2])];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
convolution2dLayer([1 1],64,"Name","res2a_branch2a","BiasLearnRateFactor",0)
batchNormalizationLayer("Name","bn2a_branch2a","Epsilon",0.001)
reluLayer("Name","activation_2_relu")
convolution2dLayer([3
3],64,"Name","res2a_branch2b","BiasLearnRateFactor",0,"Padding","same")
batchNormalizationLayer("Name","bn2a_branch2b","Epsilon",0.001)
reluLayer("Name","activation_3_relu")
convolution2dLayer([1 1],256,"Name","res2a_branch2c","BiasLearnRateFactor",0)
batchNormalizationLayer("Name","bn2a_branch2c","Epsilon",0.001)];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
```

```
convolution2dLayer([1 1],256,"Name","res2a_branch1","BiasLearnRateFactor",0)
batchNormalizationLayer("Name","bn2a_branch1","Epsilon",0.001)];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
additionLayer(2,"Name","add_1")
reluLayer("Name","activation_4_relu")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
convolution2dLayer([1 1],64,"Name","res2b_branch2a","BiasLearnRateFactor",0)
batchNormalizationLayer("Name","bn2b_branch2a","Epsilon",0.001)
reluLayer("Name","activation_5_relu")
convolution2dLayer([3
3],64,"Name","res2b_branch2b","BiasLearnRateFactor",0,"Padding","same")
batchNormalizationLayer("Name","bn2b_branch2b","Epsilon",0.001)
reluLayer("Name","activation_6_relu")
convolution2dLayer([1 1],256,"Name","res2b_branch2c","BiasLearnRateFactor",0)
batchNormalizationLayer("Name","bn2b_branch2c","Epsilon",0.001)];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
additionLayer(2,"Name","add_2")
reluLayer("Name","activation_7_relu")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
```

```
convolution2dLayer([1 1],64,"Name","res2c_branch2a","BiasLearnRateFactor",0)
batchNormalizationLayer("Name","bn2c_branch2a","Epsilon",0.001)
reluLayer("Name","activation_8_relu")
convolution2dLayer([3
3],64,"Name","res2c_branch2b","BiasLearnRateFactor",0,"Padding","same")
batchNormalizationLayer("Name","bn2c_branch2b","Epsilon",0.001)
reluLayer("Name","activation_9_relu")
convolution2dLayer([1 1],256,"Name","res2c_branch2c","BiasLearnRateFactor",0)
batchNormalizationLayer("Name","bn2c_branch2c","Epsilon",0.001)];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
additionLayer(2,"Name","add_3")
reluLayer("Name","activation_10_relu")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
convolution2dLayer([1    1],128,"Name","res3a_branch2a","BiasLearnRateFactor",0,"Stride",[2
2])
batchNormalizationLayer("Name","bn3a_branch2a","Epsilon",0.001)
reluLayer("Name","activation_11_relu")
convolution2dLayer([3
3],128,"Name","res3a_branch2b","BiasLearnRateFactor",0,"Padding","same")
batchNormalizationLayer("Name","bn3a_branch2b","Epsilon",0.001)
reluLayer("Name","activation_12_relu")
```

```
convolution2dLayer([1 1],512,"Name","res3a_branch2c","BiasLearnRateFactor",0)
batchNormalizationLayer("Name","bn3a_branch2c","Epsilon",0.001)];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
convolution2dLayer([1 1],512,"Name","res3a_branch1","BiasLearnRateFactor",0,"Stride",[2 2])
batchNormalizationLayer("Name","bn3a_branch1","Epsilon",0.001)];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
additionLayer(2,"Name","add_4")
reluLayer("Name","activation_13_relu")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
convolution2dLayer([1 1],128,"Name","res3b_branch2a","BiasLearnRateFactor",0)
batchNormalizationLayer("Name","bn3b_branch2a","Epsilon",0.001)
reluLayer("Name","activation_14_relu")
convolution2dLayer([3
3],128,"Name","res3b_branch2b","BiasLearnRateFactor",0,"Padding","same")
batchNormalizationLayer("Name","bn3b_branch2b","Epsilon",0.001)
reluLayer("Name","activation_15_relu")
convolution2dLayer([1 1],512,"Name","res3b_branch2c","BiasLearnRateFactor",0)
batchNormalizationLayer("Name","bn3b_branch2c","Epsilon",0.001)];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
```

```
additionLayer(2,"Name","add_5")
reluLayer("Name","activation_16_relu")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
convolution2dLayer([1 1],128,"Name","res3c_branch2a","BiasLearnRateFactor",0)
batchNormalizationLayer("Name","bn3c_branch2a","Epsilon",0.001)
reluLayer("Name","activation_17_relu")
convolution2dLayer([3
3],128,"Name","res3c_branch2b","BiasLearnRateFactor",0,"Padding","same")
batchNormalizationLayer("Name","bn3c_branch2b","Epsilon",0.001)
reluLayer("Name","activation_18_relu")
convolution2dLayer([1 1],512,"Name","res3c_branch2c","BiasLearnRateFactor",0)
batchNormalizationLayer("Name","bn3c_branch2c","Epsilon",0.001)];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
additionLayer(2,"Name","add_6")
reluLayer("Name","activation_19_relu")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
convolution2dLayer([1 1],128,"Name","res3d_branch2a","BiasLearnRateFactor",0)
batchNormalizationLayer("Name","bn3d_branch2a","Epsilon",0.001)
reluLayer("Name","activation_20_relu")
```

```
convolution2dLayer([3
3],128,"Name","res3d_branch2b","BiasLearnRateFactor",0,"Padding","same")
batchNormalizationLayer("Name","bn3d_branch2b","Epsilon",0.001)
reluLayer("Name","activation_21_relu")
convolution2dLayer([1 1],512,"Name","res3d_branch2c","BiasLearnRateFactor",0)
batchNormalizationLayer("Name","bn3d_branch2c","Epsilon",0.001)];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
additionLayer(2,"Name","add_7")
reluLayer("Name","activation_22_relu")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
convolution2dLayer([1     1],1024,"Name","res4a_branch1","BiasLearnRateFactor",0,"Stride",[2
2])
batchNormalizationLayer("Name","bn4a_branch1","Epsilon",0.001)];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
convolution2dLayer([1     1],256,"Name","res4a_branch2a","BiasLearnRateFactor",0,"Stride",[2
2])
batchNormalizationLayer("Name","bn4a_branch2a","Epsilon",0.001)
reluLayer("Name","activation_23_relu")
convolution2dLayer([3
3],256,"Name","res4a_branch2b","BiasLearnRateFactor",0,"Padding","same")
```

```
batchNormalizationLayer("Name","bn4a_branch2b","Epsilon",0.001)
reluLayer("Name","activation_24_relu")
convolution2dLayer([1 1],1024,"Name","res4a_branch2c","BiasLearnRateFactor",0)
batchNormalizationLayer("Name","bn4a_branch2c","Epsilon",0.001)];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
additionLayer(2,"Name","add_8")
reluLayer("Name","activation_25_relu")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
convolution2dLayer([1 1],256,"Name","res4b_branch2a","BiasLearnRateFactor",0)
batchNormalizationLayer("Name","bn4b_branch2a","Epsilon",0.001)
reluLayer("Name","activation_26_relu")
convolution2dLayer([3
3],256,"Name","res4b_branch2b","BiasLearnRateFactor",0,"Padding","same")
batchNormalizationLayer("Name","bn4b_branch2b","Epsilon",0.001)
reluLayer("Name","activation_27_relu")
convolution2dLayer([1 1],1024,"Name","res4b_branch2c","BiasLearnRateFactor",0)
batchNormalizationLayer("Name","bn4b_branch2c","Epsilon",0.001)];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
additionLayer(2,"Name","add_9")
reluLayer("Name","activation_28_relu")];
```

```
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
convolution2dLayer([1 1],256,"Name","res4c_branch2a","BiasLearnRateFactor",0)
batchNormalizationLayer("Name","bn4c_branch2a","Epsilon",0.001)
reluLayer("Name","activation_29_relu")
convolution2dLayer([3
3],256,"Name","res4c_branch2b","BiasLearnRateFactor",0,"Padding","same")
batchNormalizationLayer("Name","bn4c_branch2b","Epsilon",0.001)
reluLayer("Name","activation_30_relu")
convolution2dLayer([1 1],1024,"Name","res4c_branch2c","BiasLearnRateFactor",0)
batchNormalizationLayer("Name","bn4c_branch2c","Epsilon",0.001)];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
additionLayer(2,"Name","add_10")
reluLayer("Name","activation_31_relu")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
convolution2dLayer([1 1],256,"Name","res4d_branch2a","BiasLearnRateFactor",0)
batchNormalizationLayer("Name","bn4d_branch2a","Epsilon",0.001)
reluLayer("Name","activation_32_relu")
convolution2dLayer([3
3],256,"Name","res4d_branch2b","BiasLearnRateFactor",0,"Padding","same")
batchNormalizationLayer("Name","bn4d_branch2b","Epsilon",0.001)
```

```
reluLayer("Name","activation_33_relu")
convolution2dLayer([1 1],1024,"Name","res4d_branch2c","BiasLearnRateFactor",0)
batchNormalizationLayer("Name","bn4d_branch2c","Epsilon",0.001)];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
additionLayer(2,"Name","add_11")
reluLayer("Name","activation_34_relu")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
convolution2dLayer([1 1],256,"Name","res4e_branch2a","BiasLearnRateFactor",0)
batchNormalizationLayer("Name","bn4e_branch2a","Epsilon",0.001)
reluLayer("Name","activation_35_relu")
convolution2dLayer([3
3],256,"Name","res4e_branch2b","BiasLearnRateFactor",0,"Padding","same")
batchNormalizationLayer("Name","bn4e_branch2b","Epsilon",0.001)
reluLayer("Name","activation_36_relu")
convolution2dLayer([1 1],1024,"Name","res4e_branch2c","BiasLearnRateFactor",0)
batchNormalizationLayer("Name","bn4e_branch2c","Epsilon",0.001)];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
additionLayer(2,"Name","add_12")
reluLayer("Name","activation_37_relu")];
lgraph = addLayers(lgraph,tempLayers);
```

```
tempLayers = [
convolution2dLayer([1 1],256,"Name","res4f_branch2a","BiasLearnRateFactor",0)
batchNormalizationLayer("Name","bn4f_branch2a","Epsilon",0.001)
reluLayer("Name","activation_38_relu")
convolution2dLayer([3
3],256,"Name","res4f_branch2b","BiasLearnRateFactor",0,"Padding","same")
batchNormalizationLayer("Name","bn4f_branch2b","Epsilon",0.001)
reluLayer("Name","activation_39_relu")
convolution2dLayer([1 1],1024,"Name","res4f_branch2c","BiasLearnRateFactor",0)
batchNormalizationLayer("Name","bn4f_branch2c","Epsilon",0.001)];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
additionLayer(2,"Name","add_13")
reluLayer("Name","activation_40_relu")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
convolution2dLayer([1     1],512,"Name","res5a_branch2a","BiasLearnRateFactor",0,"Stride",[2
2])
batchNormalizationLayer("Name","bn5a_branch2a","Epsilon",0.001)
reluLayer("Name","activation_41_relu")
convolution2dLayer([3
3],512,"Name","res5a_branch2b","BiasLearnRateFactor",0,"Padding","same")
batchNormalizationLayer("Name","bn5a_branch2b","Epsilon",0.001)
```

```
reluLayer("Name","activation_42_relu")
convolution2dLayer([1 1],2048,"Name","res5a_branch2c","BiasLearnRateFactor",0)
batchNormalizationLayer("Name","bn5a_branch2c","Epsilon",0.001)];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
convolution2dLayer([1      1],2048,"Name","res5a_branch1","BiasLearnRateFactor",0,"Stride",[2
2])
batchNormalizationLayer("Name","bn5a_branch1","Epsilon",0.001)];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
additionLayer(2,"Name","add_14")
reluLayer("Name","activation_43_relu")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
convolution2dLayer([1 1],512,"Name","res5b_branch2a","BiasLearnRateFactor",0)
batchNormalizationLayer("Name","bn5b_branch2a","Epsilon",0.001)
reluLayer("Name","activation_44_relu")
convolution2dLayer([3
3],512,"Name","res5b_branch2b","BiasLearnRateFactor",0,"Padding","same")
batchNormalizationLayer("Name","bn5b_branch2b","Epsilon",0.001)
reluLayer("Name","activation_45_relu")
convolution2dLayer([1 1],2048,"Name","res5b_branch2c","BiasLearnRateFactor",0)
batchNormalizationLayer("Name","bn5b_branch2c","Epsilon",0.001)];
```

```
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
additionLayer(2,"Name","add_15")
reluLayer("Name","activation_46_relu")];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
convolution2dLayer([1 1],512,"Name","res5c_branch2a","BiasLearnRateFactor",0)
batchNormalizationLayer("Name","bn5c_branch2a","Epsilon",0.001)
reluLayer("Name","activation_47_relu")
convolution2dLayer([3
3],512,"Name","res5c_branch2b","BiasLearnRateFactor",0,"Padding","same")
batchNormalizationLayer("Name","bn5c_branch2b","Epsilon",0.001)
reluLayer("Name","activation_48_relu")
convolution2dLayer([1 1],2048,"Name","res5c_branch2c","BiasLearnRateFactor",0)
batchNormalizationLayer("Name","bn5c_branch2c","Epsilon",0.001)];
lgraph = addLayers(lgraph,tempLayers);
tempLayers = [
additionLayer(2,"Name","add_16")
reluLayer("Name","activation_49_relu")
globalAveragePooling2dLayer("Name","avg_pool")
fullyConnectedLayer(2,"Name","fc")
softmaxLayer("Name","fc_softmax")
classificationLayer("Name","classoutput")];
```

```
lgraph = addLayers(lgraph,tempLayers);

% clean up helper variable

clear tempLayers;

lgraph = connectLayers(lgraph,"max_pooling2d_1","res2a_branch2a");

lgraph = connectLayers(lgraph,"max_pooling2d_1","res2a_branch1");

lgraph = connectLayers(lgraph,"bn2a_branch2c","add_1/in1");

lgraph = connectLayers(lgraph,"bn2a_branch1","add_1/in2");

lgraph = connectLayers(lgraph,"activation_4_relu","res2b_branch2a");

lgraph = connectLayers(lgraph,"activation_4_relu","add_2/in2");

lgraph = connectLayers(lgraph,"bn2b_branch2c","add_2/in1");

lgraph = connectLayers(lgraph,"activation_7_relu","res2c_branch2a");

lgraph = connectLayers(lgraph,"activation_7_relu","add_3/in2");

lgraph = connectLayers(lgraph,"bn2c_branch2c","add_3/in1");

lgraph = connectLayers(lgraph,"activation_10_relu","res3a_branch2a");

lgraph = connectLayers(lgraph,"activation_10_relu","res3a_branch1");

lgraph = connectLayers(lgraph,"bn3a_branch2c","add_4/in1");

lgraph = connectLayers(lgraph,"bn3a_branch1","add_4/in2");

lgraph = connectLayers(lgraph,"activation_13_relu","res3b_branch2a");

lgraph = connectLayers(lgraph,"activation_13_relu","add_5/in2");

lgraph = connectLayers(lgraph,"bn3b_branch2c","add_5/in1");

lgraph = connectLayers(lgraph,"activation_16_relu","res3c_branch2a");

lgraph = connectLayers(lgraph,"activation_16_relu","add_6/in2");

lgraph = connectLayers(lgraph,"bn3c_branch2c","add_6/in1");
```

```
lgraph = connectLayers(lgraph,"activation_19_relu","res3d_branch2a");
lgraph = connectLayers(lgraph,"activation_19_relu","add_7/in2");
lgraph = connectLayers(lgraph,"bn3d_branch2c","add_7/in1");
lgraph = connectLayers(lgraph,"activation_22_relu","res4a_branch1");
lgraph = connectLayers(lgraph,"activation_22_relu","res4a_branch2a");
lgraph = connectLayers(lgraph,"bn4a_branch1","add_8/in2");
lgraph = connectLayers(lgraph,"bn4a_branch2c","add_8/in1");
lgraph = connectLayers(lgraph,"activation_25_relu","res4b_branch2a");
lgraph = connectLayers(lgraph,"activation_25_relu","add_9/in2");
lgraph = connectLayers(lgraph,"bn4b_branch2c","add_9/in1");
lgraph = connectLayers(lgraph,"activation_28_relu","res4c_branch2a");
lgraph = connectLayers(lgraph,"activation_28_relu","add_10/in2");
lgraph = connectLayers(lgraph,"bn4c_branch2c","add_10/in1");
lgraph = connectLayers(lgraph,"activation_31_relu","res4d_branch2a");
lgraph = connectLayers(lgraph,"activation_31_relu","add_11/in2");
lgraph = connectLayers(lgraph,"bn4d_branch2c","add_11/in1");
lgraph = connectLayers(lgraph,"activation_34_relu","res4e_branch2a");
lgraph = connectLayers(lgraph,"activation_34_relu","add_12/in2");
lgraph = connectLayers(lgraph,"bn4e_branch2c","add_12/in1");
lgraph = connectLayers(lgraph,"activation_37_relu","res4f_branch2a");
lgraph = connectLayers(lgraph,"activation_37_relu","add_13/in2");
lgraph = connectLayers(lgraph,"bn4f_branch2c","add_13/in1");
lgraph = connectLayers(lgraph,"activation_40_relu","res5a_branch2a");
```

```
lgraph = connectLayers(lgraph,"activation_40_relu","res5a_branch1");
lgraph = connectLayers(lgraph,"bn5a_branch1","add_14/in2");
lgraph = connectLayers(lgraph,"bn5a_branch2c","add_14/in1");
lgraph = connectLayers(lgraph,"activation_43_relu","res5b_branch2a");
lgraph = connectLayers(lgraph,"activation_43_relu","add_15/in2");
lgraph = connectLayers(lgraph,"bn5b_branch2c","add_15/in1");
lgraph = connectLayers(lgraph,"activation_46_relu","res5c_branch2a");
lgraph = connectLayers(lgraph,"activation_46_relu","add_16/in2");
lgraph = connectLayers(lgraph,"bn5c_branch2c","add_16/in1");
plot(lgraph);
options = trainingOptions("sgdm",...
"ExecutionEnvironment","auto",...
"InitialLearnRate",0.001,...
"MiniBatchSize",128,...
"Shuffle","every-epoch",...
"ValidationFrequency",30,...
"Plots","training-progress",...
"ValidationData",valTrain);
net = trainNetwork(ds,lgraph,options);
```