THE STUDY OF TRANSFER LEARNING IN SLEEP STAGE SCORING

การศึกษาการโอนถ่ายองค์ความรู้ที่คอมพิวเตอร์ใช้ในการวิเคราะห์การนอน

BY

MR. WORAWIT	**LUEWICHANA**	**5988018**
MR. CHAIYAKARN	**KHANAN**	**5988130**
MISS PRANPRIYA	**SHATSENA**	**5988190**

ADVISOR

DR. AKARA SUPRATAK

A Senior Project Submitted in Partial Fullfillment of the Requirement for

THE DEGREE OF BACHELOR OF SCIENCE (INFORMATION AND COMMUNICATION TECHNOLOGY)

Faculty of Information and Communication Technology
Mahidol University

2019

ACKNOWLEDGEMENTS

Foremost, we would like to express our sincere gratitude to our advisor Dr. Akara Supratak for the continuous support of our senior project for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped us in all the time of research and writing of this senior project on the topic ”The study of transfer learning in sleep stage scoring”. We could not have imagined having a better advisor.

Besides our advisor, we would like to thank the rest of our senior committee: Professor Dr. Peter Haddawy, and Assistant Professor Dr. Srisupa Palakvangsa Na Ayudhya for their encouragement, insightful comments, and hard questions. However, any errors are our own and should not tarnish the reputations of these esteemed persons.

Mr. Worawit Luewichana

Mr. Chaiyakarn Khanan

Miss Pranpriya Shatsena

THE STUDY OF TRANSFER LEARNING IN SLEEP STAGE SCORING

MR. WORAWIT LUEWICHANA 5988018 ITCS/B
MR. CHAIYAKARN KHANAN 5988130 ITCS/B
MISS PRANPRIYA SHATSENA 5988190 ITCS/B

B.Sc.(INFORMATION AND COMMUNICATION TECHNOLOGY)

PROJECT ADVISOR: DR. AKARA SUPRATAK

ABSTRACT

Machine learning and deep learning are a popular technique used in sleep stage scoring. Deep Learning is a technique that helps sleep experts to analyze and score the sleep stage. Deep learning requires a large size of train data. Therefore, every kind of research use transfer learning to help alleviate the problem of insufficient training data. Transfer learning has several things to concern, such as the factor that affect the performance of transferability.

This research proposes to study the factor that affects the performance of transferability in terms of performance gain. For the transferability process, a few research focus on the effects of transferability with insufficient training data. Moreover, there has not been any research that studies the performance decreasing when increasing the degree of dissimilarity between source data and transfer data in sleep stage scoring. In this research, we aim a method to study the transferability in three types, which are transferring deep learning model in the same dataset with the different channels, different datasets with the same channel, different datasets with different channels. This allows us to know whether the channel or the dataset has more impact on the performance gain.

According to our experimental results, transferability in sleep stage scoring improves performance in all cases. Transferability across the channel in the same dataset provides better performance than across the dataset. The dataset mismatch more impact on the transferability because of the difference in collecting protocol force the model to resample the data. Hence, the data might lose a significant part. These findings suggest that the performance of transferability is better when the collecting protocol of the base similar to the target dataset. Consequently, using the nearest channel with the transfer channel also help the model increase the performance.

KEYWORDS: SLEEP STAGE SCORING, DEEP LEARNING, TRANSFER LEARNING

76 P.

การศึกษาการโอนถ่ายองค์ความรู้ที่คอมพิวเตอร์ใช้ในการวิเคราะห์การนอน

นาย วรวิทย์ ลือวิชนะ 5988018 ITCS/B
นาย ชัยกาญจน์ ขนาน 5988130 ITCS/B
นางสาว ปรานต์ปริยา ษัฎเสน 5988190 ITCS/B

วท.บ. (เทคโนโลยีสารสนเทศและการสื่อสาร)

อาจารย์ที่ปรึกษาโครงการ: ดร. อัคร สุประทักษ์

บทคัดย่อ

ในปัจจุบันได้มีการนำเทคโนโลยีเข้ามาช่วยในการวิเคราะห์การนอนหลับให้มีประสิทธิภาพมากยิ่งขึ้น คือ การเรียนรู้เชิงลึก โดยที่การเรียนรู้เชิงลึกนั้นเข้ามาช่วยผู้เชี่ยวชาญด้านวิเคราะห์การนอนหลับเพื่อวิเคราะห์สัญญานการนอนหลับ แต่ทว่าเทคโนโลยีนี้ยังมีข้อจำกัดอยู่ คือ ฐานข้อมูลต้องมีขนาดใหญ่เพื่อใช้สอนโมเดลเหล่านั้น มีงานวิจัยมากมายกล่าวว่าการโอนถ่ายองค์ความรู้ช่วยแก้ปัญหาขนาดของชุดข้อมูลที่ใช้สอนโมเดลได้

ทางทีมพัฒนาต้องการที่จะศึกษาปัจจัยที่มีผลกระทบต่อประสิทธิภาพของการโอนถ่ายองค์ความรู้ในการวิเคราะห์การนอนหลับโดยวัดประสิทธิภาพจากประสิทธิภาพที่เพิ่มขึ้นในปัจจุบันนี้ยังไม่มีงานวิจัยใดที่ศึกษาเกี่ยวกับผลกระทบต่อประสิทธิภาพที่ลดลง เมื่อเพิ่มความแตกต่างระหว่างชุดข้อมูลตั้งต้นและชุดข้อมูลปลายทาง โดยทีมพัฒนาจะศึกษาการโอนถ่ายองค์ความรู้ ๓ ลักษณะคือ ๑. ชุดข้อมูลเดียวกันแต่ช่องสัญญาณแตกต่างกัน ๒. ชุดข้อมูลแตกต่างกันแต่ช่องสัญญาณเดียวกัน ๓. ชุดข้อมูลแตกต่างกันและช่องสัญญาณแตกต่างกัน หลังจากที่ทีมพัฒนาได้ทำการทดลองแล้ว ทีมพัฒนาจะทราบว่าปัจจัยใดที่มีผลต่อประสิทธิภาพข้อมูลมากที่สุด

หลังจากทำการทดลอง ทีมพัฒนาได้ทราบว่าการโอนถ่ายองค์ความรู้ในการวิเคราะห์การนอนหลับนั้นเพิ่มประสิทธิในทุกการทดลอง โดยการที่เราโอนถ่ายองค์ความรู้ข้ามช่องสัญญาณนั้นให้ประสิทธิภาพที่สูงกว่าการทำข้ามชุดข้อมูลด้วยช่องสัญญาณเหมือนกัน การค้นพบนี้ทำให้ทีมพัฒนารู้ว่าชุดข้อมูลส่งผลต่อประสิทธิภาพการโอนถ่ายองค์ความรู้มาก เนื่องจากการเก็บข้อมูลของแต่ละชุดข้อมูลนั้นมีความแตกต่างกัน จึงจำเป็นต้องเพิ่มหรือลดข้อมูลเพื่อให้สามารถทำการโอนถ่ายได้ ด้วยเหตุนี้ทีมพัฒนาจึงสามารถแนะนำได้ว่าการถ่ายโอนองค์ความรู้ควรเลือกจากองค์ประกอบของชุดข้อมูลตามด้วย เลือกช่องสัญญาณที่ใกล้เคียงกันมาถ่ายโอนจึงจะได้โมเดลที่ดีที่สุด

KEYWORDS: การวิเคราะห์การนอนหลับ, การเรียนรู้เชิงลึก, การโอนถ่ายองค์ความรู้
76 หน้า

CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 Motivation

Sleep plays an essential role in human health. People who are lack of sleep have a high chance of disease occur. The unhealthy sleep or disrupting while sleeping produces a high chance to be insomnia or sleep apnea [6]. Therefore, all disease that occurs from unhealthy sleep might make the people stress on the disease. This reason makes analyzing the sleep become a significant part of the medical. To understand how well people sleep, sleep experts use a set of signals recorded from different parts of the human body. This process is called analyzing sleep stage scoring. Accordingly, the doctor uses the resulting form analyzes the sleep stage scoring to improve the quality of patients checking.

Generally, sleep experts analyze the quality of sleep from the signal that recording from the sensor that attached to different parts of the body. However, machine learning is a helpful technique that sleep expert use to help them analyze the sleep stage. Nevertheless, deep learning is a popular choice in sleep stage scoring because deep learning is a neural network with many layers which can learn to extract representations that are useful for the sleep stage scoring from raw signals, making hand-engineering features no longer necessary. In the other hand, to make accurate deep learning model, it requires a large amount of dataset. A large sleep dataset is difficult to collect because it can be recorded only in the hospital or specific locations related to sleep analysis from the sleep experts, and it expensive to collect. Moreover, large sleep dataset for training a deep learning model is expensive to collect the dataset. To reduce expenses of collecting data, we need to utilize the data that existing annotated dataset. To utilize the existing annotated dataset is use the deep learning model which train from a large existing dataset transfer to train a small dataset.

Typically, the useful technique that helps the requiring size of training data for deep learning and insufficient training data is ”Transfer Learning”. Transfer learning technique has been employed in many fields such as computer vision and natural language processing (e.g. [7][8][9]). With transfer learning, the deep learning model, trained with the large dataset, is finetuned to target datasets. This process of transfer learning makes the model that train in the small size dataset has chances to provide high accuracy.

1.2 Problem Statement

In an ideal world, transfer learning is used to solve the lacking amount of data, which used to train deep learning model. Accordingly, transfer learning technique is used in a few research in the sleep stage scoring. It helps the sleep dataset, which has a small size of annotated data can access the deep learning model. As a result, all research focuses on the performance of transfer learning. Consequently, all research is not studied about the factor that affects the performance of the deep learning model, which come from the transfer learning technique. Hence, this research creates transfer learning in sleep stage scoring and quantifies the factor that affects the quality of transfer learning in sleep stage scoring model.

1.3 Objective of project

- To develop the transfer learning method which transfers the knowledge from the deep learning model that trained from one sleep stage dataset to train in another sleep stage dataset.
- To develop a new way to quantify how the performance of each step of transfer learning model decrease when both datasets increase the dissimilarity.

1.4 Scope of the project

This research is to study the impact factor that affects the transferability in sleep stage scoring. We scope three problems for the experiment. First, the effect of channel mismatch to the transferability. Second, the effect of the dataset mismatch to the transferability. Lastly, the combination of the channel mismatch and the dataset mismatch factor. We propose to find the characteristic of each effect. Hence, this research scope the transferability to address three questions as follow:

- Is transferring across the dataset with the same channel affect the transferability
- Is transferring in the same dataset with across the channel affect the transferring
- Is transferring across the dataset with different channel affect the transferability

1.5 Expected benefits

- To utilize existing annotated data, which have already been collected to creating the deep learning model with transfer learning technique.
  - The dataset that does not have enough annotated data to train the useful deep learning model
  - To help the doctor monitoring the sleep stage at home instead of monitoring at the hospital.
- To minimize the risk factor that affects transferring process.

CHAPTER 2
BACKGROUND

2.1 Sleep Stage Scoring

The sleep stage scoring is the summary from the signal corrected when the subject sleeps. Then, sleep experts normally look at multiple channels, such as polysomnography (PSG) or sleep study, as a diagnostic tool in sleep medicine. Sleep stage is measured in a sleep laboratory or sleep center to attach the electrodes to the head to take three types of measurement by using an electroencephalogram (EEG) to detect brain activity along with other systems to monitor eye movement, muscle activity, and heart rhythms. The process of collecting the sleep signal start after the patient meets the doctor and appointment the date. Consequently, the doctor uses several collected sleep signal channels to classify into five stages. The sleep stage scoring consists of five stages (W, N1, N2, N3, and REM). In this study, we are trying to build a model that relies only on a single channel which is suitable from the wearable device. To help the patients collect the sleep signal in their home and minimize the cost that they have to spend on collecting the sleep signal.

First of the sleep stage scoring is wake up (W). Wake up, or W represents the stage that the subject wakes up, or the subject is not sleeping yet. The second of sleep stage scoring is N1. N1 is a stage that the subject starts falls in sleep. This stage happens minutest because in general of people sleep who spend time in this stage a few times before changing to the next stage [4]. N1 is the stage in which every part of the body tries to relax and prepare for fall in sleep. Moreover, the N1 stage is the stage that easy to disrupt the subject to awake. Third of sleep stage scoring is N2. N2 is the stage that the subject falls in sleep more deeply than stage N1, so it's hard to disrupt the subject to awake. Furthermore, the brain signal goes to slow-moving with specific burst rapid activity. The pattern of brain signal is called sleep spindles [10]. Fourth of the sleep stage scoring is N3. N3 is the stage that the subject falls in sleep deepest. Moreover, night terrors, sleep talking, sleepwalking occur in the N3 stage. Last of the sleep stage scoring is REM. The REM stage is called in another name is the dreaming stage. The subject is disrupted to awake in this stage easier than N2 and N3 stage. All activity in the subject body in the REM stage more active than N2 and N3 stages.

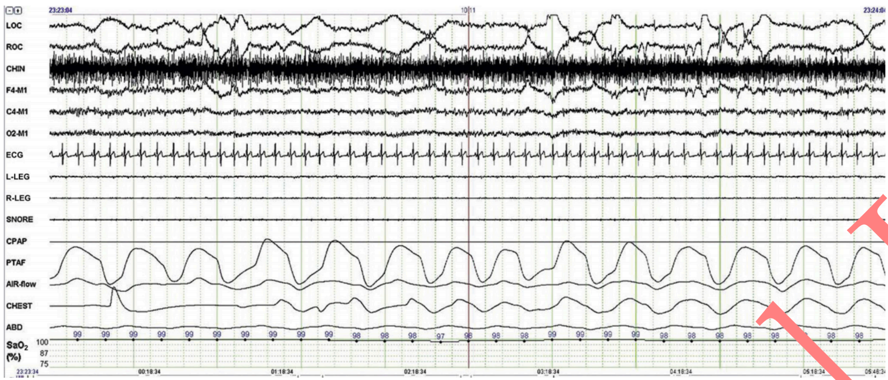


Figure 2.1: Wake up stage in sleep stage scoring

In each stage of sleep has an especially signal characteristic. To classify the sleep stage scoring the sleep expert have to understand well about the characteristics of each sleep stage signal. Figure 2.1 is the figure that shows all signal that sleeps expert collect. The signal in figure 2.1 contains stage wake in 60 seconds.

2.1.1 Sleep Stage Transition

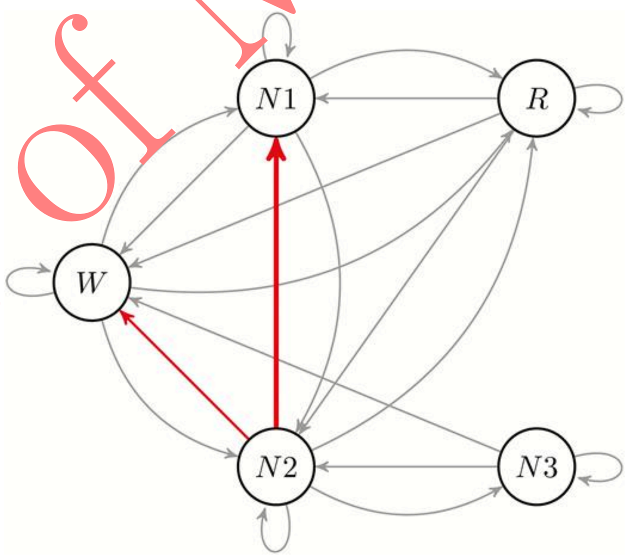


Figure 2.2: State diagram for sleep stage transitions [3]

The doctor uses the stage transition to help the classification sleep stage. The transition sleep stage is the one factor that the doctor uses to classify the people who have the problem of insomnia disorder and people who not. As figure 2.2, this figure shows the possible chance to transition from one stage to another stage. In some cases, the stages do not connect, meaning that direction occurs less than 50 percent of the experiment [3]. For example, Wake up stage or W hard to transition to the N3 stage.

From figure 2.2, the red line direction shows the transition that the insomnia disorder always changes in each night. If people always change the stage like red line direction, they are high chance to face with the insomnia disorder.

2.2 Deep Learning

Deep learning is the one class of machine learning techniques in Artificial Intelligence (AI) that copy the working of the human brain for processing a large amount of data and creating a pattern for use in decision making. Deep learning methods use neural network architectures and often referred to as deep neural networks. Deep learning models are trained by using a large amount of labelled data and neural network architectures which learn features through the data without using manual feature extraction.

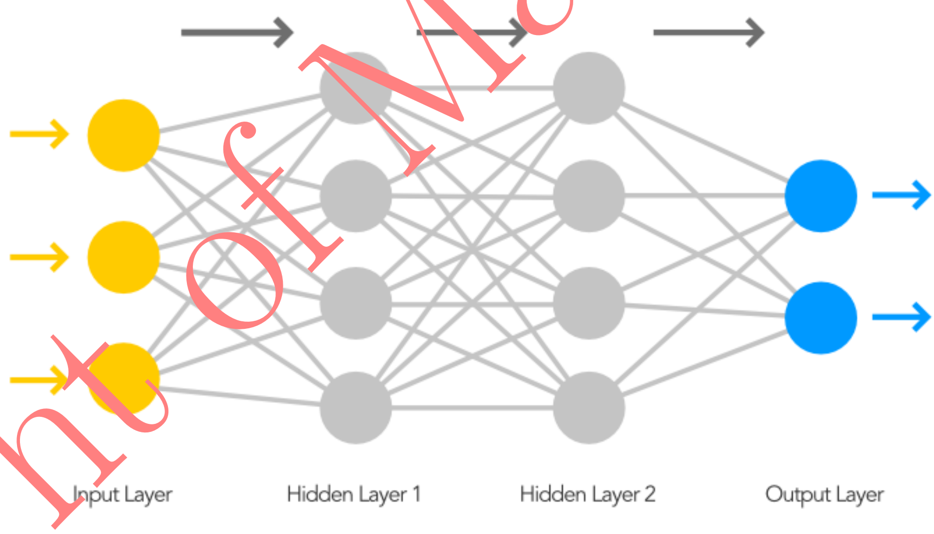


Figure 2.3: Procedure processing in Neural Networks

Neural Networks contains 3 types of layers which are input layer, hidden layer, output layer (figure 2.3). Each layer of Neural Networks has neurons interconnected to the neurons in the next layer. Input layer is all of the inputs that fed in the model and set up the weight of data. Hidden layer is a layer between the input and output layer. The hidden layer can have more than one layer. The layer, which is the hidden layer, used to calculate and learn the data that come from the input layer. The output layer is the last layer, which produces the output for the program. The neural network class which involve in this work are Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN).

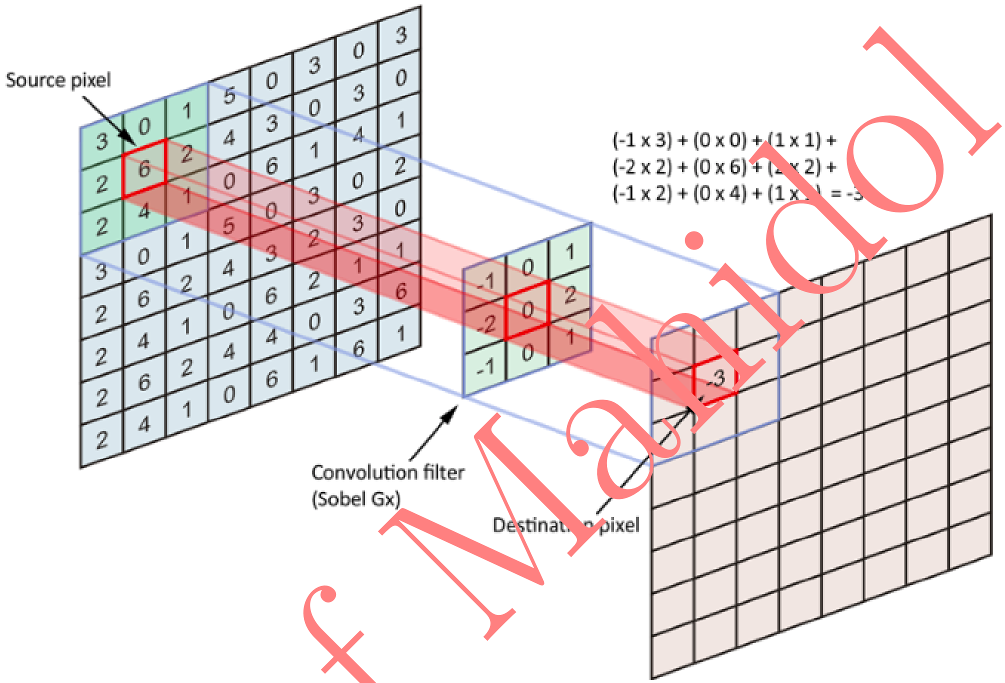


Figure 2.4: Convolutional Neural Network (CNN)

A convolutional neural network (CNN) is one type of artificial neural network, which used in image recognition and processing specifically thing. CNNs are made up of neurons that have learnable weights and biases. Each neuron of CNNs has to receive the input from the input layer or any layer. Consequently, CNN neuron use dot product and optionally follows it with a non-linearity [11]. The output of the CNNs comes from the input value dot product with the filters (Kernel). As figure 2.4, output values are performed the input value dot product with the filter. Convolutional neural networks is worked with several images processing [12].

Recurrent Neural Network (RNN) is one type of Neural Network where the output from the previous steps are fed as input in the current step. RNNs is used to process a sequence of values $vector x_1, x_2, x_3, ..., x_n$ [13]. Moreover, RNNs is used to solve the problem that the prediction needs the previous state to help the model decision. RNN is the hidden state which remembers some vital information in a sequence, which have to use in the future prediction. One reason that makes RNNs exciting is operating over a sequence of a vector. RNNs is worked with several texts processing [9].

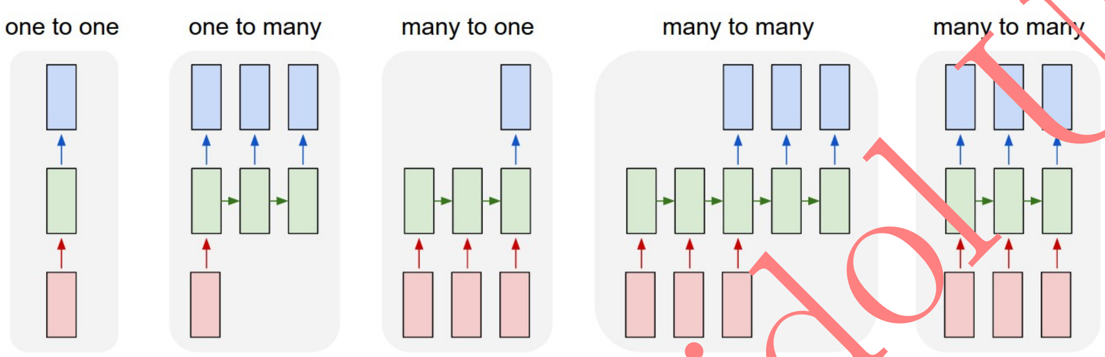


Figure 2.5: Sequences of a vector in Recurrent Neural Network (RNN)

From figure 2.5, it shows each type of sequences of a vector in Recurrent Neural Network (RNN). In each rectangle represent as a vector and arrows represent as a function. The red rectangle is an input vector, green as a vector hold the RNNs state and blue as an output vector. From left to right figure, first is one to one or Vanilla mode of processing without RNN, from fixed-sized input to fixed-sized output. Second is sequence output. The third is a sequence input. Fourth is sequence input and sequence output. The fifth is synced sequence input and output.

For this research, we implement the Deep learning model that contains the Recurrent Neural Network (RNN). The RNNs that we implement in the fifth of figure 4, which is synced sequence input and output. The reason that makes us have to use this technique is the sleep stage signal contain a long time of sleep. Consequently, sleep stage signal is divided into 30 seconds. Therefore, to predict the sleep stage scoring has to use the previous state to help the prediction. It makes the RNN is suitable to use in sleep stage scoring. In the sleep stage scoring field, is has numerous research that applies the RNN to predict the sleep stage [4][14].

2.3 Transfer Learning

Transfer learning is a machine learning method where a model developed for a task is reused as the starting point for a model on another similar task. Transfer learning is the idea of overcoming isolated learning to solve the specific task and utilizing the knowledge acquired from the task to solve related ones. It is a popular approach in deep learning where pre-trained models are used as the starting point on computer vision and natural language processing tasks. Consequently, it given the vast compute and time resources required to develop neural network models on these problems and from the huge jumps in the skill that they provide on related problems. Transfer learning aims to extract the knowledge from one or more source task and applies the knowledge that gets from them to a target task for examples, knowledge of the learning to recognize apples might help to recognize pear,and knowledge of learning to play the electric organ may help to facilitate learning the piano. The motivation of transfer learning comes from the human that can intelligently apply knowledge learned previously to solve the new problems faster, saving time and can be a better solution with getting the high performance. In addition, it uses annotated data less than build a new one.

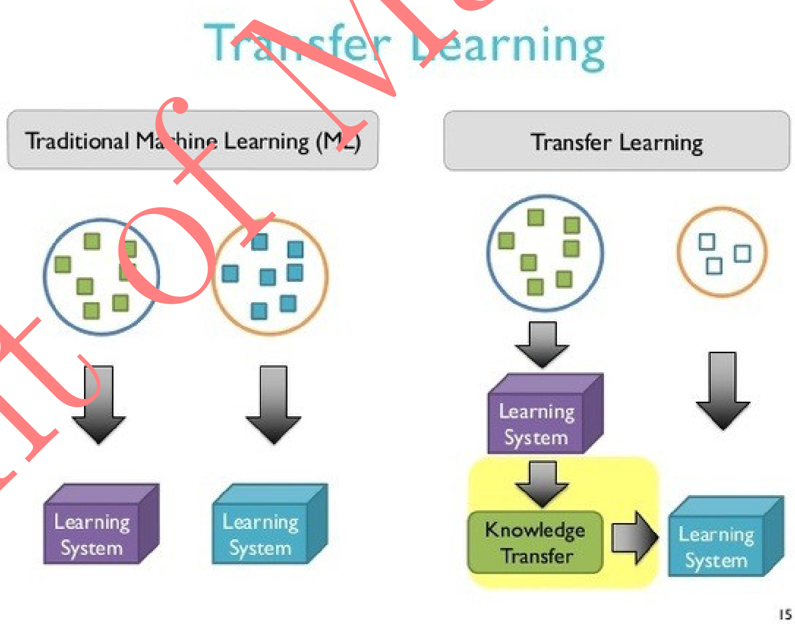


Figure 2.6: Transfer Learning

In figure 2.6 show the difference between traditional machine learning and transfer learning. For the traditional machine learning, when each dataset approach to the learning system, it get knowledge which separates certainly in each dataset. However, datasets that used in traditional machine learning should have a large size to get better performance of the model. In transfer learning, when first dataset approach to the learning system, it get some knowledge from learning system, and then target dataset use the knowledge gained from the first dataset transfer to the learning system of target dataset to get the performance of model close to the first dataset but using the smaller size of the dataset.

2.4 Literature Review

2.4.1 Deep Learning in Sleep stage scoring

In recent years, deep neural networks are used to learning and prediction in computer vision and sleep stage scoring. Deep neural networks provide impressive results, such as providing accuracy of prediction sleep stage scoring after trained more than 80 percent [4][14][15][16]. One significant key that makes these kinds of research success is the availability of a large amount of annotated sleep data. Moreover, deep neural networks are perfectly deal with an imbalance class, which always occur in sleep dataset. Thus, the imbalance class in sleep dataset is not a problem of the deep neural networks. For more details of each literature are provided below.

2.4.1.1 Automated Sleep Stage Scoring with Sequence to Sequence Deep Learning Approach (SleepEEGNet) [14]

Signals that measure from the subject have several useful channels to be a training set of the deep learning model. This research uses electroencephalogram (EEG) to be the first channel used to train and evaluate the performance. This research call itself as SleepEEGNet. SleepEEGNet is composed of convolutional neural networks (CNNs) to get the time, frequency information, and a sequence to sequence model to capture complex long short-term between sleep epoch. In addition, SleepEEGnet tries to develop a new way to reduce the effect of the imbalanced class problem. Therefore, the purpose of SleepEEGnet is scoped in three things which are using BiRNN, applying novel loss functions to address the imbalanced class problem, using raw single-channel EEGs as its input without using any handcrafted features and significant signal preprocessing such as filtering or noise removal methods.

In the conclusion of SleepEEGnet research, they can provide an automated sleep stage scoring with high accuracy. Table 2.1 shows the experiment result of SleepEEGnet. It has shown the accuracy that can gain the highest result from 20-fold in EEG channel from the new way loss calculation.

Table 2.1: The 20-fold experiment from SleepEEGnet research

Method	Dataset	CV	EEG Channel	Overall Performance			Pre-class Performance(F1)				
				ACC	MF1	k	W	N1	N2	N3	REM
SleepEEGNet	Sleep-EDF-13	20-fold CV	Fpz-Cz	**84.26**	**79.66**	**0.79**	**89.19**	**52.19**	**86.77**	**85.13**	**85.02**
Supratak et al.	Sleep-EDF-13	20-fold CV	Fpz-Cz	82.0	76.9	0.76	84.7	46.6	85.9	84.8	82.4
Tsinalis et al.	Sleep-EDF-13	20-fold CV	Fpz-Cz	78.9	73.7	-	71.6	47.0	84.6	84.0	81.4
Tsinalis et al.	Sleep-EDF-13	20-fold CV	Fpz-Cz	74.8	69.8	-	65.4	43.7	80.6	84.9	74.5
SleepEEGNet	Sleep-EDF-13	20-fold CV	Pz-Oz	**82.83**	**77.02**	**0.77**	**90.27**	**44.64**	**85.74**	**81.55**	**82.88**
Supratak et al.	Sleep-EDF-13	20-fold CV	Pz-Oz	79.8	73.1	0.72	88.1	37	82.7	77.3	80.3
SleepEEGNet	Sleep-EDF-18	10-fold CV	Fpz-Cz	<u>80.03</u>	<u>73.55</u>	<u>0.73</u>	<u>91.72</u>	<u>44.05</u>	<u>82.49</u>	<u>73.45</u>	<u>76.06</u>
SleepEEGNet	Sleep-EDF-18	10-fold CV	Fpz-Cz	<u>77.56</u>	<u>70.00</u>	<u>68.94</u>	<u>90.26</u>	<u>42.21</u>	<u>79.71</u>	<u>94.83</u>	<u>72.19</u>

2.4.1.2 Automated Sleep Stage Scoring of the Sleep Heart Health Study Using Deep Neural Networks [15]

The purpose of this research is to be an Automated PSG scoring has the potential to reduce the human labour costs and the variability inherent to this task. This research uses large sleep dataset, which has around 42,650 hours. The model which this research provide was composed of spectrograms in the input layer feeding into CNN layers and LSTM layer and RNN to achieve high accuracy. All of the technique that is used to build a model take advantage of sequential construction of PSG data. Moreover, all model that is trained in this research was evaluated on a subset of the dataset. The model also used to transferability to another channel of the sleep stage signal for collecting the performance in figure 2.7

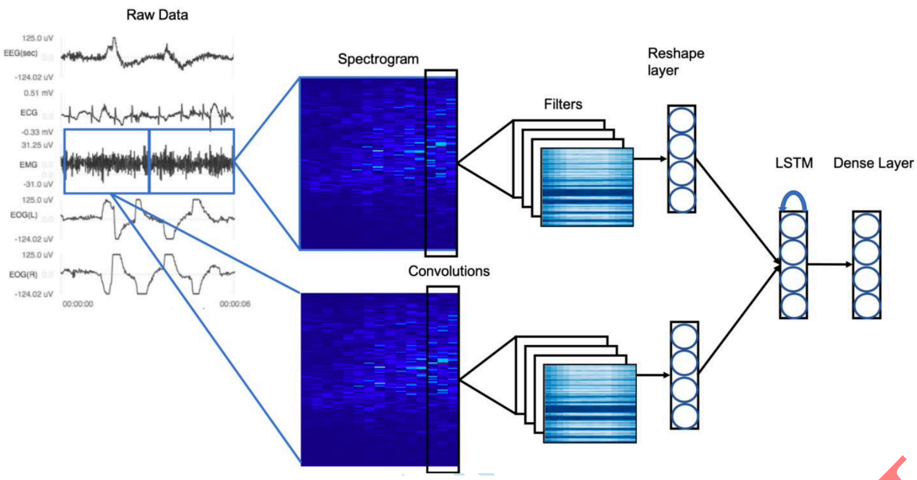


Figure 2.7: Spectrogram and filter of Automated Sleep Stage Scoring of the Sleep Heart Health Study Using Deep Neural Networks research

2.4.1.3 End-to-End Hierarchical Recurrent Neural Network for Sequence-to-Sequence Automatic Sleep Staging (SeqSleepNet) [16]

For the purpose of this work is trying to solve the sequence-to-sequence classification problem which receives a sequence of multiple epochs as input and classifies all of their labels at once. They propose a hierarchical recurrent neural network architecture named SeqSleepNet running on multichannel time-frequency image input to tackle the problem. SeqSleepNet is composed of three main components. First, parallel filter-bank layers for preprocessing. Second, an epoch-level bidirectional RNN coupled with the attention mechanism for short-term sequential modelling. Third, a sequence-level bidirectional RNN for long-term sequential modelling. The network was trained in an end-to-end training which has been proven many times in various domains. The public dataset used for evaluation is Montreal Archive of Sleep Studies (MASS) dataset which is considered as a large open-source dataset that was pooled from different hospital-based sleep laboratories.

2.4.1.4 Model for Automatic Sleep Stage Scoring Based on Raw Single-Channel EEG (DeepSleepNet) [4]

This research goal is to make the automatic sleep stage scoring based on raw single-channel EEG instead of machine learning with hand-engineered features. This research is called DeepSleepNet. DeepSleepNet consists of two main parts. First is representation learning. These phases use two CNNs. The first CNN is used to capture temporal information. First CNN is the small filter. Second CNN is used to capture frequency information which is the large filter. Another phase is the sequence residual learning. For this phase use two bidirectional-LSTMs to learn stage transition rule. This bidirectional-LSTMs estimate the next possible sleep stage. This bidirectional-LSTMs is provided to learn into two ways which are forward and backward. In contrast, both directions are not connected. From these two techniques, DeepSleepNet provides the algorithm to automate sleep stage scoring without hand-engineered follow AASM manual.

Table 2.2: DeepSleepNet experiment result comparing with another research

Method	Dataset	EEG Channel	Test Epochs	Overall Metrics			Pre-class F1-Scope (F1)				
				ACC	MF1	k	W	N1	N2	N3	REM
Non-independent Training and Test Sets											
Ref. [17]	Sleep-EDF	Fpz-Cz	[illegible]	[illegible]	[illegible]	-	77.3	46.5	**94.9**	72.2	**91.8**
Tsinalis et al.	Sleep-EDF	Pz-Oz	[illegible]	[illegible]	77	**0.86**	**97.8**	30.4	89	**85.5**	82.5
Tsinalis et al.	Sleep-EDF	Pz-Oz	7569	90.8	**80**	0.85	96.9	**49.1**	89	84.2	81.2
Independent Training and Test Sets											
Ref.[18]	Sleep-EDF	Fpz-Cz	[illegible]	78.9	73.7	-	71.6	**47.0**	84.6	84.0	81.4
Ref. [19]	Sleep-EDF	Fpz-Cz	37022	74.8	69.8	-	65.4	43.7	80.6	**84.9**	74.5
DeepSleepNet	Sleep-EDF	Fpz-Cz	41950	**82.0**	**76.9**	**0.76**	84.7	46.6	**85.9**	84.8	**82.4**
DeepSleepNet	Sleep-EDF	Pz-Oz	41950	82.0	76.9	0.76	**88.1**	37	82.7	77.3	80.3
Ref. [20]	MASS	F4-EOG (Left)	59066	85.9	80.5	-	84.6	56.3	**90.7**	**84.8**	86.1
DeepSleepNet	MASS	F4-EOG (Left)	59066	**86.2**	**81.7**	**0.80**	**87.3**	**59.8**	90.3	81.5	**89.3**

Sleep stage scoring, which predicted by deep learning, is acceptable. The technique of deep learning has two significant techniques to create the model for sleep stage scoring. The first technique is converting the signal to picture before training the model. Nevertheless, this technique is not to perform the accurate concept of deep learning. Deep learning should learn the feature by itself. The second technique is learning from the signal directly. This technique provides a better result than the first technique. This technique develops deep learning to learn the features of the signal. The 2.4.1.1 and 2.4.1.3 are using the first technique which converts to picture before train the deep learning. In contrast, 2.4.1.2 and 2.4.1.4 are using the second technique which uses deep learning to learn it.

From all research technique review, we decided to use the architecture from DeepSleepNet [4]. We use DeepSleepNet architecture to be the start of our research. Before developing transfer learning method, we train the deep learning model as same as DeepSleepNet.

2.4.2 Transfer Learning in the Medical field

Deep learning provides satisfying results in several fields (e.g. medical image classification). In contrast, deep learning technique requires a large amount of annotated data. Amount of annotated data is the significant limitation of deep learning. Therefore, transfer learning technique is created to solve this problem. Transfer learning technique used to help the dataset that does not have enough of annotated data to create useful deep learning model. There has been a range of work trying to develop a method to transfer a useful model to train in a new dataset, which does not have enough of annotated.

Nevertheless, there are a few numbers of studies trying to develop deep learning with transfer learning technique, for example [21]. Sleep stage scoring is the field that is difficult to collect the annotated data. Furthermore, the public of sleep dataset should use to help the new dataset [7][8].

2.4.2.1 Lung Nodule Classification via Deep Transfer Learning in CT Lung Images [7]

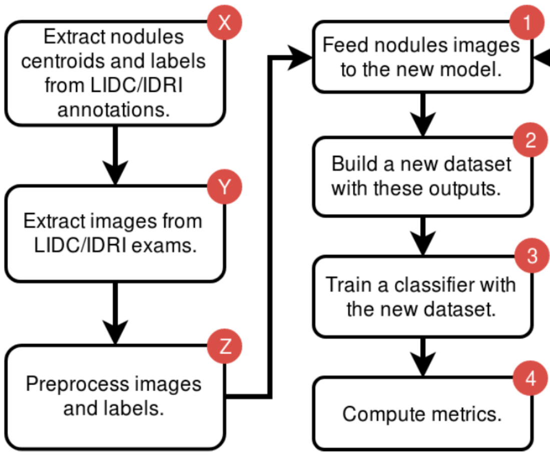


Figure 2.8: The step to pre-process of LIDC/IDRI dataset

Lung nodule classification via deep transfer learning in CT lung images is the case that success with transfer learning in the medical part. It has been developed to detect lung cancer by using several computer-aided diagnosis systems to decrease the rate of deaths. It was illustrated of deep transfer learning from the non-medical image, which extracts representational imaging biomarkers at the macroscopic level in the chest CT image. They classify lung nodule in the CT lung images, which extract in deep features. Then, they use Lung Image Database Consortium image collection (LIDC-IDRI) dataset, which has 11 different Convolution Neural Network (CNN) models to pre-trained on ImageNet Large Scale Visual Recognition Challenge (ILSVRC). Next, they apply five classification methods consists of Naive Bayes classification, Multilayer Perceptron (MLP), Support Vector Machine (SVM), Near Neighbors (KNN) and Random Forest (RF) for the extracted feature set. Last, compare the results which related five evaluation metrics: Accuracy (ACC), Area Under the Curve (AUC), True Positive Rate (TRP), Precision (PPV), and F1-Score. Accordingly, the result represented into two parts. First, they compare extractors through the evaluation metrics and concern the best combination of deep extractor and classifier. Second, it makes a comparison between the best combination with other approaches. Therefore, they found the best models in the LIDC/IDRI dataset between ResNet50 and SVM RBF that achieving an AUC of 93.19%, ACC of 88.%, F-score of 78.83%, TRP of 85.38% and PPV of 73.48%. Transfer learning improves the relevant strategy to extract the image biomarkers for lung nodule malignancy classification in chest CT images.

Table 2.3: The combination between RESNET50 deep extractor and SVM classifier.

Approach	Number of nodules	ACC(%)	AUC(%)	F-Score(%)	TRP(%)	PPV(%)
Our Approach	1536	88.41	93.19	78.83	85.38	73.48
Shen et al.	1375	86.80	-	-	-	-
Shen et al.	1375	87.14	93.00	-	77.00	-
Aerts et al.	1375	83.21	89.00	-	87.00	-
Han et al.	1356	-	92.70	-	-	-
Hussein et al.	1340	91.26	-	-	-	-
Zhu et al.	1004	90.44	-	-	-	-
Dhara et al.	891	-	95.05	-	8[illegible].73	-
Kang et al.	776	95.41	99.00	-	95.6[illegible]	-
Wei et al.	746	85.20	-	85.80	8[illegible].80	85.95
Wei et al.	366	91.00	98.40	91.[illegible]0	92.[illegible]0	92.30
Sergeeva et al.	321	81.30	89.60	76.50	70.90	83.00
Ma et al.	157	82.70	-	-	80.00	-

2.4.2.2 Transfer Representation Learning for Medical Image Analysis [8]

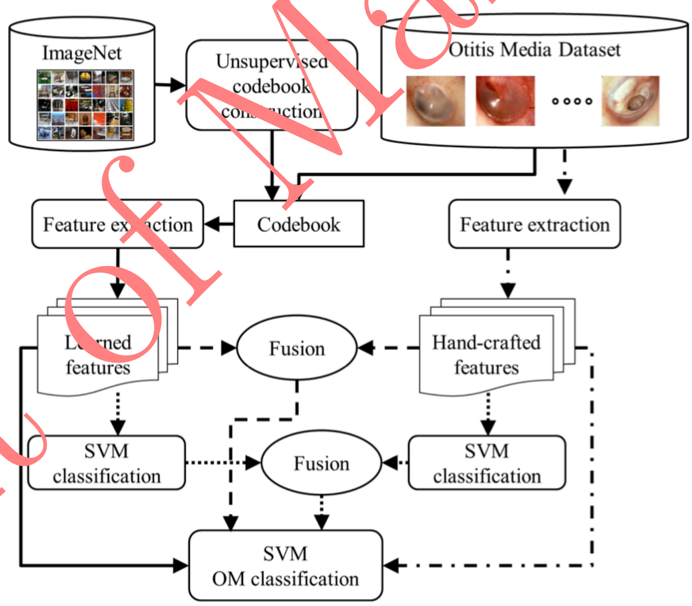


Figure 2.9: Transfer Representation Learning for Medical Image Analysis flow

Otitis media (OM) is an inflammation or infection of the middle ear and consumes significant medical resources each year. They scope the project, which consists of two significant challenges are labelled data scarcity and medical domain knowledge shortage.

They represent learning on an entirely unrelated image from ImageNet, which is the largest image dataset of daily objects. In this work, they do not use deep learning to training an OM classifier. So, they use unsupervised concerning OM of CNN to perform transfer representation learning. In the first step, they performed unsupervised codebook construction from large ImageNet dataset by using deep Convolution Neural Network (CNN). Next, encoding OM images by using the codebook. To move the image from the input layer to the output layer through the inner layers. So, each layer is a weighted combination of the previous layer. Finally, supervised learning by using transfer-learned of OM images from the fifth, sixth and seventh layer. Nevertheless, they do not extract OM features from the eighth layer because it only produces the probability of class prediction. They were using transfer-learned feature vectors to collect the image by training Support Vector Machine (SVM) classifier. Moreover, using transfer-learned feature fusion to improve classification accuracy. Then, they combined features hand-crafted by human-heuristic with features learned from the codebook. The first features fusion is combined transfer-learned and hand-crafted OM features to feature vectors and then deploy the supervised learning to train the SVM classifier. The second features fusion is to use two layer classifier fusion structure. So, in the first layer, they train different classifiers and different features set. Then, they combine all output from the first layer to train classifier in the second layer. The best accuracy achieved by using a human-heuristic method. Human-heuristic requires domain expertise to specify the features. Sometimes they may not be able to capture the most discriminative characteristics. The achieved detection accuracy is 88.5% (89.63% in sensitivity and 86.9% in specificity). Therefore, they represent to analyze medical data in transfer learning and consider potential medical applications.

Table 2.4: OM classification experimental results

Method	**Measures**			
	Accuracy	***Sensitivity***	***Specificity***	***F1-score***
Heuristic w/ seg	80.11%	83.33%	75.66%	0.822
Heuristic w/o seg	76.19%	79.38%	71.74%	0.79
Transfer w/ seg (pool5)	87.86%	89.72%	86.26%	0.89
Transfer w/o seg (pool5)	88.37%	89.16%	87.08%	0.894
Transfer w/ seg (fc6)	87.58%	89.33%	85.04%	0.887
Transfer w/o seg (fc6)	**88.5%**	89.63%	86.9%	0.895
Transfer w/ seg (fc7)	85.6%	87.5%	82.7%	0.869
Transfer w/o seg (fc7)	86.9%	88.5%	84.9%	0.879
OM-trained codebook	71.8%	95.06%	41.66%	0.818
Feature fusion	89.22%	90.08%	87.81%	0.90
Classifier fusion	**89.87%**	89.54%	90.2%	0.898

2.4.2.3 Towards More Accurate Automatic Sleep Staging via Deep Transfer Learning [21]

This work purpose to use transfer learning to handle the channel miss-match between problem, transferring from large dataset to small dataset without a guarantee that bot dataset has the same setup. In this study, using Montreal Archive of Sleep Studies (MASS) dataset to be the first training part, because this dataset is large enough to train the accurate model. Moreover, this research try to develop transfer learning scenarios, ranging from one channel to multi-channel.

Table 2.5: The table that contains the performance from each channel, and each training type.

		EEG·EOG to EEG·EOG			EEG to EEG			EOG to EOG			EEG to EOG		
		Acc.	MF1	k	Acc.	MF1	k	Acc.	MF1	k	Acc.	MF1	k
Sleep-EDF-SC	**FT SeqSleepNet+**	84.3	77.7	0.776	**85.2**	**79.6**	**0.789**	**81.7**	**75.1**	**0.737**	**80.0**	72.3	**0.709**
	FT DeepSleepNet+	**84.6**	**79.0**	**0.782**	84.4	78.8	0.781	79.8	73.4	0.713	79.4	**72.8**	0.707
	Scratch SeqSleepNet+	82.2	74.2	0.744	82.2	74.1	0.746	78.5	68.3	0.688	78.5	68.3	0.688
	Scratch DeepSleepNet+	81.9	75.2	0.744	80.8	74.2	0.731	75.9	66.9	0.652	75.9	66.9	0.652
	DT SeqSleepNet+	72.0	62.1	0.601	81.2	74.6	0.733	67.2	59.1	0.530	51.1	42.5	0.300
	DT DeepSleepNet+	70.2	59.8	0.586	74.2	66.9	0.651	54.1	41.9	0.396	39.7	35.8	0.235
Sleep-EDF-ST	**FT SeqSleepNet+**	**81.0**	**76.7**	**0.732**	81.0	**77.5**	0.734	**80.4**	**76.5**	**0.722**	**79.6**	**75.2**	**0.710**
	FT DeepSleepNet+	80.1	76.6	0.721	**81.5**	**77.5**	**0.738**	77.4	74.1	0.682	76.0	71.4	0.661
	Scratch SeqSleepNet+	79.6	74.8	0.711	76.5	70.6	0.667	78.6	71.6	0.693	78.6	71.6	0.693
	Scratch DeepSleepNet+	73.7	67.6	0.629	72.4	64.6	0.603	70.0	65.9	0.574	70.0	65.9	0.574
	DT SeqSleepNet+	73.1	64.2	0.615	80.5	75.6	0.722	67.2	59.4	0.531	56.3	48.4	0.363
	DT DeepSleepNet+	71.6	65.4	0.600	66.7	61.3	0.541	70.0	63.3	0.586	35.1	31.0	0.116
Surrey-cEEGGrid	**FT SeqSleepNet+**	**82.3**	**71.1**	**0.752**	**75.3**	**60.8**	**0.650**	**82.6**	**72.2**	**0.758**	**81.9**	**71.2**	**0.749**
	FT DeepSleepNet+	77.8	66.5	0.687	58.2	42.8	0.391	77.5	66.6	0.682	81.7	70.5	0.745
	Scratch SeqSleepNet+	81.5	66.4	0.739	71.9	55.2	0.597	81.3	67.8	0.737	81.3	67.8	0.737
	Scratch DeepSleepNet+	65.4	57.4	0.534	42.5	30.3	0.195	69.1	60.0	0.579	69.1	60.0	0.579
	DT SeqSleepNet+	19.4	14.6	0.051	10.6	9.1	−0.015	24.3	20.5	0.085	24.1	[illegible]	0.090
	DT DeepSleepNet+	38.3	11.7	0.020	38.4	11.6	0.012	39.3	25.4	0.214	38.9	25.4	0.195
	Random Forests2 [11]	72.0	-	0.600	70.0	-	0.580	-	-	-	-	-	
Surrey-PSG	**FT SeqSleepNet+**	**82.2**	**73.7**	**0.753**	**80.3**	**71.3**	**0.729**	**80.3**	**72.3**	**[illegible]**	79.8	70.8	0.722
	FT DeepSleepNet+	78.3	69.5	0.699	79.8	69.9	0.718	79.5	71.3	0.716	**80.4**	**71.2**	**0.729**
	Scratch SeqSleepNet+	79.9	67.6	0.719	75.8	62.3	0.661	79.4	67.8	0.713	79.4	67.8	0.713
	Scratch DeepSleepNet+	75.1	68.4	0.661	69.4	60.9	0.588	70.6	62.4	[illegible]	[illegible]	62.4	0.597
	DT SeqSleepNet+	32.8	20.8	0.177	18.6	14.8	0.041	15.3	13.0	-0.001	10.6	9.3	-0.026
	DT DeepSleepNet+	40.3	25.9	0.234	39.0	26.3	0.221	[illegible]	20.9	0.151	27.6	15.6	0.046
	Random Forests [11]	77.5	-	0.675	-	-	-	-	-	-	-	-	-

From Table 2.5, this is the result of the Transfer Learning process that use the MASS dataset to be the source dataset. This research aims to develop the result table from two types of training, which are scratch training, direct transfer. Direct transfer contains finetuning and direct transfer. Finetuning is the transfer that tuning the model while training the new dataset. Direct transfer is the transfer process that trains the new dataset without tuning more than the source dataset. At the last part of this research, transfer learning method provides a better result than every pure research train from a small dataset.

2.5 How different between our work and other work.

From the literature review section, there are a few numbers that develop transfer learning in sleep stage scoring. Transfer learning is useful for the small dataset that prefers to use the deep learning model in sleep stage scoring. The main advantage of transfer learning is improving the accuracy of deep learning model that trains from the small dataset.

This research introduces software of quantifying the transferability of deep learning features in sleep stage scoring. This research is different from the existing work, which develops transfer learning method, then evaluate the result of transfer learning. This research aims to find a new way to quantify how the performance decrease when the source and target dataset increase the dissimilarity. Consequently, sleep expert can collect the signal, which sets support transfer learning. Moreover, this research develops the web base for the people who prefer to try transfer learning process.

CHAPTER 3
METHODOLOGY

This study investigates the transferability of deep learning model using different settings in sleep datasets. In this section, we describe the model architecture, which used for transferability. Consequently, we present the transfer learning flow, which used to answer three questions in section 1.4.

Deep learning model is the significant architecture which affects the transferability directly. The model architecture, which we use, has to achieve acceptable accuracy with every dataset. The range of accuracy, which the model provides in each dataset, should be close to every dataset. In this part, we use to explain the deep learning model architecture, which includes two crucial parts. Two crucial parts are representation learning and sequence residual learning.

3.1 Model Architecture

Deep learning model is the significant part before working on the transfer learning process. In this research, we use the DeepSleepNet model architecture [4] to train the data before transferring the model. Deep learning model architecture is separated into two parts which are representation learning and sequence residual learning. The first part is used to extract the time-invariant features. The second part is used to encode the temporal information (i.e. stage transition rules)

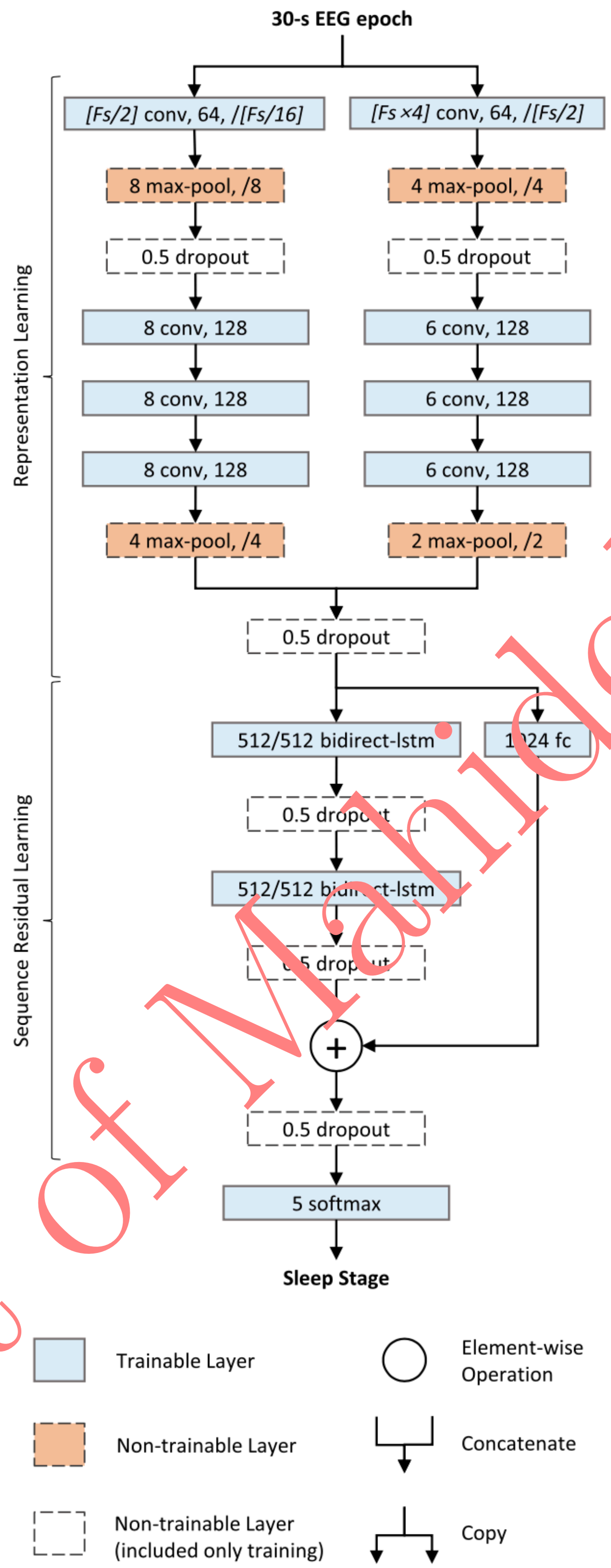


Figure 3.1: DeepSleepNet model architecture [4]

3.1.1 Representation Learning

In representation learning, this phase uses two CNNs. The first CNN is used to capture temporal information (i.e. signal pattern). The second CNN is used to capture frequency information. Both CNN consist of four convolutional layers and two max-pooling layers. All convolutional layer consists of three operations, which are 1D-convolutional, batch normalization [22], and applying the Rectified Linear Unit (ReLU) activation. The Rectified Linear Unit (ReLU) activation is performed follow formula 3.1.

$$relu(x) = max(0, x) \tag{3.1}$$

At the last of representation learning, this phase concatenates both outputs of CNN. The concatenates output is the input of the sequence residual learning.

3.1.2 Sequence Residual Learning

Sequence residual learning is used to learn the residual part. This phase consists of two main parts, which are bidirectional and a shortcut connection. This phase uses two main parts to estimate the next possible of sleep stages based on the previous stage. The bidirectional LSTM technique uses two LSTMs process which are forward and backward [23]. In contrast, the result from the forward and backward are not connected to each. The model enables to add the temporal information that learns from the previous into feature extracted from the CNN. Furthermore, it uses a fully-connected layer in the shortcut connection to transform the feature from CNNs into a vector.

3.2 Transfer Learning in Sleep Stage Scoring

After choosing and understanding the model architecture, we use this section to illustrate the definition of the transferability variable (i.e. source model, target model). Consequently, we describe the flow of transfer learning in our research. The flow of transfer learning is the flow, which we design to find the answer three questions in section 1.4. Finally, we introduce the configuration, which use to find the explanation of each question.

3.2.1 Source Model

Source model is the trained model from a considerable size of train data. The source model has to be precisely model because we use the source to be the starting point of the transferability. Our research trains several source model for transferability. Each source model is trained from the different configuration (i.e. sleep channel, a sampling rate of the channel, datasets). All source model, which we use to be the starting point of other models, has to be accurate and trained from DeepSleepNet [4] architecture.

3.2.2 Target Model

Target models are trained from insufficient train data. The reason that target model train with 20% of the public dataset is the simulation of insufficient training data. In each step of the experiment, we have two target model. First is the target model that trains with the random weight of deep learning model. Second is the target model that uses the source model as a starting point of the training process. Two targets model can be used to find the performance of the transferability (Evaluation via Performance Gain). We use many target model to evaluate the performance of transferability. Hence each target model is trained from different configuration (i.e. sampling rate, epoch duration).

3.2.3 Flow of Transfer Learning

We assemble the source and target model to be transfer learning flow. Transfer learning flow is the procedure of our experiments. We process the experiments follow the flow for finding the result. Consequently, we use the result to investigate the effect of transferability. We use this section to describe the element in the transfer learning flow and define the variable for each step of transferability.

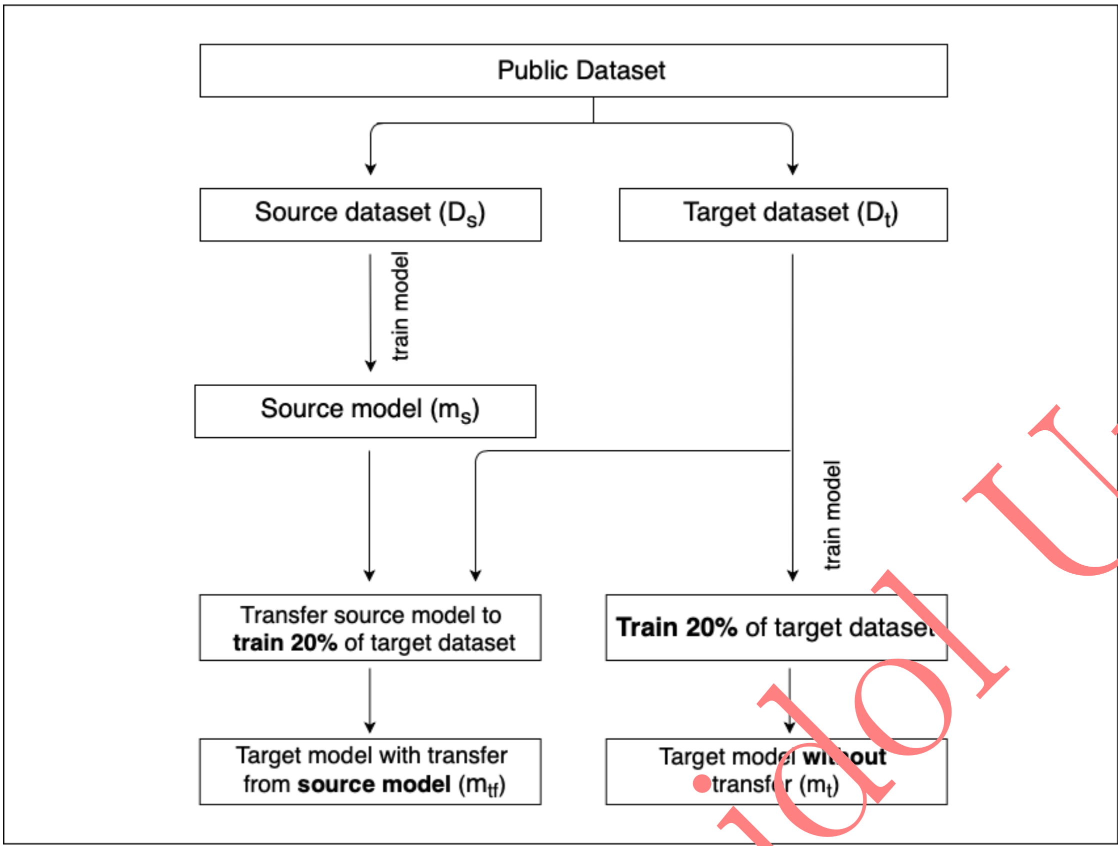


Figure 3.2: Flow of Transfer Learning in our research

According to the transfer learning flow in figure 3.2, the first box is the sleep signal public dataset. We use public datasets for experiments and simulation the insufficient train data. The reason for the insufficient train data simulation is the sleep signal from a wearable device. The wearable device collecting protocol cannot collect the sleep signal enough for generally training the deep learning model because the wearable device is used at house. Due to this fact, it makes the wearable device does not collect sleep signal enough. We define that 20% of public dataset to be train and valid is the insufficient training data. Consequently, 80% of public data be the test set. Let denote $P = (p_1, p_2, ..., p_n)$, where P denote the public dataset and n denote the number of the public dataset. Moreover, let denote $Ch = (ch^i_1, ch^i_2, ch^i_3, ..., ch^i_m)$, where Ch denote the channel in the public dataset i, and m is the size of channel in the pubic dataset i.

After preparing the public dataset, we define D_s as a source dataset and D_t as a target dataset. D_s and D_t can be the same public dataset, or different dataset depend on the experiment setting. Consequently, we train the model commonly with the D_s.

Commonly train is mean training the 80% of public dataset and evaluates with 20% of the public dataset with a specific channel. The model, trained from the D_s, is called source model (m_s). For the D_t, we use 20% of D_t. The models, which come from D_t, contain two models. First, letting the model random weight to train D_t or we call this as target model without transferability (m_t). Secondly, we load the m_s to be the starting point for training with the D_t. This second model is called target model with transferability (m_{tf}).

For the evaluation, we decide to use the m_t and m_{tf} to evaluate the performance of transferability. The performance gain is the formula, which use to evaluate m_t and m_{tf}. Consequently, we use the performance gain 3.2.4.1 to find the factor that affects the transferability and answer three questions.

There are several conditions for transferability, so we propose to scope the condition only the channel relies on EEG. EEG is the prevalent channel, which useful for home-based sleep monitoring. EEG is a proper channel for the wearable device because it works well with the micro-nano system technology [24][25]. Therefore, transferability can minimize the time of labelling the sleep stage, which comes from the wearable device if the transferability relies on EEG effective. Consequently, we propose a transferability scenario for answering three questions below.

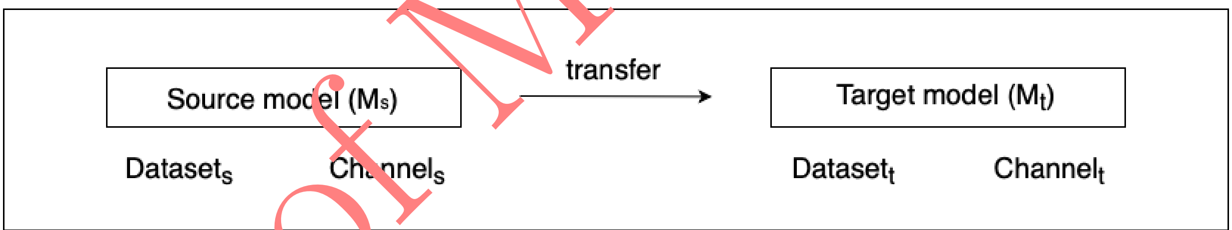


Figure 3.3: Transfer model from source model to target model with specific channel

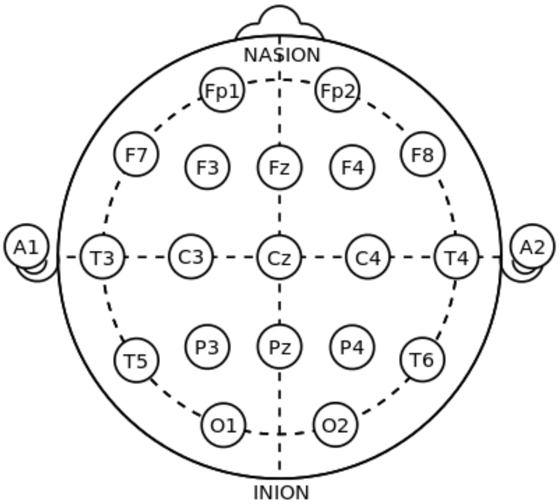


Figure 3.4: Electrode placement on the scalp [5]

3.2.3.1 Question1: Is transferring same the dataset with across the channel affect the transferability ?

Sleep signal datasets contain several channels depend on collecting protocol. It makes the transferability get the high chance to deal with the channel mismatch. Therefore, we propose to study the transferability from the source model to target model with the channel mismatch problem. The source model and the target model have to use the channel from different collection signal area to transferability. As figure 3.4, the sleep signal collected from many positions. We can group the collecting position into four groups (Frontal pole Fp, Central C, Parietal P, Occipital O). After the experiment of channel mismatch setting, it helps to understand the dissimilarity degree between two channels compared by performance gain. To avoid the effect of data collecting protocol, we have to transfer across the channel in the same dataset.

In this question, we try to find the answer: If data is transferred in the same dataset, but with a different channel (channel mismatch), it affects the transferability. In each experiment, we set source dataset (D_s) and target dataset (D_t) to be the same and set source channel (Ch_s) and target channel (Ch_t) to be difference depends on channel contained in each dataset (As figure 3.3). It makes we can study the factor from the channel mismatch. The experiment result shown in table 4.3

$$D_s = D_t \tag{3.2}$$

$$Ch_s \neq Ch_t \tag{3.3}$$

3.2.3.2 Question2: Is transferring in across the dataset with the same channel affect the transferability ?

The transferability is not only deal with the channel mismatch but also deal with the data collecting protocol. Each of the sleep signal datasets uses the different collecting protocol. For example, MASS subset1 collect the sleep signal with 7680 sampling rate while SleepEDF collects the sleep signal with 3000 sampling rate. Therefore, we propose to study the effect of the different recording environment to the performance gain. To decrease the effect from another factor, we set up the source channel (Ch_s) and target channel (Ch_t) to be the same. If the source and target dataset do not have the same channel, we pick the nearest channel on the target dataset to be target channel (Ch_t). For instance, dataset A contains Fz channel while dataset B do not have Fz channel (As figure 3.3). Therefore, the transferability process picks $F4$, which is the nearest channel with Fz from dataset B.

In this question, we try to find the answer that if data is transferred across the dataset but in the same channel (dataset mismatch), how it affects the transferability. In each experiment, we set source dataset (D_s) different from target dataset (D_t) and set source channel (Ch_s) and target channel (Ch_t) to be the same depends on channel contained in each dataset. It makes we can study the factor from the dataset mismatch. The experiment result shown in table 4.4

$$D_s \neq D_t \tag{3.4}$$

$$Ch_s = Ch_t \tag{3.5}$$

3.2.3.3 Question3: Is transferring across the dataset with different channel affect the transferability ?

For the last question, we propose to increase the dissimilarity degree by combining the two questions above. The setting for an experiment in this question is transferred across the dataset with the channel mismatch. We prefer to study the effect of this setting to performance gain when increasing the dissimilarity degree. This configuration is the highest dissimilarity degree because of transferability with different collecting protocol and channel mismatch.

In this question, we try to find the answer that if data is transferred across the dataset with channel mismatch, how it affects the transferability. In each experiment, we set source dataset (D_s) different from target dataset (D_t) and source channel (Ch_s), and target channel (Ch_t) to be the difference depends on channel contained in each dataset (As figure 3.3). It makes we can understand the effect of accuracy gain and f1 gain when the dissimilarity increase (This experiment is the highest dissimilarity degree). The experiment result shown in table 4.5

$$D_s \neq D_t \tag{3.6}$$

$$Ch_s \neq Ch_t \tag{3.7}$$

3.2.4 Dissimilarity Measure

We have to include two more formulas for evaluating the transferability. The formula for evaluating the transferability is accuracy gain and macro f1 gain. Every trained model are evaluated follow the general statistic formula (*i.e.* $Accuracy$, $F1$, $Precision$, $Recall$, $Macro\ F1$). Consequently, accuracy gain and macro f1 gain are calculated from the difference in each model's general statistic. We use this section to explain the accuracy gain and macro f1 gain. We prefer to use the accuracy gain and macro f1 gain to evaluate the transferability (Performance gain).

3.2.4.1 Accuracy Gain

Accuracy gain is the differentiate of the accuracy of m_tf and m_t The accuracy gain is the value used for evaluating the ability of the transferability. On the other hand, the accuracy gain is used to quantify the dissimilarity between D_s with Ch_i^s and D_t with Ch_j^t. We use the accuracy of m_{tf} and m_t for finding the accuracy gain. The formula of the accuracy gain is provided in formula 3.8. To analyze the accuracy gain, we assume two way of the values follows below.

1. When the accuracy gain is high, it means D_s with Ch_i^s and D_t with Ch_j^t are similar.

2. When the accuracy gain is high, it means D_s with Ch_i^s and D_t with Ch_j^t are dis-similar.

$$Accuracy\,gain = accuracy\,of\,m_{tf} - accuracy\,of\,m_t \tag{3.8}$$

3.2.4.2 Macro F1 Gain

Macro F1 gain is the differentiate of the macro f1 of m_tf and m_t Macro F1 gain is the value that helps us to evaluate the transferability performance. We decide to use the macro f1 gain with the performance gain because of training data are imbalance. In every sleep datasets, N1 class or class, which the subject starts fall in sleep, contains least because it is the stage in which the brain starts to pause the activity, and the brain signal frequency is slow-wave. Moreover, the wake-up class contains too much because the doctor does not know when the subject falls asleep. It makes the doctors have to collect before and after the subject sleeps too long. Therefore, macro f1 help us to scope into the performance after knowing the accuracy gain. Macro f1 helps us to weigh the class, which contains a huge number of datasets down and up the class, containing a small number of datasets. To analyze the macro f1, we assume two groups as the same as the accuracy gain. we use macro f1 of m_{tf} and m_t for finding the macro f1 gain. The macro f1 gain formula is provided in 3.9.

$$macro\,f1\,gain = macro\,f1\,of\,m_{tf} - macro\,f1\,of\,m_t \tag{3.9}$$

CHAPTER 4
TESTING AND EVALUATION

In response to the three questions of this research, we use this section to clarify information regarding the used datasets and channels of each dataset. Consequently, we interpret specific experiments set up. The specific experiments contain several trained models. Hence, this section illustrate the difference between the target model (m_t) and transfer model (m_{tf}). Both models are trained with a network parameter, which set depends on the configuration of the channel and dataset. In the last of this section, we show the experimental results with the performance metric used to evaluate each step of the training model and each step of transfer.

4.1 Dataset Information

Introduction of dataset information is the first essential for the experiment. We have to study the information about each dataset. Therefore, we use this section to explain the characteristic of each dataset, including the manual that sleeps expert use to label the sleep signal. Moreover, we provide the table for showing the channel of the sleep signal in each dataset. Dataset information is the first step of the experiment setup for finding the three questions answer.

Table 4.1: Characteristic of public dataset [1][2]

Dataset	Epoch Duration	Sex		Age	Sleep Stage Rule	Sampling frequency	Sleep Stage					
		male	female				W	N1	N2	N3	REM	total
SleepEDF	30	10	10	18-76	R&K standard	100 Hz	7927	2804	17799	5703	7717	41950
MASS_SS1	30	34	19	55-76	AASM standard	256 Hz	12241	7112	22166	3407	6365	51291
MASS_SS2	20	8	11	18-33	R&K standard	256 Hz	2827	1483	13090	4401	4910	26711
MASS_SS3	30	28	34	20-69	AASM standard	256 Hz	6442	4839	29802	7653	10581	59317
MASS_SS4	20	14	26	18-35	R&K standard	256 Hz	6701	4021	25807	8188	10593	55940
MASS_SS5	20	13	13	20-59	R&K standard	256 Hz	2972	1904	17064	6734	7735	36409

4.1.1 SleepEDF

SleepEDF dataset [1] consists of 20 subjects from Caucasian males and females with age between 21 to 35 years old, and all of them are healthy subjects. Polysomnography (PSG) [26] record several things about sleep. SleepEDF dataset records the whole night with sampling frequency rate equal to 100 Hz. SleepEDF measure the sleep quality from several parts of the body, such as brain signal (EEG), and eyes movement (EOG). Sleep expert label the sleep signal from SleepEDF every 30 seconds follow the R&K manual. We separate the label of SleepEDF to be seven stages (W, N1, N2, N3, N4, REM, and M) as table 4.1.

4.1.2 MASS

MASS dataset [2] consists of five subsets from a different country; hence each subset has a different configuration of collecting the polysomnography (PSG). MASS subsets can group into two groups, which provided below.

1. 30 seconds per epoch is the first group. Subset 1 and 3 use 30 seconds per epoch configuration and using AASM standard to label the sleep signal.

2. 20 seconds per epoch is the first group. Subset 2, 4, and 5 use 20 seconds per epoch configuration and using R&K standard to label the sleep signal.

Both groups of MASS dataset use sampling frequency rate equal to 256 Hz to collect the sleep signal. Moreover, this dataset consists of healthy and unhealthy subjects. MASS collect many types, such as brain signal (EEG), and eyes movement (EOG). However, we consider only EEG parts, which is the best channel to use in a wearable device. The sleep stages consist of W, N1, N2, N3, and REM with 200 subjects as table 4.1.

4.1.3 EEG Channels

Two datasets above contain many EEG channel. To make it easier to imagine, we provide the table 4.2. In this table 4.2 show the popular channel, which most of the sleep dataset always collect. Moreover, the channels provided in the table are considered to be transfer.

Table 4.2: EEG Channel in each Dataset

Dataset\Channel	Fpz_Cz	Fpz	Fz	F3	F4	Cz	C3	C4	Pz_Oz	Pz	P3	P4	O1	O2
SleepEDF	✓								✓					
MASS_SS1			✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓
MASS _SS2		✓		✓	✓	✓	✓	✓		✓	✓	✓	✓	✓
MASS_SS3			✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓
MASS_SS4							✓	✓					✓	✓
MASS_SS5			✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓

4.2 Network Parameter

The network parameter is the configuration used to train the model. All of the trained models are using the same network parameter. Moreover, it helps the training process to reduce over-fitting and make it more reliable. For the framework, we use the *TensorFlow* for set up, run and monitoring the training process. Hence, we use this section to clarify each parameter fed to train the model.

4.2.1 Adam Optimization

Adam optimization is the algorithm, which combines the advantage of two classical stochastic gradient descent [27]. Adam optimization contains several parameters for update the network weights iterative in the training process. In our model architecture figure 3.1, it uses four parameters of adam optimization.

1. Adam Beta1: The exponential decay rate, which uses for the first-moment estimation (*adam beta1 = 0.9*).
2. Adam Beta2: The exponential decay rate, which uses for the second-moment estimation (*adam beta2 = 0.999*).
3. Adam Epsilon: It is the prevention of any division with zero (*adam epsilon = 10^{-8}*).
4. Gradient norm: It is the value for initial the norm of the gradient (*clip_grad_value = 5.0*)

4.2.2 Learning rate

The learning rate is the hyperparameter that controls the weight of the neural network. The learning rate which we set is 1e-4. The learning rate cannot be too high because the model has a high chance of missing the local minima. In contrast, the model takes a long time to find the best model when the learning rate is too low. Due to this, we use 1e-4, which is not too small to find the best weight that is suitable for the signal data.

4.2.3 Early Stopping

Early stopping is the value that makes the training process done early. It occurs when the model does not provide better than checkpoint for a while. The early stopping used with the source model training (m_s). The target model (m_t and $m_t f$) is not used because it trains with insufficient data. Hence, it makes the model has to learn longer than m_s. The early stopping of the model is set to be *50 epoch*. If the model always achieves better, the model ends at the epoch duration equal to 200. The best model is the model, which achieves the best performance between epoch 1 to epoch 200.

4.2.4 Input Size

The input size is the size of the data, which used to feed to train or test the model. The input size is calculated from the sampling rate and the epoch duration. Therefore, the input size comes from three option below.

1. Sampling rate 100 with 30 epoch duration, the input size equals 3000
2. Sampling rate 256 with 30 epoch duration, the input size equals 7680
3. Sampling rate 256 with 20 epoch duration, the input size equals 5120

4.2.5 Number of class

The label data from the doctor come with many class of sleep signal file. However, the preparing process of our research decides to use five class of doctor labelled for training and testing the model. We set the number of class, which feed to the model to be 5.

4.2.6 Data Augmentation

Data augmentation is expanding the size of the training dataset. It used to generate new train signal from the original signals by applying random such as rotations, shearing and changes in scale. Data augmentation of the model is set to be true.

4.2.7 Sequence Length and Batch size

Batch size is the number of subject data which feed to the model each time. Sequence length is the size of the data, which the model used to train or test. For our research, the model received subjects equal to the batch size in each time. Consequently, the model learns with a sequence length of the batch size data at a time. If the resource of training is not a large scale, the batch size and sequence length are changed to be a small number.

4.3 Performance Metrics

Training process contains *200 epoch* for getting one model. To know which epoch suitable with the data, we have to use the statistic formula for evaluation. We use the statistic formula inside the field of classification. Therefore, we use this section to clarify the useful formula.

4.3.1 Evaluation formula

This research evaluates the performance of the model using per class precision (PR), per class recall (RE), per class F1-score, Cohen's Kappa coefficient [28][29], macro-averaging F1-score $(MF1)$, and model accuracy. Per class, the evaluation part considers in the single class. The macro-averaging F1-score $(MF1)$, Accuracy and other formula are calculated follow the formula below.

$$ACC = \sum_{c=1}^{C} \frac{TP_c}{N} \tag{4.1}$$

$$MF1 = \sum_{c=1}^{C} \frac{F1_c}{C} \tag{4.2}$$

$$Precision = \frac{TP_c}{TP_c + FP_c} \tag{4.3}$$

$$Recall = \frac{TP_c}{TP_c + FN_c} \tag{4.4}$$

$$F1 = 2 \cdot \frac{precision_c \cdot recall_c}{precision_c + recall_c} \tag{4.5}$$

Generally, TP_c is True positive of class c. FP_c is False positive. $F1_c$ is the $F1$ score of class C. Moreover, C is the number of sleep stage scoring. N is a number of test epoch. All fold of the training part is using the same formula to evaluate the performance.

4.3.2 Evaluate with validation

For the model, which trains from all data in the dataset without split more than one fold, is use the formula as same as evaluation 20-30 fold cross-validation (formula 4.1, 4.2, 4.3, 4.4, and 4.5). In contrast, this evaluation part does not have the test set in the training process, so it uses the validation set to be the set that used to evaluate. Therefore, the performance of the best model in the training process comes from the evaluation of the validation set.

4.4 Experiment Setup

The experiment setup is created for an empirical study of the transferability in sleep stage scoring. We also use the result of the experiment to quantify the factor that affects the transferability. Moreover, we prefer to answer the three questions in our scope of this research. The experiment configuration of each deep learning model is provided in tables 4.3, 4.4, and 4.5. In each table, it contains the configuration of Source dataset (D_s) and Target dataset (D_t) with the specific channel (Ch_s, and Ch_t). The table also includes accuracy gain and f1 gain, which each configuration provides.

4.4.1 Data splitting

Source and target models train with various data. To make it more reliable, we have to create the configuration pattern of trained data. The configuration pattern is splitting the data into train set and test set. Moreover, the source model and target model use the same network parameters. The network parameters are provided in the 4.2 section.

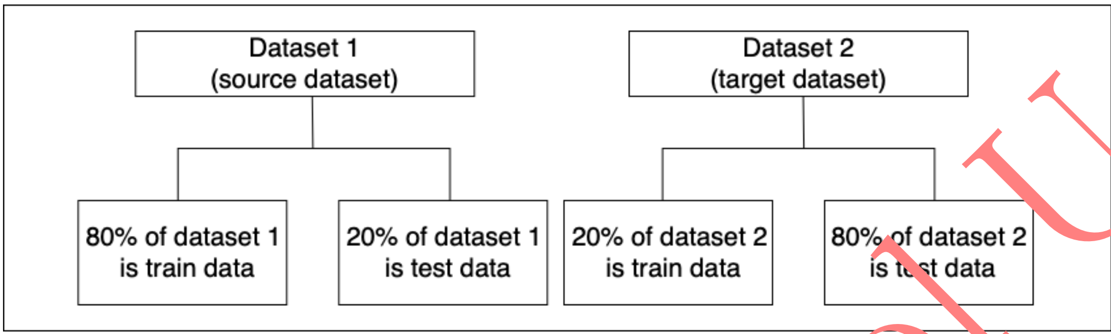


Figure 4.1: Train and Test splitting formula in our research

4.4.1.1 Train and Validate sets

Train and validate sets are the set used to train the model. The size of the train and validate sets are separate into two cases. The first case is the case of training source model (m_s). In this case, we separate 80% of the public dataset to be train set. For the validate set, it uses 10% of the data in the train set. The validation set is the set used to evaluate the model in the training process. The second case is the training target model (m_t and m_{tf}). For this case, it differences if compared with the first case. We use only 20% to be train data and 10% of train data for validation. It separates 20% of the dataset because this research proposes to simulate the insufficient training data. Therefore, 20% of the dataset is not large enough to train the deep learning model. To make it easier to understand, figure 4.1 show the pattern of splitting the dataset.

4.4.1.2 Test set

The test set is the set of the data, which the model has never learn and evaluate until finish the training process. Therefore, the test set of the source and target model are split differently. For the source model, the test set is split 20% from the source dataset (D_s). In contrast, the test set of the target model is split 80% of the dataset (D_t).

The reason, which split the test set from the target dataset to large, is to evaluate that is transferability make the model accurate enough? This research uses the evaluation of the test set to quantify the transferability performance and answering the objective of the research.

4.5 Result

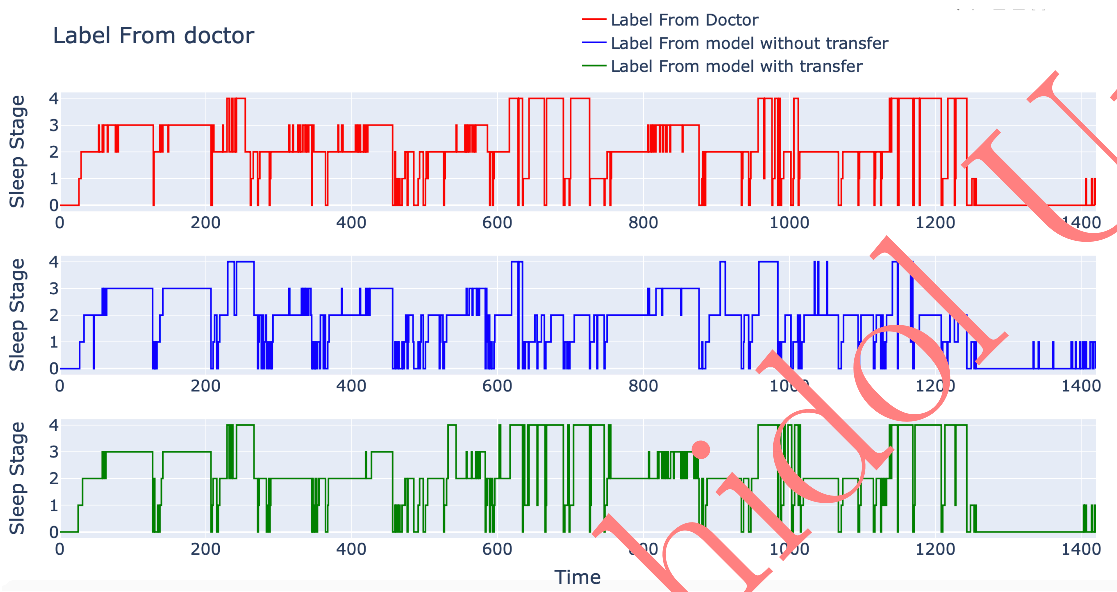


Figure 4.2: Hypnogram of the Deep Learning model without transferring and Deep Learning model with transferring compare with Hypnogram that label from the doctor

We use the predicting result of the target model without transferring and target model with transferring plot the hypnogram as figure 4.2 comparing with the label from the doctor. Following the experiment setup, we execute each configuration follow the table 4.3, 4.4, and 4.5. Three tables also contain the accuracy gain and f1 gain. After we are getting the result of each experiment, we visualize the data via boxplot. Hence, we use this section to clarify the data that we get.

Table 4.3: Accuracy gain and F1 gain of transferring in the same dataset with channel mismatch for answering question 1

	Source Dataset (D_s)	Source Channel (Ch_s)	Target Dataset (D_t)	Target Channel (Ch_t)	Performance Gain	
					Accuracy	Macro F1
1	SleepEDF	EEG Pz_Oz	SleepEDF	EEG Fpz_Cz	7.14	7.19
2	SleepEDF	EEG Fpz_Cz	SleepEDF	EEG Pz_Oz	3.25	2.75
3	SleepEDF	EEG Pz_Oz	SleepEDF	EEG Fpz_Cz	1.78	2.35
4	SleepEDF	EEG Fpz_Cz	SleepEDF	EEG Pz_Oz	4.09	4.99
5	SleepEDF	EEG Pz_Oz	SleepEDF	EEG Fpz_Cz	6.35	10.31
6	SleepEDF	EEG Fpz_Cz	SleepEDF	EEG Pz_Oz	5.51	7.06
7	MASS_SS1	EEG F4	MASS_SS1	EEG Pz	18.5	16.44
8	MASS_SS1	EEG C4	MASS_SS1	EEG Pz	21.48	18.12
9	MASS_SS1	EEG C4	MASS_SS1	EEG Pz	6.45	8.12
10	MASS_SS1	EEG Pz	MASS_SS1	EEG C4	9.18	6.03
11	MASS_SS1	EEG Pz	MASS_SS1	EEG C4	19.51	22.2
12	MASS_SS1	EEG Pz	MASS_SS1	EEG F4	6.25	7.97
13	MASS_SS2	EEG C4	MASS_SS2	EEG Fpz	4.3	6.36
14	MASS_SS2	EEG C4	MASS_SS2	EEG Pz	4.95	5.95
15	MASS_SS2	EEG F4	MASS_SS2	EEG Fpz	8.9	12.09
16	MASS_SS2	EEG F4	MASS_SS2	EEG Pz	3.68	5.46
17	MASS_SS2	EEG Fpz	MASS_SS2	EEG C4	4.18	6.63
18	MASS_SS2	EEG Fpz	MASS_SS2	EEG F4	6.87	9.7
19	MASS_SS2	EEG Pz	MASS_SS2	EEG C4	-1.81	-1.82
20	MASS_SS2	EEG Pz	MASS_SS2	EEG F4	0.66	1.04
21	MASS_SS3	EEG Fz	MASS_SS3	EEG Cz	16.24	23.16
22	MASS_SS3	EEG Fz	MASS_SS3	EEG Pz	6.32	8.44
23	MASS_SS3	EEG Fz	MASS_SS3	EEG O1	6.69	8.66
24	MASS_SS3	EEG Fz	MASS_SS3	EEG O2	3.6	3.45
25	MASS_SS3	EEG Cz	MASS_SS3	EEG Fz	4.33	5.58
26	MASS_SS3	EEG Pz	MASS_SS3	EEG Fz	3.69	5.01
27	MASS_SS3	EEG O1	MASS_SS3	EEG Fz	1.85	2.26
28	MASS_SS3	EEG O2	MASS_SS3	EEG Fz	2.1	2.68
29	MASS_SS3	EEG Pz	MASS_SS3	EEG O1	5.71	7.53
30	MASS_SS3	EEG Pz	MASS_SS3	EEG O2	3.83	4.85
31	MASS_SS3	EEG O1	MASS_SS3	EEG Pz	2.51	4.51
32	MASS_SS3	EEG O2	MASS_SS3	EEG Pz	5.59	7.9
33	MASS_SS4	EEG C4	MASS_SS4	EEG O1	12.85	11.86
34	MASS_SS4	EEG C4	MASS_SS4	EEG O2	6.88	6.17
35	MASS_SS4	EEG O2	MASS_SS4	EEG C4	30.69	15.02
36	MASS_SS4	EEG C3	MASS_SS4	EEG O1	24.01	15.68
37	MASS_SS4	EEG O1	MASS_SS4	EEG C3	30.84	17.4
38	MASS_SS4	EEG O1	MASS_SS4	EEG C4	30.5	17.78
39	MASS_SS5	EEG C4	MASS_SS5	EEG Pz	9.05	12.99
40	MASS_SS5	EEG Pz	MASS_SS5	EEG C4	5.85	8.11
41	MASS_SS5	EEG Pz	MASS_SS5	EEG C4	6.6	9.55
42	MASS_SS5	EEG C4	MASS_SS5	EEG Pz	3.13	4.5
43	MASS_SS5	EEG Pz	MASS_SS5	EEG C3	5.19	7.78
44	MASS_SS5	EEG C3	MASS_SS5	EEG Pz	2.54	4.14
45	MASS_SS5	EEG Pz	MASS_SS5	EEG O1	17.61	27.9
46	MASS_SS5	EEG O1	MASS_SS5	EEG Pz	10.95	12.47

Table 4.4: Accuracy gain and F1 gain of transferring across the dataset with the same channel for answering question 2

	Source Dataset (D_s)	Source Channel (Ch_s)	Target Dataset (D_t)	Target Channel (Ch_t)	Performance Gain	
					Accuracy	Macro F1
1	SleepEDF	EEG Pz_Oz	MASS_SS1	EEG Pz	11.06	-0.39
2	SleepEDF	EEG Pz_Oz	MASS_SS3	EEG Pz	0.83	0.53
3	SleepEDF	EEG Fpz_Cz	MASS_SS1	EEG Fz	1.81	3.69
4	SleepEDF	EEG Fpz_Cz	MASS_SS3	EEG Fz	2.03	2.4
5	MASS_SS1	EEG C4	MASS_SS3	EEG C4	3.38	3.45
6	MASS_SS1	EEG Pz	SleepEDF	EEG Pz_Oz	3.93	6.57
7	MASS_SS1	EEG F4	MASS_SS3	EEG F4	2.7	2.92
8	MASS_SS1	EEG Fz	MASS_SS3	EEG Fz	2.36	2.91
9	MASS_SS1	EEG Cz	MASS_SS3	EEG Cz	3.43	4.09
10	MASS_SS1	EEG Pz	MASS_SS3	EEG Pz	6.62	[illegible]
11	MASS_SS1	EEG O1	MASS_SS3	EEG O1	2.09	5.58
12	MASS_SS1	EEG O2	MASS_SS3	EEG O2	4.23	4.14
13	MASS_SS1	EEG Fz	SleepEDF	EEG Fpz_Cz	[illegible]	6.64
14	MASS_SS3	EEG C4	MASS_SS1	EEG C4	[illegible]	1.89
15	MASS_SS3	EEG F4	MASS_SS1	EEG F4	3.42	2.98
16	MASS_SS3	EEG Fz	MASS_SS1	EEG Fz	[illegible]	11.06
17	MASS_SS3	EEG Cz	MASS_SS1	EEG Cz	4.26	4.87
18	MASS_SS3	EEG Pz	MASS_SS1	EEG Pz	11.67	14.15
19	MASS_SS3	EEG O1	MASS_SS1	EEG O1	8.3	11.36
20	MASS_SS3	EEG O2	MASS_SS1	EEG O2	7.19	8.77
21	MASS_SS3	EEG Pz	SleepEDF	EEG Pz_Oz	6.82	8.93
22	MASS_SS3	EEG Fz	SleepEDF	EEG Fpz_Cz	5.18	6.05
23	MASS_SS2	EEG C3	MASS_SS4	EEG C3	11.07	0.8
24	MASS_SS2	EEG C4	MASS_SS4	EEG C4	10.76	0.85
25	MASS_SS2	EEG O1	MASS_SS4	EEG O1	13.67	10.6
26	MASS_SS2	EEG F4	MASS_SS5	EEG F4	1.44	2.53
27	MASS_SS2	EEG Cz	MASS_SS5	EEG Cz	1.15	2.19
28	MASS_SS2	EEG Pz	MASS_SS5	EEG Pz	1.87	2.63
29	MASS_SS4	EEG C4	MASS_SS5	EEG C4	4.86	7.76
30	MASS_SS4	EEG C3	MASS_SS2	EEG C3	3.3	4.98
31	MASS_SS4	EEG C4	MASS_SS2	EEG C4	3.32	5.32
32	MASS_SS4	EEG O1	MASS_SS2	EEG O1	5.59	5.82
33	MASS_SS4	EEG C3	MASS_SS5	EEG C3	2.29	3.26
34	MASS_SS4	EEG C4	MASS_SS5	EEG C4	1.58	1.71
35	MASS_SS4	EEG O1	MASS_SS5	EEG O1	7.65	5.74
36	MASS_SS5	EEG F4	MASS_SS2	EEG F4	4.64	5.86
37	MASS_SS5	EEG C4	MASS_SS4	EEG C4	11.29	11.94
38	MASS_SS5	EEG Cz	MASS_SS2	EEG Cz	3.5	5.87
39	MASS_SS5	EEG Pz	MASS_SS2	EEG Pz	3.71	4.53
40	MASS_SS5	EEG O1	MASS_SS4	EEG O1	18.51	13.21
41	MASS_SS5	EEG C3	MASS_SS4	EEG C3	14.39	2.82
42	MASS_SS5	EEG C4	MASS_SS4	EEG C4	15.31	7.23
43	MASS_SS1	EEG Fz	SleepEDF	EEG Fpz_Cz	8.39	9.00
44	MASS_SS1	EEG Pz	SleepEDF	EEG Pz_Oz	9.86	9.97
45	SleepEDF	EEG Fpz_Cz	MASS_SS1	EEG Fz	6.64	7.81
46	SleepEDF	EEG Pz_Oz	MASS_SS1	EEG Pz	1.75	3.82
47	MASS_SS3	EEG Fz	SleepEDF	EEG Fpz_Cz	5.07	5.82
48	MASS_SS3	EEG Pz	SleepEDF	EEG Pz_Oz	9.52	8.52
49	SleepEDF	EEG Fpz_Cz	MASS_SS3	EEG Fz	0.55	0.33
50	SleepEDF	EEG Pz_Oz	MASS_SS3	EEG Pz	6.51	6.79

Table 4.5: Accuracy gain and F1 gain of transferring across the dataset with channel mismatch for answering question 3

	Source Dataset (D_s)	Source Channel (Ch_s)	Target Dataset (D_t)	Target Channel (Ch_t)	Performance Gain	
					Accuracy	Macro F1
1	SleepEDF	EEG Pz_Oz	MASS_SS1	EEG Fz	6.72	6.85
2	SleepEDF	EEG Fpz_Cz	MASS_SS1	EEG Cz	6.04	8.54
3	SleepEDF	EEG Fpz_Cz	MASS_SS1	EEG Pz	-3.25	-0.98
4	SleepEDF	EEG Fpz_Cz	MASS_SS3	EEG Fz	1.1	2.58
5	SleepEDF	EEG Pz_Oz	MASS_SS3	EEG Pz	0.03	-1.42
6	SleepEDF	EEG Fpz_Cz	MASS_SS3	EEG O1	5.96	6.06
7	SleepEDF	EEG Pz_Oz	MASS_SS3	EEG C4	-0.25	-1.08
8	SleepEDF	EEG Fpz_Cz	MASS_SS3	EEG C4	1.96	1.96
9	MASS_SS1	EEG Fz	SleepEDF	EEG Pz_Oz	6.22	5.98
10	MASS_SS1	EEG Cz	SleepEDF	EEG Fpz_Cz	4.24	4.73
11	MASS_SS1	EEG Pz	SleepEDF	EEG Fpz_Cz	2.93	4.26
12	MASS_SS1	EEG O1	SleepEDF	EEG Fpz_Cz	2.16	1.66
13	MASS_SS1	EEG Cz	MASS_SS3	EEG O1	3.97	5.30
14	MASS_SS1	EEG Pz	MASS_SS3	EEG O1	5.35	6.83
15	MASS_SS1	EEG O1	MASS_SS3	EEG Cz	-0.38	[illegible].17
16	MASS_SS1	EEG O1	MASS_SS3	EEG Pz	2.9	3.74
17	MASS_SS3	EEG Fz	SleepEDF	EEG Pz_Oz	6.91	5.01
18	MASS_SS3	EEG Cz	SleepEDF	EEG Fpz_Cz	[illegible].67	4.72
19	MASS_SS3	EEG Pz	SleepEDF	EEG Fpz_Cz	4.4[illegible]	4.75
20	MASS_SS3	EEG O1	SleepEDF	EEG Fpz_Cz	[illegible].78	4.35
21	MASS_SS3	EEG Cz	MASS_SS1	EEG O1	18.77	28.25
22	MASS_SS3	EEG Pz	MASS_SS1	EEG O[illegible]	19.81	27.66
23	MASS_SS3	EEG O1	MASS_SS1	EEG Cz	34.28	1.01
24	MASS_SS3	EEG O1	MASS_SS1	EEG Pz	35.56	15.67
25	MASS_SS2	EEG Fpz	MASS_SS4	EEG C4	0.11	1.68
26	MASS_SS2	EEG Fpz	MASS_SS4	EEG O1	2.5	2.05
27	MASS_SS2	EEG C4	MASS_SS[illegible]	EEG O1	2.57	2.82
28	MASS_SS2	EEG Pz	MASS_SS4	EEG C4	-5.86	-5.31
29	MASS_SS2	EEG O1	MASS_SS4	EEG C4	3.98	5.64
30	MASS_SS2	EEG Fpz	MASS_SS5	EEG C4	-0.91	-1.72
31	MASS_SS2	EEG Fpz	MASS_SS5	EEG Pz	0.45	1.22
32	MASS_SS2	EEG Fpz	MASS_SS5	EEG O1	-0.98	-2.92
33	MASS_SS2	EEG C4	MASS_SS5	EEG Fz	1.5	1.18
34	MASS_SS2	EEG Pz	MASS_SS5	EEG Fz	2.38	1.81
35	MASS_SS2	EEG O1	MASS_SS5	EEG Fz	2.31	3.85
36	MASS_SS4	EEG C4	MASS_SS2	EEG O1	6.01	6.75
37	MASS_SS4	EEG O1	MASS_SS2	EEG Fpz	6.2	7.52
38	MASS_SS4	EEG O1	MASS_SS2	EEG C4	0.87	0.90
39	MASS_SS4	EEG O1	MASS_SS2	EEG Pz	7.04	5.89
40	MASS_SS4	EEG C4	MASS_SS5	EEG O1	6.15	6.00
41	MASS_SS4	EEG O[illegible]	MASS_SS5	EEG Fz	1.78	4.02
42	MASS_SS4	EEG O1	MASS_SS5	EEG C4	0.56	-0.38
43	MASS_SS4	EEG O1	MASS_SS5	EEG Pz	3.37	4.81
44	MASS_SS5	EEG F4	MASS_SS2	EEG C4	2.08	2.06
45	MASS_SS5	EEG F4	MASS_SS2	EEG Pz	4.93	6.41
46	MASS_SS5	EEG F4	MASS_SS2	EEG O1	2.67	3.33
47	MASS_SS5	EEG Fz	MASS_SS2	EEG Pz	5.65	6.88
48	MASS_SS5	EEG Fz	MASS_SS2	EEG O1	2.52	3.01
49	MASS_SS5	EEG C4	MASS_SS2	EEG F4	0.08	0.55
50	MASS_SS5	EEG C4	MASS_SS2	EEG Pz	5.49	6.74
51	MASS_SS5	EEG C4	MASS_SS2	EEG O1	5.71	6.50
52	MASS_SS5	EEG Pz	MASS_SS2	EEG Fpz	1.68	3.62
53	MASS_SS5	EEG Pz	MASS_SS2	EEG F4	-1.04	-0.20
54	MASS_SS5	EEG Pz	MASS_SS2	EEG C4	1.07	1.74
55	MASS_SS5	EEG O1	MASS_SS2	EEG Fpz	1.67	2.59
56	MASS_SS5	EEG O1	MASS_SS2	EEG F4	-5.72	-6.39
57	MASS_SS5	EEG O1	MASS_SS2	EEG C4	1.93	2.01
58	MASS_SS5	EEG F4	MASS_SS4	EEG C4	7.41	9.69
59	MASS_SS5	EEG F4	MASS_SS4	EEG O1	0.11	1.21
60	MASS_SS5	EEG C4	MASS_SS4	EEG O1	4.17	4.61
61	MASS_SS5	EEG Pz	MASS_SS4	EEG C4	5.42	7.26
62	MASS_SS5	EEG O1	MASS_SS4	EEG C4	5.48	7.20

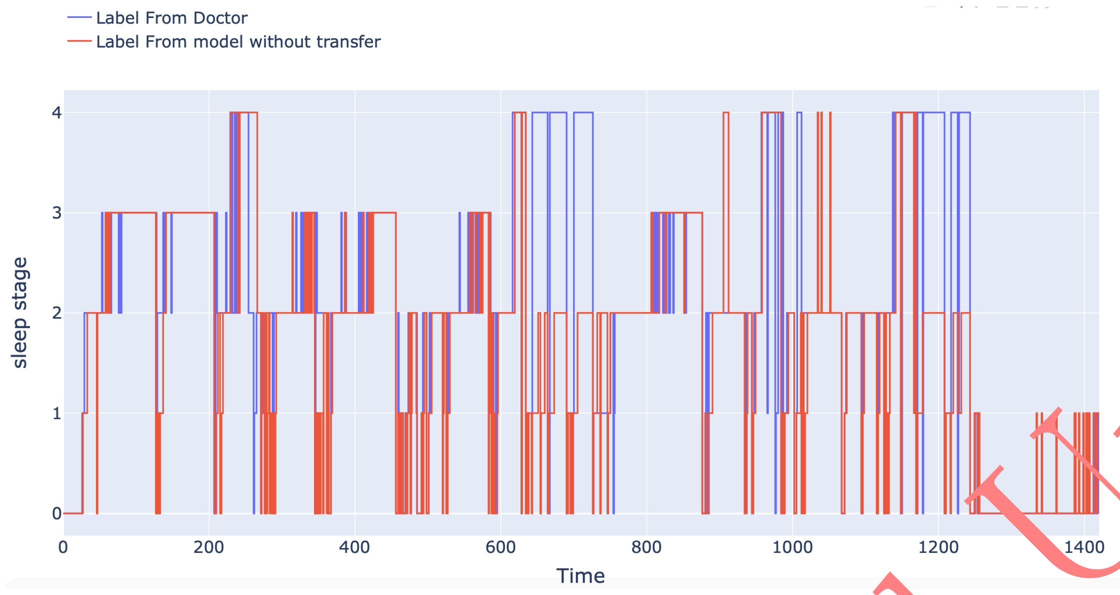


Figure 4.3: Hypnogram of the Deep Learning model without transferring compare with Hypnogram that label from the doctor

The model without transferring can not label the sleep signal correctly when the patient changes the sleep stage more frequently. Due to this problem, the model train with insufficient training data. The model does not perform useful enough because the size of the training data is too small. Therefore, the model has to learn only with a specific pattern of changing the sleep stage. As figure 4.3, the model always labels incorrect when the sleep signal changes the stage frequently. Hence, this problem solved when the model transferred from the useful source dataset.

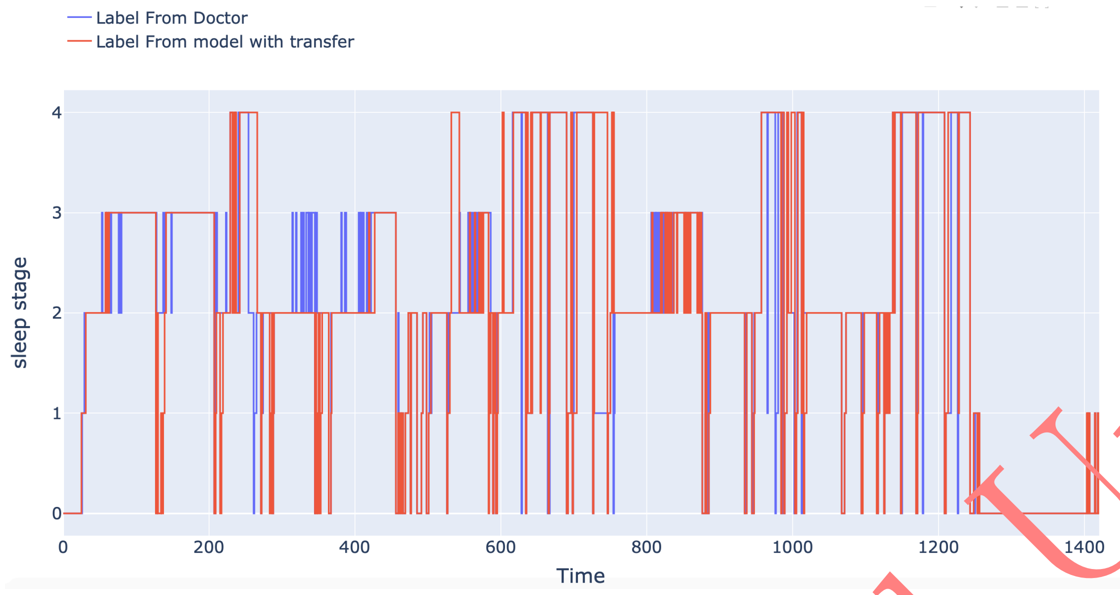
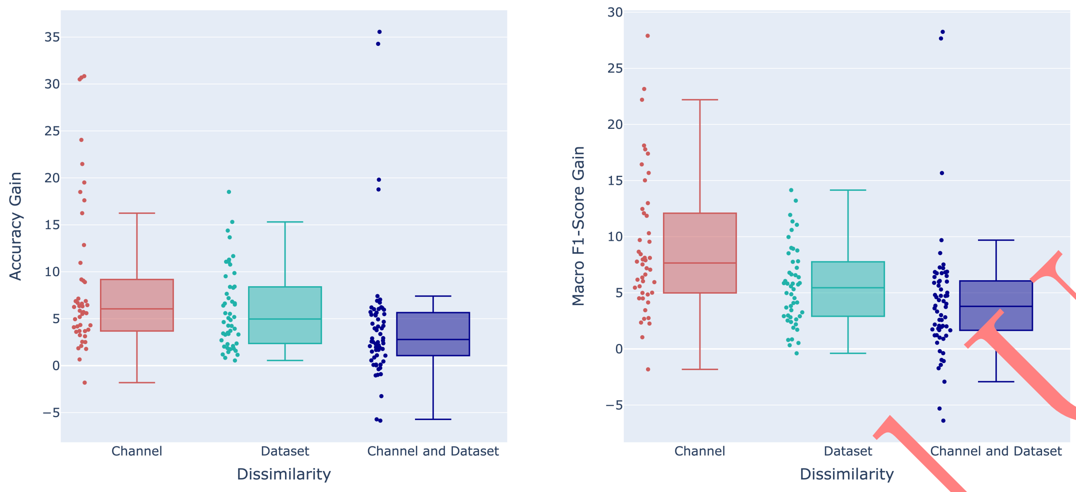


Figure 4.4: Hypnogram of the Deep Learning with transferability compare with Hypnogram that label from doctor

The transferring model labels the sleep stage more correctly when the patient always changes the sleep stages. The model with transferred has experience with sleep stage changing patterns. Therefore, when the patient often swaps the sleep pattern, the model can label more correctly. As figure 4.4, the model labels the stage that changing the sleep stage more correctly when the stage has high-frequency changing.

4.5.1 Overall Performance in Each experiment Question



(a) Accuracy Gain (b) F1 Gain

Figure 4.5: Box Plot of Three experiment setup

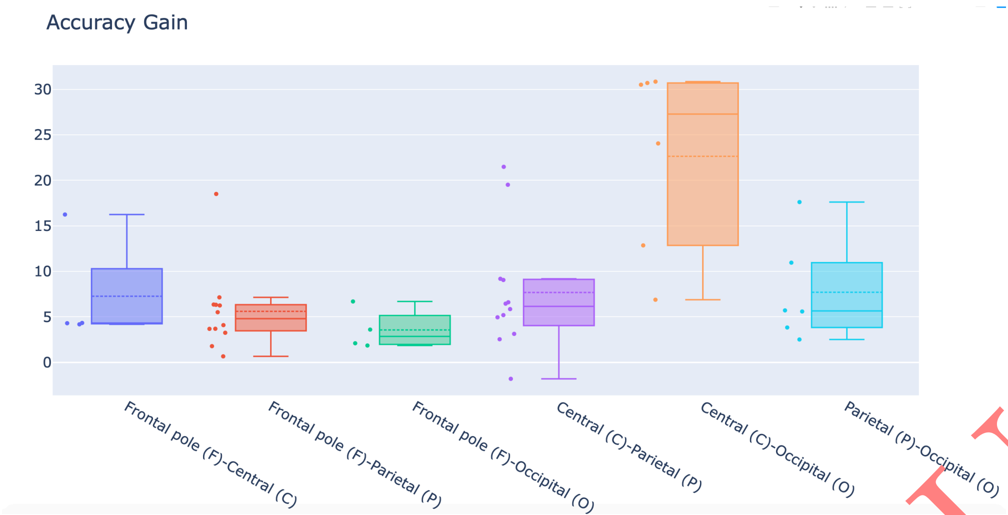
As figure 4.5a, 4.5b, it uses all of each experiment to create the box. The X-axis shows the average label box plot of each experimental setup while Y-axis shows the accuracy gain and f1 gain of transferability. For the red box, it represents the question 1 answer or channel mismatch. The red box provides the average accuracy and f1 gain better than another box (average accuracy gain = 8.70, and average f1 gain = 9.01). In this case, it is showing that channel mismatch is the least impact on the transferability factor.

For the green box, it is the experiment of dataset mismatch or question 2. It provides the average of the experiment to be the middle between the channel mismatch and transferability across the dataset with channel mismatch (average accuracy gain = 5.94, and average f1 gain = 5.35). After speculation of the result, the dataset mismatch is a higher dissimilarity degree than the channel mismatch. The dataset mismatch contains a high dissimilarity degree because the dataset contains several arguments. For example, SleepEDF collects the sleep signal with 100 Hz while the MASS collects the sleep signal with 256 Hz. Moreover, SleepEDF label 30 seconds per epoch, while MASS label 20 and 30 seconds per epoch depend on the subset (more information in table 4.1). This reason might effect to transferability that it has to learn with distinct knowledge more than channel mismatch.

For the dark blue box, This box represents the transferability across the dataset with the channel mismatch. This experiment is the highest dissimilarity degree because it contains the dataset mismatch and channel mismatch. Hence, it provides the least of average accuracy gain and f1 gain (average accuracy gain = 4.28, and average f1 gain = 4.25). As our hypothesis, this experiment should not provide accuracy gain and f1 gain better because it contains dataset and channel mismatch. It makes the deep learning model has to learn with too much new pattern of the sleep signal. We think the model has to learn with several new patterns, and knowledge from the source model does not help too much. Hence, this experiment provides the slightest accuracy gain and f1 gain.

4.5.2 Channel Mismatch

The channel mismatch provides the highest accuracy gain and f1 gain. The channel mismatch contains several directions of transferability. To understand better with the channel mismatch, we use the data from table 4.3 to visualize inside of channel mismatch. Hence, we use this section to describe the information that we get from visualizing the boxplot.



(a) Accuracy Gain

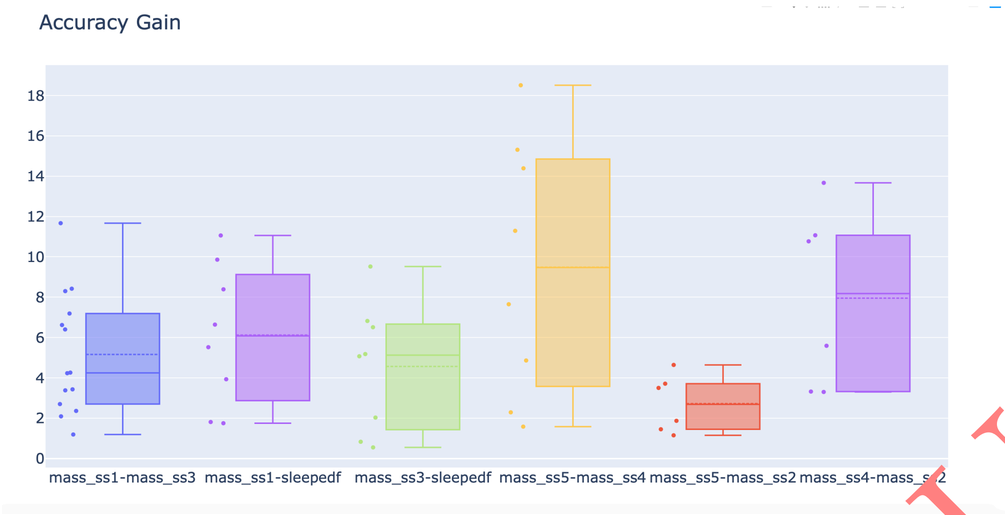
(b) F1 Gain

Figure 4.6: Box Plot of Channel Mismatch Grouping by Source Channel position to transfer channel position

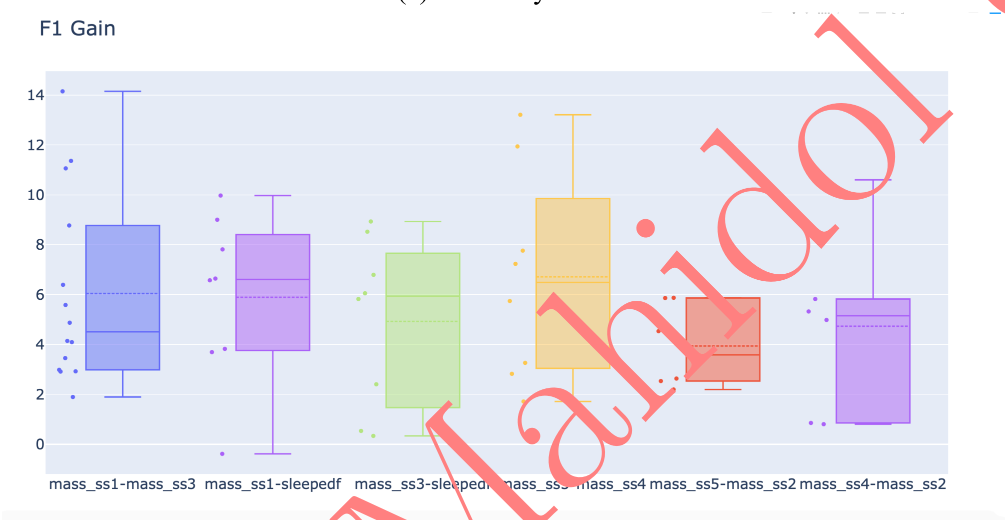
Transferring with the nearest channel provide performance better than the farther channel. As figure 4.6a and 4.6b, it shows the grouping of source-channel position and transfers position in the channel mismatch problem. The X-axis represents the group name. For example, frontal pole (F) - central (C) is mean transferability from the frontal channel to the central channel and transferability from the central channel to the frontal channel. The Y-axis defined as accuracy gain or f1 gain. From both graphs in this section, it shows that transferability between the nearest channel performs better than a farther channel. For instance, transferability from the frontal (F) to the occipital (O) compare with transferability from the central (C) to the occipital (O). Using central (C) to be the source channel provides better than use frontal (F) to be the source channel because the central (C) is close to occipital (figure 3.4 shows the distance between each position).

4.5.3 Dataset Mismatch

The dataset mismatch is various for transferring because each dataset uses different collecting protocol. Hence, the dissimilarity degree between the two datasets is significant to consider. We decide to use this section to show the detail of the performance between datasets via box plot. The box plot uses the data from the table 4.4 for visualizing.



(a) Accuracy Gain



(b) F1 Gain

Figure 4.7: Box Plot of Dataset Mismatch Grouping by Source Dataset to transfer dataset

Data collecting protocol affect the transferability directly. From figure 4.7a and 4.7b, it shows the group of transferability in dataset mismatch. The Y-axis represents the accuracy gain and f1 gain. The X-axis shows the group of datasets. For example, in X-axis, MASS subset 3 - MASS subset 1 is the combination of transferring MASS subset 3 to MASS subset 1 and MASS subset 1 to MASS subset 3. This visualization graph shows the inside of the dataset mismatch about the dissimilarity degree between the dataset. The accuracy gain and f1 gain in dataset mismatch vary directly with dataset argument, such as sampling rate. The transferability is effective when both datasets use similar collecting protocol.

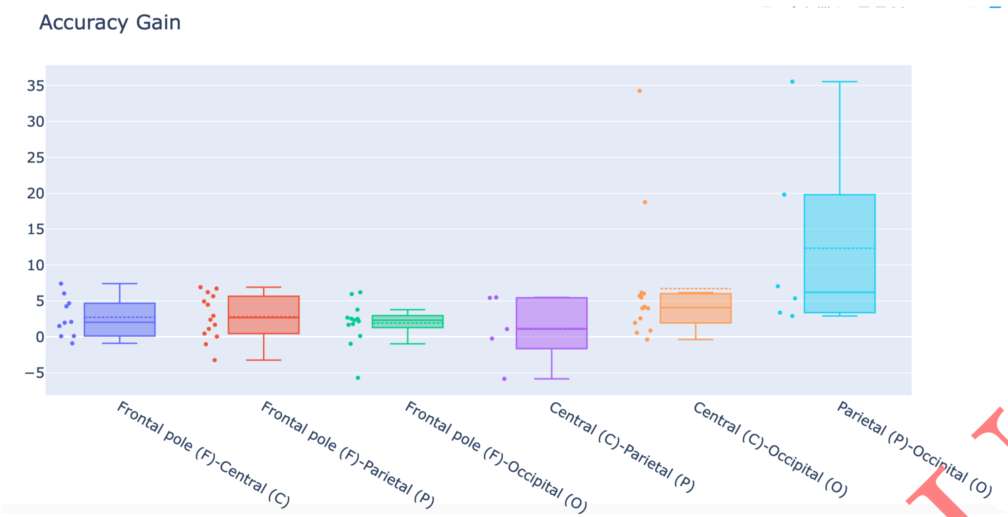
As figure 4.7a and 4.7b, transferability between MASS dataset and SleepEDF is not better than transferability between MASS subset 1 and MASS subset 3. This result is going on this way because every subset in the MASS dataset uses a similar collecting protocol while SleepEDF uses difference. For instance, MASS dataset collects the signal with 256 Hz while SleepEDF collects the signal with 100 Hz.

4.5.4 Dataset and Channel Mismatch

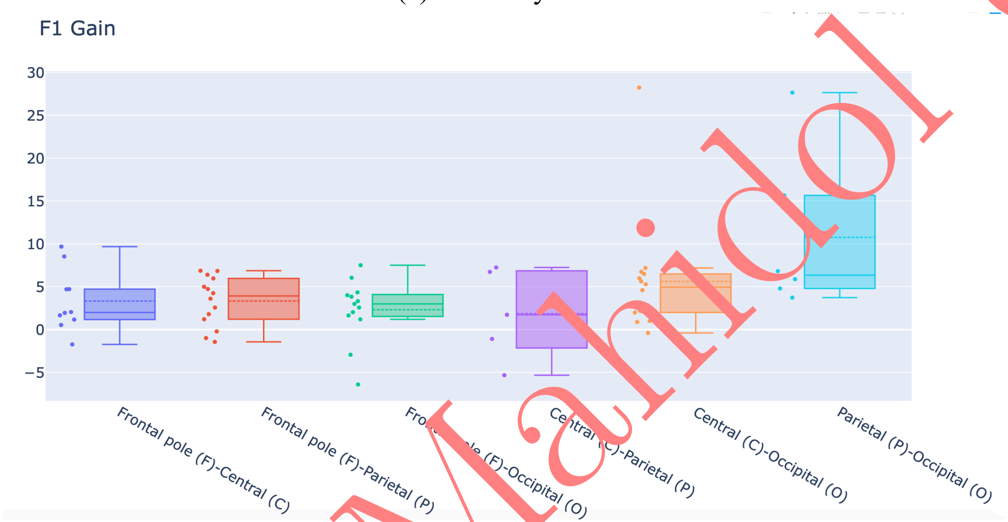
Transferring across the dataset with channel mismatch contains more configurations than another question configuration. This configuration is the highest dissimilarity degree because the source model (m_s) and the target model (m_t, and $m_t f$) are obviously different. It also makes the combination of the dataset and channel more than another experiment question. Hence, we use the information from the table 4.5 to visualize the insight of this configuration.

To analyze with question 3 result, it separates into two groups. First, we group the result with the channel mismatch as figure 4.8a, and 4.8b. Second, we group the result with the dataset mismatch as figure 4.9a, and 4.9b. The purpose of the two groups is to support question 1 and question 2 results. Moreover, we use this information to find the relation between the dataset and the channel.

Question 1 provides better performance than question 3 when comparing the configuration, which is almost the same on the channel mismatch problem. The channel mismatch is the configuration that questions 1 and question 3 contain. Hence, we group the result from the table 4.5 by channel. We can this result to find more insight than question 1. From table question 1, it is a transferring from channel $EEG\ Fpz$ to channel $EEG\ C4$ in the MASS subset2 (line 17). We use this line to be the example for comparing with line 25 in table question3. Line 25 is a transferring from MASS subset 2 to MASS subset 4 with channel mismatch as same as line 17 of table question1. These two lines show the accuracy gain and f1 gain that question 1 performance provides better than question 3 performance. The accuracy gain and f1 gain in question 3 decrease because of the dataset mismatch.



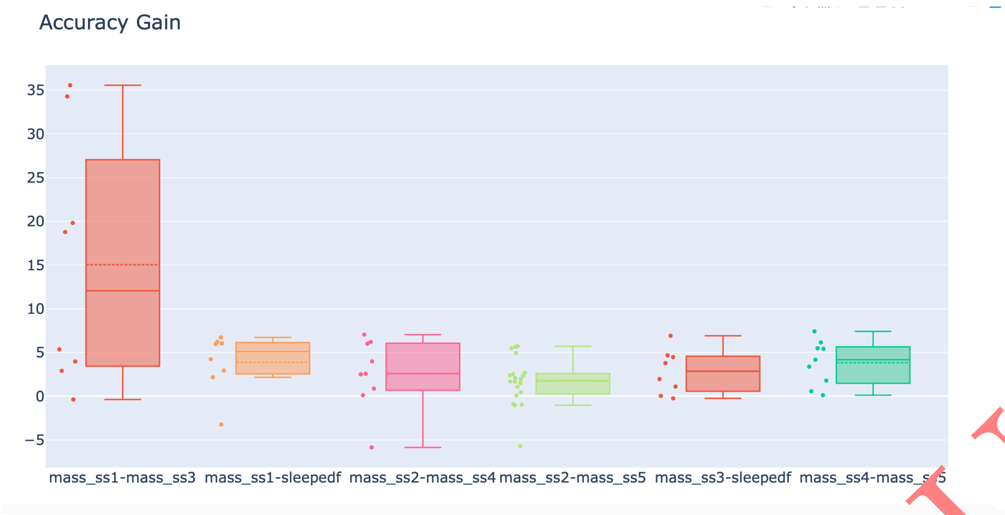
(a) Accuracy Gain



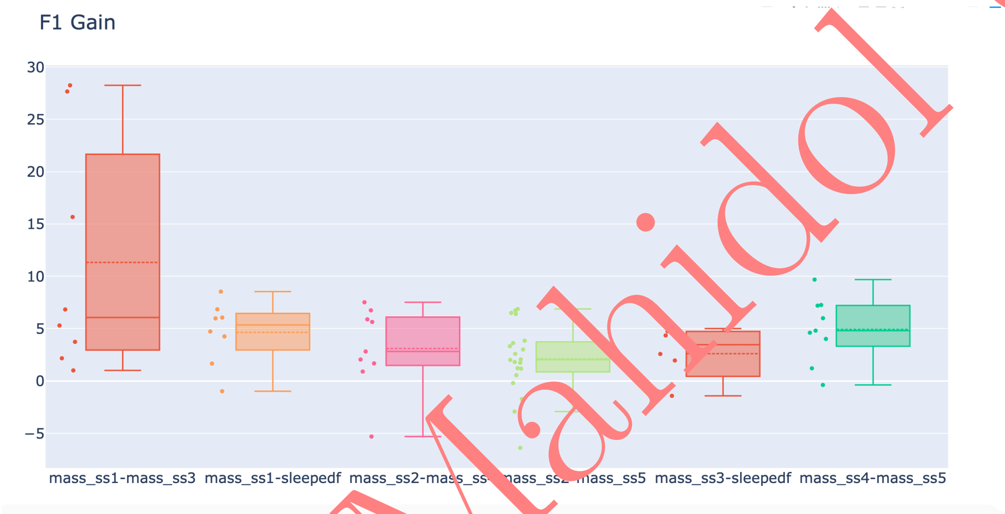
(b) F1 Gain

Figure 4.8: Box Plot of Transferability across dataset with channel mismatch Grouping by Channel

Transferring in questions 1 and 3 with the nearest channel still increases the performance of the transfer. As figure 4.8a and 4.8b, it shows the box plot which grouping the base position channel and transfer position channel. After the box plot generated, the results of figures 4.8a and 4.8b similar to the box plot of channel mismatch in question 1 (figure 4.6a, and 4.6b). The result of both configuration different only the central - occipital. For another channel grouping, it is going in the same way with the channel mismatch in question 1.



(a) Accuracy Gain



(b) F1 Gain

Figure 4.9: Box Plot of Transferability across dataset with channel mismatch. Grouping by dataset

After we study the transferability result between question 1 and question 3, we think the comparison between question 2 and question 3 is still challenging. Hence, we compare table 4.4 and table 4.5. Our understanding go on the same way with the difference between question 1 and question 3. It makes the result of question 2 is still better than question 3 because it contains only dataset mismatch, while question 3 contains dataset mismatch and channel mismatch.

To analyze into an insight of experimental in question 3, we visualize the result from the table via box plot as figures 4.9a and 4.9b. After the box plot processed, we see the pattern of the box plot from question 3 is going in a similar pattern with question 2 as figure 4.7a and 4.7b. If the accuracy gain and f1 gain from two questions are sorted, both question provides a similar order except transferability between MASS subset 1 and MASS subset 3.

CHAPTER 5
DISCUSSION

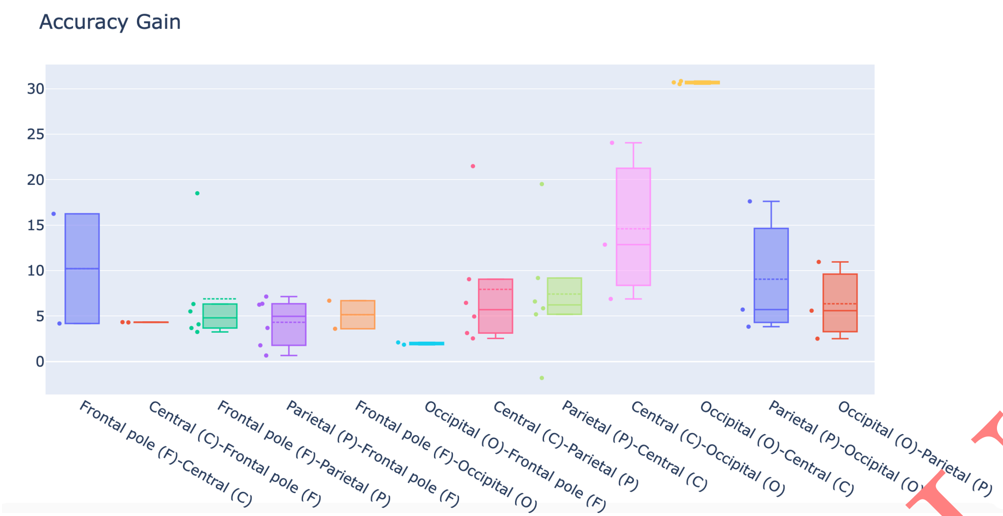
Following the experimental result analyzed, it is still challenging for us to understand the result in depth. The experimental results contain the unique characteristic of transferability. Hence, we propose to use this section to clarify the idea to make the transferability better. This section is designed to explain why the front over EEG channel should be the source channel in transferability. Moreover, we try to show a similar thing between each experimental result table. The concerning condition of the transferability and the challenges of the problem are included in this section.

5.1 Different Channels

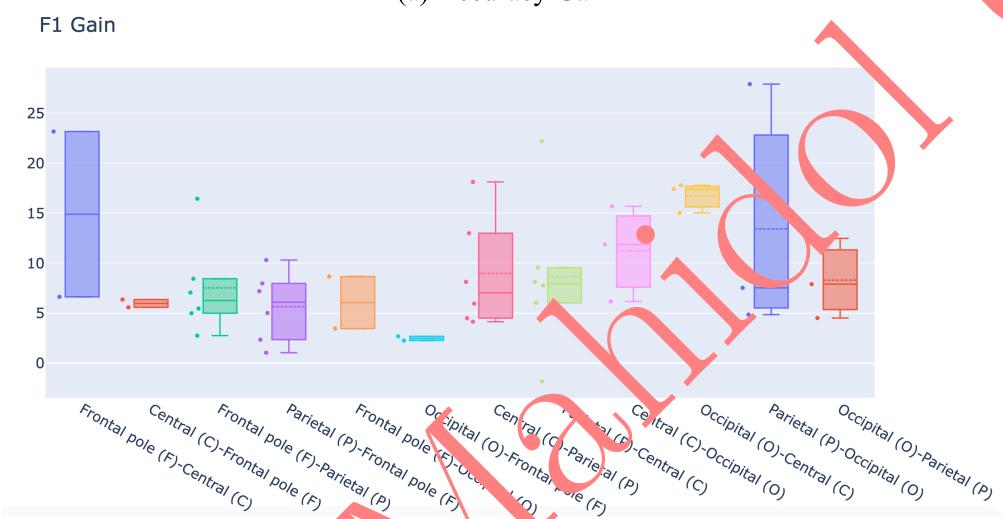
Transferability with the nearest channel minimizes the dissimilarity degree better than the farther channel. From figures 4.6a and 4.6b in chapter 4, it show that all of the channel provide better performance gain when it transfer with the nearest channel. One exception case is the occipital channel mismatch. When the transferability with occipital, it curious results. It occurs with MASS subset 4 (Line 35, 36, and 37 in table 4.3). On the other hand, the curious result in the occipital channel provides the macro f1 with an almost regular pattern. Hence, we study in-depth with the size of each class in MASS subset 4. We found that MASS subset 4 have the data in all class nearly balance. It seems helpful for the model to train with the balancing data, but MASS subset 4 has N3 and REM highest class if comparing with other datasets (size of each class provided in table 4.1). Therefore, the model does not get a high chance to learn the stage changing too much when we train the target model without transferring. It performs more accurately when the model reaches the transferability procedure.

Figure 5.1: Box Plot F1 Gain of Channel Mismatch Grouping by Type of Transferability

Transferring from the front over transfer channel to the transfer channel is increase the performance gain of the transferability. Using front over the transfer channel to be the source channel is better than transferring from the back over of the transfer channel. To make it more clear, using figure 3.4 to identify the direction of transferability. Frontal to back include F position to C, P, and O. C to P, O and P to O is also the group as $frontal$ to $back$. For the back to the frontal group is the reverse direction of the first group. From figure 5.1 and our experimental result, it supports the decision that transferability from the channel front over the transfer channel provides better performance than the reverse direction (frontal to back group provide f1 equal to 9.77 while the reverse direction provides 8.08).



(a) Accuracy Gain



(b) F1 Gain

Figure 5.2: Box Plot Channel Mismatch Matching direction forward and backward in each configuration

Figures 5.2a and 5.2b support the transferring with the nearest channel and the front to back direction is the best pattern to reduce the dissimilarity of the channel mismatch. We can compare the performance of the source channel and target channel via the boxplot. As the boxplot in figure 5.2a and 5.2b show, the accuracy and macro f1 gain still higher when the source channel is front over than the transfer channel. The channel almost follows the nearest channel and the front over direction pattern except for the occipital channel. The occipital channel follows only the nearest channel pattern. It occurs because the experiment of the occipital channel is experiment with the MASS subset4, which is not good on the data distribution.

5.2 Different Environments

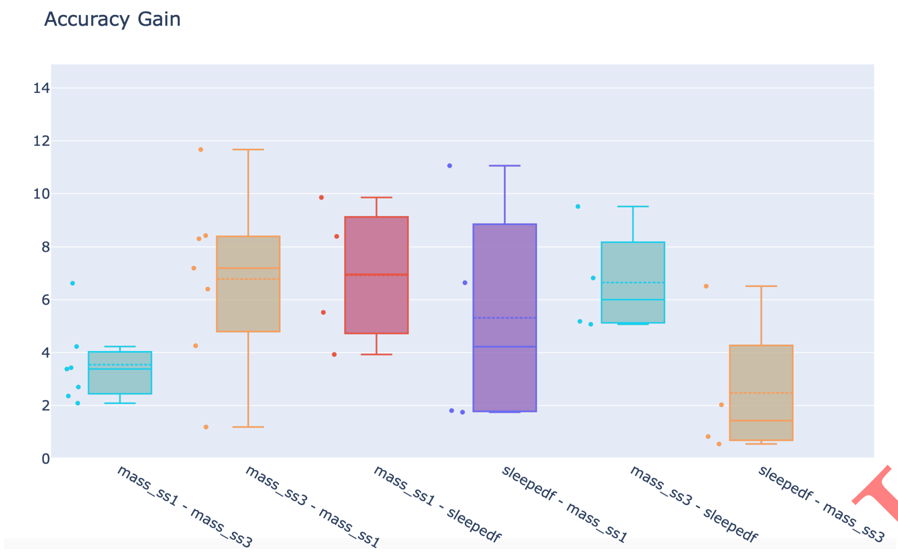
Different environments or different dataset is high dissimilarity than the channel mismatch because of the collecting protocol factor. The collecting protocol of each dataset is different. For instance, MASS subset 1 collects the sleep signal with 256 Hz, while SleepEDF collects the sleep signal with 100 Hz. The collecting protocol of each dataset contains several elements, such as epoch duration. Hence, the different datasets are higher dissimilarity degree than the channel mismatch. When the dissimilarity degree higher, the accuracy gain and f1 gain decrease. We try to find which element in the collecting protocol that most affects the transferability. Therefore, we separate the collecting protocol into two groups, follow below.

1. The dataset uses 30 seconds per epoch duration (SleepEDF, MASS subset 1, and 3).
2. The dataset uses 20 seconds per epoch duration (MASS subset 2, 4, and 5).

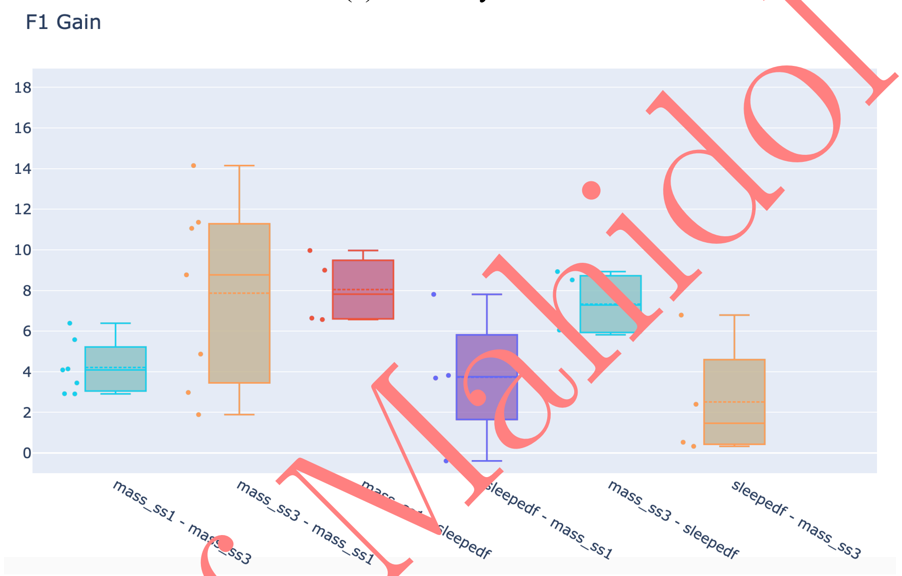
We separate from the epoch duration because the different sampling rates can change the sampling with the least effect. In contrast, changing the data from 30 seconds to 20 seconds is the process that might remove essential waves of sleep signal. Hence, we use this section to discuss the performance of the transferability and unique characteristics of transferability in each group.

5.2.1 Group of 30 seconds per epoch durations

There is one difference in 30 seconds group, which is the sampling rate of the dataset. SleepEDF collects the sleep signal with 100 Hz while the MASS dataset collects the sleep signal with 256 Hz. Signal hertz affects the input size directly because the input size calculated by signal hertz multiply with the epoch duration. Hence, transferring across the dataset, which use different input size, has to downsample and upsample the signal before transferring. To know the impact of the input size, we use the information from figure 5.3a and 5.3b to decision the effect of input sizes factor.



(a) Accuracy Gain



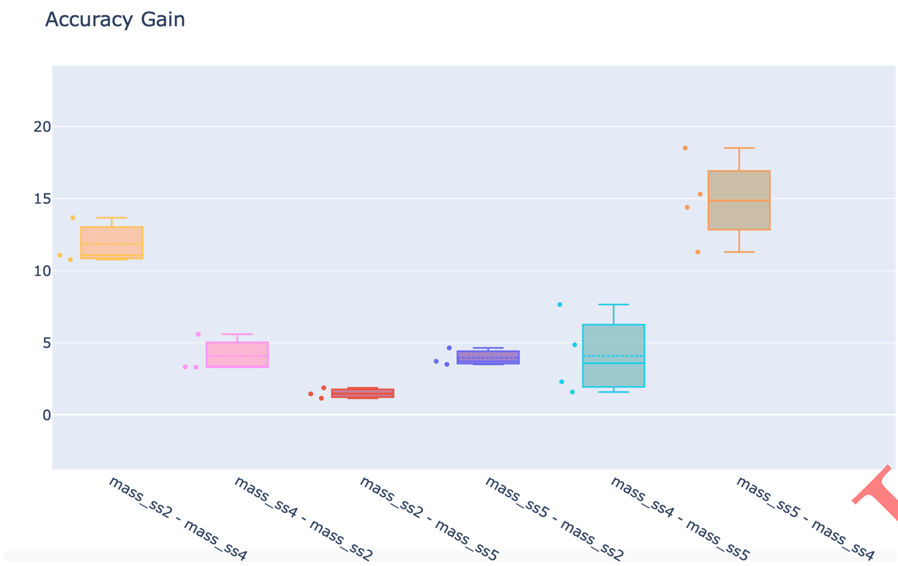
(b) F1 Gain

Figure 5.3: Box Plot Dataset Mismatch Grouping with source dataset and transfer dataset in Group of 30 seconds per epoch durations

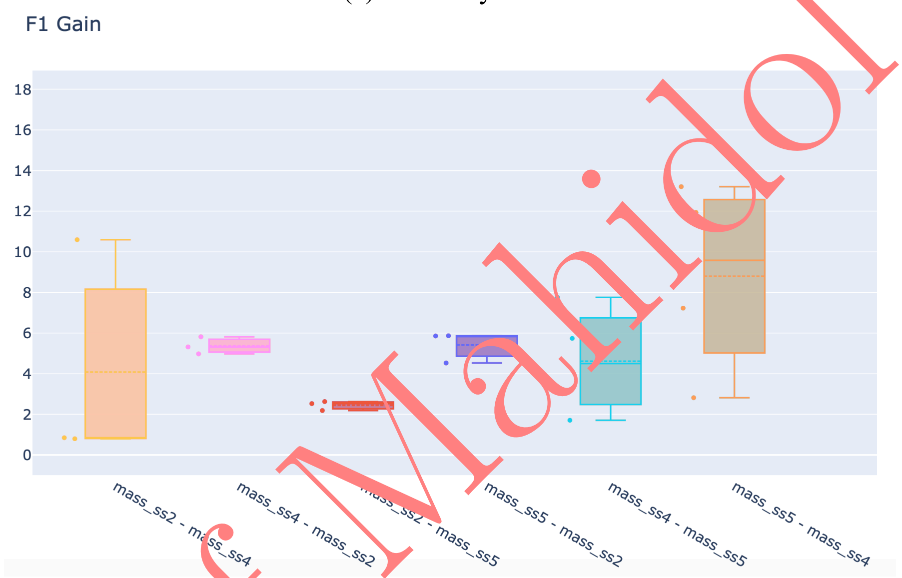
Changing the input size before transferring is the most negative effect factor to the transferability. Reshaping the input size has a high chance of removing the necessary pattern of the sleep signal. It makes the model lose the lesson of some unique pattern of the sleep signal. As this reason match with the boxplot in figure 5.3a and 5.3b. The boxplot shows the transferability from SleepEDF to other datasets is worst than other transferability. It happens because the other subsets have to decrease the length of the signal to match with the input size of SleepEDF. Hence, it has a high chance of removing necessary sleep stage information from the signal. However, transferability from other datasets to SleepEDF provides accuracy gain and f1 gain better than using SleepEDF as a source dataset (D_s). The reason that helps us think SleepEDF should be a target dataset (D_t) is data distribution. From table 1, SleepEDF is the destitute data distribution if comparing with other datasets. Due to this fact, the accuracy gain and f1 gain is seemed high when set the SleepEDF to be the target dataset (D_t).

5.2.2 Group of 20 seconds per epoch durations

MASS subset 2,4, and 5 use epoch duration equal to 20. For the transferability in this group, it does not require the resample of the sleep signal because the input size of these three datasets are the same. Three datasets in this group use almost the same device to collect the sleep signal. Hence, one different thing that these three datasets might different is the location of sleep signal collecting and the environment during the sleep signal collecting.



(a) Accuracy Gain



(b) F1 Gain

Figure 5.4: Box Plot Dataset Mismatch Grouping with source dataset and transfer dataset in Group of 20 seconds per epoch durations

Data distribution is beware factor to the transferability. As figures 5.4a and 5.4b show the boxplot of a specific base position and transfer position. When MASS subset 2 is the source dataset (D_s), the accuracy gain and f1 gain are lower than using MASS subset 2 to be the target dataset (D_t). Due to this information, MASS subset 2 is inferior data distribution. From table 4.4, the data distribution of MASS subset 2 is the lowest (total class = 26,711). Moreover, MASS subset 2 contains N2 class more than necessary. From sleep signal characteristics, the model should more consider with the N3 class and be careful with the N1 class (N1 class is the hardest class to predict because of N1 class size). This makes the MASS subset 2 does not perform good enough to be source dataset (D_s)

The difference environment in transferability is a more negative effect on the transferability than the channel mismatch. The channel mismatch transfer during the same dataset, therefore it minimizes the problem of data distribution while the dataset mismatch has to face the difference of data distribution. Dataset mismatch also faces the different input sizes between the dataset while channel mismatch does not. Input size forces the process to reshape the signal pattern and signal size; hence, the sleep data might lose the significant characteristic. Lastly, the accuracy gain and f1 gain cannot provide better enough as we expect because of the difference between the dataset. ***The most impact effects are data distribution and input size.***

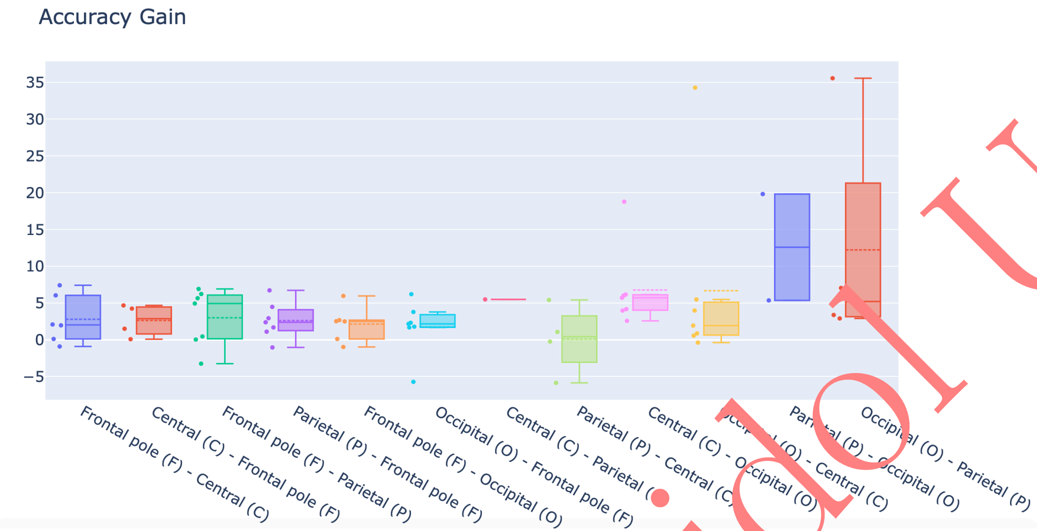
5.3 Different Channels and Environments

Whenever the accuracy gain and f1 gain are not higher, it is the effect of the dissimilarity degree between the source dataset (D_s) and the target dataset (D_t). Transferability across the dataset with the channel mismatch is the highest dissimilarity degree because it includes the dissimilar of the channel and the collecting protocol. Hence, this configuration provides the lowest accuracy gain and f1 gain. To analyze the insight of this configuration, we compare the performance from this configuration with the transferability channel mismatch and transferability dataset mismatch. Therefore, we use this section to illustrate the pattern of transferability with different environments and channels.

5.3.1 Data distribution

The performance of the transferability in the highest dissimilarity degree performs better when the target dataset is low data distribution. The accuracy gain and f1 gain from question 3 configuration should lower than another question. In contrast, the accuracy gain and f1 gain is higher when the transfer dataset is MASS subset 1 (D_t = MASS subset 1 as table 4.5). Hence, we study in-depth with MASS subset 1. We found that the data distribution of MASS subset 1 significantly imbalance (Too much on the wake-up class). Therefore, the deep learning model cannot learn well enough when the size of the training data is too small. This problem does not affect the channel mismatch because it transfers in the same dataset, so distribution is the same.

5.3.2 Comparing with Channel mismatch



(a) Accuracy Gain

(b) F1 Gain

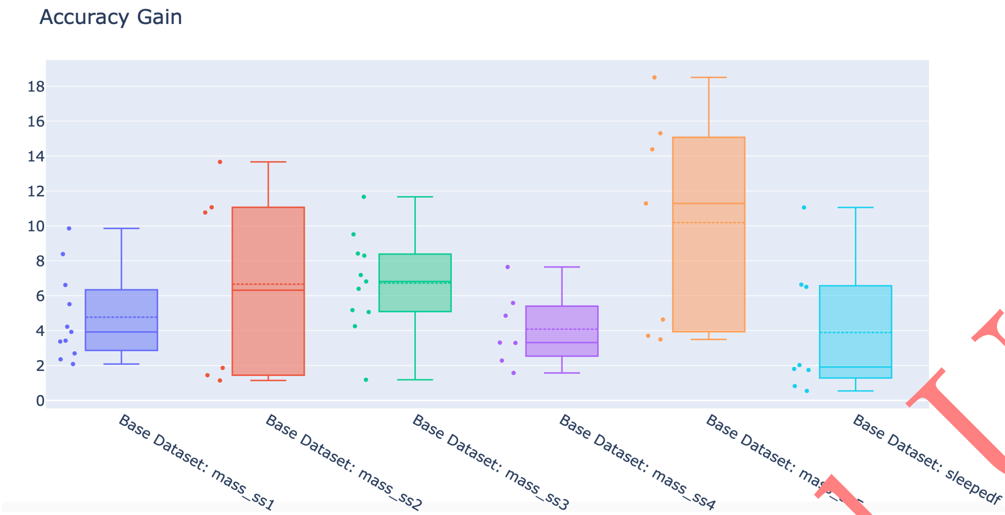
Figure 5.5: Box Plot of transferability across dataset with channel mismatch Grouping from source channel and transfer channel

The pattern of the channel mismatch on question 1 and question 3 is going on the same pattern. As Figure 5.5a and 5.5b, it shows the characteristic of transferability across the dataset with channel mismatch. Moreover, the output of the question 3 look similar to the figure 5.2a and 5.2b, which is the box plot of channel mismatch in the same dataset. Transferability with the nearest channel with the transfer channel is still better to improve the quality of the model. The nearest channel contains two directions. Using the channel in which collecting position more front over is better than the position that collects from back over.

Even though characteristic of the transferability is similar, the accuracy gain and f1 gain still enormously different. The average of the boxplot 5.5a and 5.5b are around 5 to 10 while the average of the box plot 5.2a and 5.2b provide around 10 to 20. The range of the difference between figures 5.2a , 5.2b and figures 5.5a , 5.5b is different too high because of the range of the dissimilarity degree. Transferability across the dataset with channel mismatch contains the dissimilarity between dataset, so accuracy gain and f1 gain decrease have to occur. For example, transferability from Fpz channel to $C4$ channel in MASS subset 2 provides performance gain equal to 4.18 and f1 gain equal to 6.63 (Line 17 from table 1). While transferability from MASS subset 2 with Fpz channel to $C4$ channel in MASS subset 4 provides the accuracy gain equal to 0.11 and f1 gain equal to 1.68 (Line 25 from table 4). Half of the accuracy gain and f1 gain are decreasing because of the collecting protocol of the dataset. If the difference of the collecting protocol minimized, the accuracy gain and f1 gain are better.

The performance gain of the transferability across the dataset with channel mismatch has not forced the performance by the channel mismatch. In contrast, it provides low performance of the transferability because of the difference in the collecting protocol. The difference in collecting protocol has more impact on the transferability more than channel mismatch.

5.3.3 Comparing with Dataset mismatch

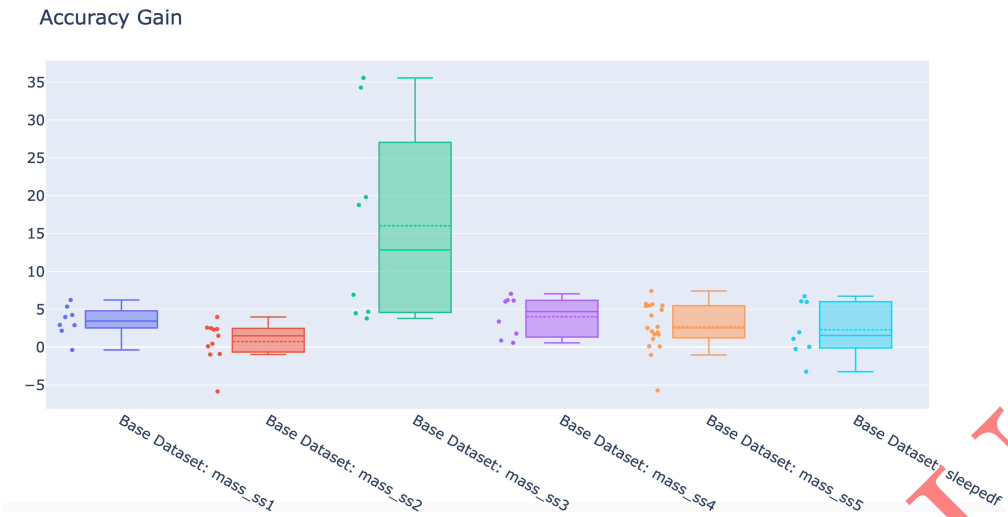


(a) Accuracy Gain

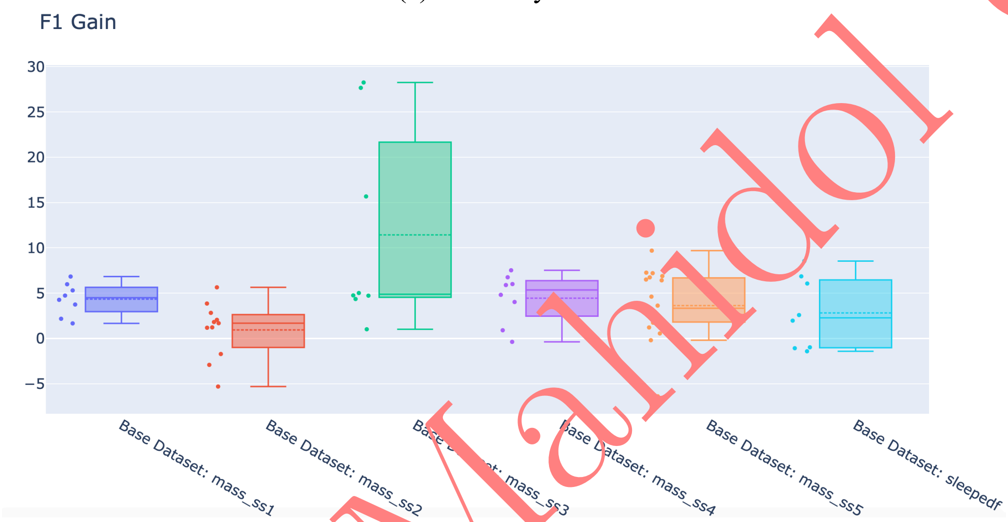
(b) F1 Gain

Figure 5.6: Box Plot of transferability across dataset Grouping fron source dataset

From figure 5.6a and 5.6b, it shows the transferability with dataset mismatch, which grouping the information from the source dataset (D_s). These two figures transfer without channel mismatch. Therefore, we prefer to find the pattern between the transferability across the dataset with channel mismatch and without channel mismatch. The transferability across the dataset still contains two groups, which are a group of 30 seconds and 20 seconds. To find the pattern, we have to compare figure 5.6a, 5.6b, 5.7a, and 5.7b.



(a) Accuracy Gain



(b) F1 Gain

Figure 5.7: Box Plot of transferability across dataset with channel mismatch Grouping from source dataset

The dataset mismatch factor is the most impact factor to the transferability because the performance of the transferability across the dataset with channel mismatch and without channel mismatch is more closely. The accuracy gain and f1 gain of the transferability that contains channel mismatch perform worse than without channel mismatch a little bit. The range of the difference is too close because the dataset is still the factor that more impact the transferability. Due to the dataset impact, the accuracy gain and f1 gain in figures 5.7a and 5.7b less than figures 5.6a and 5.6b around five. While the comparison of the transferability in section 5.3.2 different around ten, it supports that the dataset mismatch always forces the model to make the model cannot improve well. The dataset mismatch factor contains several different protocols to force the performance of the transferability model, such as epoch duration, sampling rate, and signal hertz. Hence, transferability across the dataset with a similar channel as much as it can still be better than across the dataset with strictly different.

Dataset or collecting protocol is the most detrimental effect to the transferability because the difference between dataset contains more than one factor. Moreover, the most impact factor inside the dataset is data distribution. Data distribution provides knowledge to the model. If the data distribution is not good enough, it does not make sense to care about the size of the data. To reduce dataset factor, using the source dataset, which data distribution is more diversify, is be the best procedure to improve the performance of the transferability.

5.4 Selecting Source Dataset and Source Channel

The transferability improves the performance of the model that has to train with the insufficient data. Improvement is not always high if the source dataset and source model have a high dissimilarity degree with the target. To minimize the dissimilarity, it has to consider several factors. First, the environment mismatch or the dataset mismatch is the first factor which the procedure of transferability should consider. Using the source dataset, which uses the epoch duration and sampling rate as same as the target, is a significant factor. Avoiding resamples the training data is to help the model to learn with the train data without removing some important of the train data. Follow the first criteria to help the model to understand the train data fully.

To reduce more impacted factor should focus on the data distribution. Data distribution is one of the most impact factors. When the data distribution is dense into some class, the model fits firmly into the dense class. For example, MASS subset 1 has too much on class N1. Hence, the model from MASS subset 1, which trains with insufficient data, always predicts incorrectly. Consequently, the transferability model has to learn with the target data with unuseful knowledge from MASS subset 1.

However, the environments are not the only ones considerable before transferability. Channel mismatch also is one of the critical factors before transferability. Selecting useful channel help the transferability process to achieve better performance. To know which channel should be transferred is has to consider on the target channel. The excellent channel for transferability is the channel, which is the nearest channel with the target channel. In contrast, the nearest channel contains two directions, which are front over and back over the transfer channel. The channel, which should use, has to be front over a channel with the target channel.

When the selecting source dataset and channel dataset follow the criteria, it helps the transferability to improve the performance. Following all criteria is not required to be strict. If it is hard to focus on every criterion, minimize only the environments between source dataset and target dataset is better.

5.5 Future Works

Attribution Maps is the space that maps the deep learning model into space. Consequently, the performance of transferring is estimated by the distance between the model. If the distance of the two models is close, the performance of the transferring is better than the farther distance. The attribution maps is proved with the image classification [30]. The attribution maps decrease the time to pick up the model for transferring in the image classification that uses convolution neural networks.

We propose to know the attribution maps with the deep learning in sleep stage scoring. We use this section to clarify the sample experiment of the attribution maps with the sleep stage scoring model.

5.5.1 Sleep Stage Scoring Attribution Maps Results

The correlation of two models distance and the performance gain is close to zero (As table 5.1). Due to this fact, the correlation that closes to zero means the distance, and the performance gain does not have any pattern or step to estimate each value.

We can use attribution maps to estimate the performance of the transferring image classification because the image classification uses only the convolution neural networks (CNN). In contrast, sleep stage scoring use convolution and recurrent neural networks

(CNN and RNN). The RNN uses the previous stage to calculate the current stage. Hence, the attribution maps collect only the focus point that the model use to predict the current stage of the information without using the previous information to help the prediction.

In conclusion, the attribution maps estimate the transferring better when the deep learning model architecture using only the convolution neural network. The attribution maps randomly estimate the transferring if the deep learning model architecture uses the recurrent neural networks.

Table 5.1: The correlation of two models distance and the performance gain

	Distance	Accuracy Gain	F1 Gain
Pearson correlation of distance calculated by Saliency Map	1.000000	-0.133823	[illegible]0.210646
Spearman correlation of distance calculated by Saliency Map	1.000000	-0.184059	-0.[illegible]21279
Pearson correlation of distance calculated by Epsilon Map	1.000000	-0.057535	-0.020[illegible]
Spearman correlation of distance calculated by Epsilon Map	1.000000	-0.0[illegible]4943	0.068388
Pearson correlation of distance calculated by Integrate Gradient Map	1.000000	-0.302[illegible]29	-0.253807
Spearman correlation of distance calculated by Integrate Gradient Map	1.000000	-0.097326	-0.121131

CHAPTER 6
CONCLUSIONS

The study of transfer learning in sleep stage scoring is trying to solve the problem of labeling the sleep signal. Generally, sleep experts have to label the sleep signal each second manually. In the present, several techniques help sleep experts to analyze the sleep stage. The popular technique is deep learning. The deep learning technique helps the sleep expert to label the sleep signal without any feature extraction. However, using deep learning still have a limitation about the size of train data. Deep learning technique requires a massive size of training data, which the medical data cannot provide. Medical data is expensive if we need to collect a large size of it. Hence, we propose the transfer learning technique to solve the size of the training data problem. We also propose to study the factor that impacts the transfer learning technique.

The transfer learning technique is using the model, which trained from a large size of training data, as a starting point of the target model. This technique uses to help the Deep learning model, which has to train with insufficient data, more accurate. The source data and the target data should have similar characteristics. If those data are not similar, the transfer learning technique might not work. This technique helps with insufficient training data.

The transfer learning technique is not researched in the sleep stage scoring field. Hence, we aim to research the transfer learning technique in sleep stage scoring. Our research also studies the factor that impacts the performance of the transferability. Before we get the information about the factor, we have to find the public dataset about the sleep stage scoring. The sleep stage scoring dataset includes two sets, which used in this research. Two datasets are SleepEDF and MASS. The MASS dataset contains five subsets from different countries in Europe. Each of the datasets uses a different protocol and environment to collects the sleep signal, such as the sampling rate of collecting sleep signals. Moreover, each dataset contains several channels. The channels can group as five classes, which are Frontal Channel, Central Channel, Parietal Channel, and Occipital Channel sorted by front to back position. From the public dataset, our research group it to three research questions.

The first research question focuses on the impact of channel mismatch on the transferability. In this question, we prefer to know the accuracy gain and f1 gain of the transferability when transfer in the same environment with channel mismatch. From the experimental result, it shows the nearest channel with the target channel performs better than the farther channel. In contrast, the nearest channel contains two directions. To pick the channel that suitable for transferability should pick the channel which fronts over the target channel. This research question transferred on the same dataset, such as transferability in the SleepEDF, transferability in the MASS subset 1. Transferability in the same dataset helps the process to minimize the different environments and data distribution factors.

The second research question focuses on the differences between datasets. To reduce the impacted factor that is not the dataset, we transferability across the dataset with channel mismatch. After the experiment, the result separated into two groups. Two groups are different with the epoch duration.

Dataset, which uses 30 seconds per epoch duration, is the first group of dataset mismatch. This group contains three datasets, which are SleepEDF, MASS subset 1, and MASS subset 3. From this group, we can summarize that resample the data to be the same with the transfer dataset impact to the transferability directly. For example, the model has a high chance of learning with the data that does not contain the vital signal wave when the model transferred from the SleepEDF. Due to this fact, SleepEDF dataset uses different input sizes. Transferred from it has to reshape the data before starting the transferability. Reshape the training data might add unnecessarily or remove necessary information.

Another dataset group uses 20 seconds per epoch duration, which contains MASS subset 2, 4, and 5. After the experiment, we found more factors that impact the performance of the transferability. Data distribution is one of the factors, which affects transferability directly. If the source dataset contains inferior data distribution, the knowledge does not help when training with the target dataset. Moreover, the dataset, which should not use to be a source dataset, is the dataset that contains too much wake-up class.

The last research question is transferability across the dataset with channel mismatch. This configuration is the highest dissimilarity degree because it combines the configuration from question 1 and question 2. When the source and target dataset strictly different, the pattern of it still process in the same way with question 1 and 2. Due to this information, we separate the result into two parts to compare with the pattern from questions 1 and 2.

The first group in the last question is comparing with the channel mismatch. The conclusion in this group is a similar pattern with the channel mismatch in question 1. From the result, it also better when the source channel and target channel are the nearest positions. However, the range of accuracy gain and f1 gain still far from each. It too far because of the dataset mismatch. Hence, we can summarize a little bit that the channel mismatch still goes on the same pattern while the dataset mismatch forces the performance of the transferability.

The second group in the last question is comparing with the dataset mismatch. The result is a similar pattern with the dataset mismatch. Moreover, the accuracy gain and f1 gain are close to the dataset mismatch. It shows that the different environment is the factor that forces the performance of the transferability. Due to the dataset is the most impact factor, minimize the dissimilarity degree of the dataset mismatch helps the performance of transferability directly.

From the experiment result in our research question, the most impact factor is the different environment. The pattern and the performance of the transferability in the dataset mismatch and question 3 configuration go in the same pattern with the lowest range. This supports the dataset mismatch to be the most impact factor. Due to this fact, the dataset contains several environments that force transferability performance. For example, data distribution, epoch duration, input sizes. The different dataset increases the dissimilarity degree directly. Reduce the dataset dissimilarity might improve the performance transferability more than reduce the channel mismatch factor.

The transferability improves the quality of the deep learning model, which face insufficient training data problem. It helps the deep learning model to learn with useful data before train with the new data. This is an essential part of the sleep stage scoring.

In the nearest future, the wearable device, which use to monitor the sleep stage, is a necessary device with unhealthy people. It is too possible to collect a massive size of sleep signal from a wearable device. For future work, we would like to experiment with the sleep signal collected by the wearable device. Hence, it minimizes the cost of collecting the medical data in the sleep signal fields. The cost that the patients have to spend on monitoring the sleep stage is minimizing with transfer learning technique. Lastly, the impacted factor, which our research suggests, help the doctor to avoid lousy factor before transferability the deep learning model to the wearable device.

REFERENCES

[1] Goldberger AL., Amaral LAN., Glass L., Hausdorff JM., Ivanov PC., Mark RG., et al., "PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals";101(23), [Online]. Available: https://www.ahajournals.org/doi/10.1161/01.CIR.101.23.e215.

[2] O'Reilly C., Gosselin N., Carrier J., Nielsen T., "Montreal Archive of Sleep Studies: an open-access resource for instrument benchmarking and exploratory research";23(6):628–635, [Online]. Available: http://doi.wiley.com/10.1111/jsr.12169.

[3] Wei Y., Colombo MA., Ramautar JR., Blanken TF., van der Werf YD., Spiegelhalder K., et al., "Sleep Stage Transition Dynamics Reveal Specific Stage 2 Vulnerability in Insomnia";40(9), _eprint: https://academic.oup.com/sleep/article-pdf/40/9/zsx117/19920560/zsx117.pdf, [Online]. Available: https://doi.org/10.1093/sleep/zsx117.

[4] Supratak A., Dong H., Wu C., Guo Y., "DeepSleepNet: A Model for Automatic Sleep Stage Scoring Based on Raw Single-Channel EEG";25(11):1998–2008, [Online]. Available: https://ieeexplore.ieee.org/document/7961240/.

[5] "10–20 system (EEG) - Wikipedia";, [Online]. Available: https://en.wikipedia.org/wiki/10%E2%80%9320$_{s}ystem_{(}EEG)$.

[6] Medic G., Wille M., Hemels M., "Short- and long-term health consequences of sleep disruption";Volume 9:151–161, [Online]. Available: https:// www.dovepress.com/ short–and-long-term-health-consequences-of-sleep-disruption-peer-reviewed-article-NSS.

[7] Nobrega RVMd., Peixoto SA., Silva SPPd., Filho PPR., "Lung Nodule Classification via Deep Transfer Learning in CT Lung Images", In: 2018 IEEE 31st

International Symposium on Computer-Based Medical Systems (CBMS). IEEE;. p. 244–249, [Online]. Available: https://ieeexplore.ieee.org/document/8417245/.

[8] Shie CK., Chuang CH., Chou CN., Wu MH., Chang EY., "Transfer representation learning for medical image analysis", In: 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). IEEE;. p. 711–714, [Online]. Available: https://ieeexplore.ieee.org/document/7318461/.

[9] Liu P., Qiu X., Huang X., "Recurrent Neural Network for Text Classification with Multi-Task Learning";p. 7.

[10] Andrillon T., Nir Y., Staba RJ., Ferrarelli F., Cirelli C., Tononi G., et al., "Sleep Spindles in Humans: Insights from Intracranial EEG and Unit Recordings";31(49): 17821–17834, [Online]. Available: http://www.jneurosci.org/cgi/doi/10.1523/JNEUROSCI.2604-11.2011.

[11] "CS231n Convolutional Neural Networks for Visual Recognition";, [Online]. Available: http://cs231n.github.io/convolutional-networks/.

[12] Sultana F., Sufian A., Dutta P., "Advancements in Image Classification using Convolutional Neural Network";p. 122–129, [Online]. Available: http://arxiv.org/abs/1905.03288.

[13] Goodfellow I., Bengio Y., Courville A., Deep Learning, MIT Press; 2016, http://www.deeplearningbook.org.

[14] Mousavi S., Afghah F., Acharya UR., "SleepEEGNet: Automated sleep stage scoring with sequence to sequence deep learning approach";14(5):e0216456, [Online]. Available: http://dx.plos.org/10.1371/journal.pone.0216456.

[15] Zhang L., Fabbri D., Upender R., Kent D., "Automated sleep stage scoring of the Sleep Heart Health Study using deep neural networks";42(11):zsz159, [Online]. Available: https://academic.oup.com/sleep/article/doi/10.1093/sleep/zsz159/5530377.

[16] Phan H., Andreotti F., Cooray N., Chén OY., De Vos M., "SeqSleepNet: End-to-End Hierarchical Recurrent Neural Network for Sequence-to-Sequence Automatic Sleep Staging";27(3):400–410, [Online]. Available: http://arxiv.org/abs/1809.10932.

[17] Hsu YL., Yang YT., Wang JS., Hsu CY., "Automatic sleep stage recurrent neural classifier using energy features of EEG signals";104:105–114, [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0925231212008387.

[18] Tsinalis O., Matthews PM., Guo Y., "Automatic Sleep Stage Scoring Using Time-Frequency Analysis and Stacked Sparse Autoencoders";44(5):1587–1597, [Online]. Available: http://link.springer.com/10.1007/s10439-015-1444-y.

[19] Tsinalis O., Matthews PM., Guo Y., Zafeiriou S., "Automatic Sleep Stage Scoring with Single-Channel EEG Using Convolutional Neural Networks";[Online]. Available: http://arxiv.org/abs/1610.01683.

[20] Dong H., Supratak A., Pan W., Wu C., Matthews PM., Guo Y., "Mixed Neural Network Approach for Temporal Sleep Stage Classification";26(2):324–333, [Online]. Available: http://arxiv.org/abs/1610.06421.

[21] Phan H., Chén OY., Koch P., Lu Z., McLoughlin I., Mertins A., et al., "Towards More Accurate Automatic Sleep Staging via Deep Transfer Learning";[Online]. Available: http://arxiv.org/abs/1907.13177.

[22] Ioffe S., Szegedy C., "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift";[Online]. Available: http://arxiv.org/abs/1502.03167.

[23] Hochreiter S., Schmidhuber J., "Long Short-Term Memory"; 9(8): 1735–1780, [Online]. Available: http://www.mitpressjournals.org/doi/10.1162/neco.1997.9.8.1735.

[24] Casson AJ., Smith S., Duncan JS., Rodriguez-Villegas E., "Wearable EEG: what is it, why is it needed and what does it entail?", In: 2008 30th Annual International

Conference of the IEEE Engineering in Medicine and Biology Society. IEEE;. p. 5867–5870, [Online]. Available: http://ieeexplore.ieee.org/document/4650549/.

[25] Brown L., van de Molengraft J., Yazicioglu RF., Torfs T., Penders J., Van Hoof C., "A low-power, wireless, 8-channel EEG monitoring headset", In: 2010 Annual International Conference of the IEEE Engineering in Medicine and Biology. IEEE;. p. 4197–4200, [Online]. Available: http://ieeexplore.ieee.org/document/5627393/.

[26] Kosmadopoulos A., Sargent C., Darwent D., Zhou X., Roach GD., "Alternatives to polysomnography (PSG): A validation of wrist actigraphy and a partial-PSG system";46(4):1032–1041, [Online]. Available: http://link.springer.com/10.3758/s13428-013-0438-7.

[27] Kingma DP., Ba J., "Adam: A Method for Stochastic Optimization";[Online]. Available: http://arxiv.org/abs/1412.6980.

[28] Cohen J., "A Coefficient of Agreement for Nominal Scales";20(1):37–46, [Online]. Available: http://journals.sagepub.com/doi/10.1177/001316446002000104.

[29] Sokolova M., Lapalme G., "A systematic analysis of performance measures for classification tasks"; 45(4): 427–437, [Online]. Available: https:// linkinghub.elsevier.com/retrieve/pii/S0306457309000259.

[30] Song J., Chen Y., Wang X., Shen C., Song M., "Deep Model Transferability from Attribution Maps";[Online]. Available: http://arxiv.org/abs/1909.11902.

BIOGRAPHIES

NAME	Mr. Worawit Luewichana
DATE OF BIRTH	10 January 1998
PLACE OF BIRTH	Bangkok, Thailand
INSTITUTIONS ATTENDED	Suankularb Wittayalai School, 2016:
	High School Diploma
	Mahidol University, 2019:
	Bachelor of Science (ICT)

NAME	Mr. Chaiyakarn Khanan
DATE OF BIRTH	31 August 1997
PLACE OF BIRTH	Bangkok, Thailand
INSTITUTIONS ATTENDED	Suankularb Wittayalai School, 2016:
	High School Diploma
	Mahidol University, 2019:
	Bachelor of Science (ICT)

NAME	Miss Pranpriya Shatsena
DATE OF BIRTH	18 October 1997
PLACE OF BIRTH	Bangkok, Thailand
INSTITUTIONS ATTENDED	Triamudomsuksapattanakarn School, 2016:
	High School Diploma
	Mahidol University, 2019:
	Bachelor of Science (ICT)