Develop

Brain Computer Interface for Mindfulness Training

Author	Oh Zhi Jie
Supervisor	Prof Guan Cuntai
Examiner	Associate Prof Bo An
Project ID	SCE17-0121

Submitted in fulfilment of the requirements for the Degree of Bachelor of Engineering (Computer Science) of Nanyang Technological University

AY 2017/2018

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

NANYANG TECHNOLOGICAL UNIVERSITY

ABSTRACT

Being mindfulness is shown to have wide ranges of health benefits. Many existing implementations of achieving mindfulness are focused on scheduled mediation and playback of smoothing nature sounds and scenes. However, these implementations are not able to provide any feedback to the user.

This report aims to assist in designing a real-time feedback for mindfulness meditation. It is an Android application which uses EEG to predict the current state of the user using SVM. Real-time feedback during the session will be provided to bring users back from distraction and assist them in achieving deeper mindfulness.

This project explores the possibility accurately predicting the user state as well as to lay down the foundations to create the real-time feedback for mindfulness meditation.

ACKNOWLEDGEMENT

I express deepest gratitude to my supervisor, Professor Guan Cuntai, for his constant guidance, professional ideas and support throughout the entire project. He had always made time for me to clarify my doubt and had always been willing to offer his help to the best of his abilities.

I would like to express my wholehearted gratitude to Mr. Aung Aung Phyo Wai for his guidance, experience, suggestion and feedback for my project. His experience in electroencephalogram processing and Support Vector Machine has allowed me to gain insights in this field.

Lastly, I would like to express my thank the participants Tan Ying Hao, Oh Zhi Chao, Wong Yi Jie, Eugene Tan Ping Hong who has taken their time out during their busy schedule to assist me in my experiments.

Without all the aforementioned people's help and support provided throughout the course of my Final Year Project, this report would not have been made possible.

Abbreviations

EEG	Electroencephalogram
SVM	Support Vector Machine
CSV	Comma Separated Values
MIST	Montreal Imaging Stress Task
BCI	Brain Computer Interface

Table of Contents

List of Figures

List of Tables

Chapter 1: **Introduction**

1.1 Background

There has been an increasing amount of research which supports mental health improvements based on participating in mindfulness training. Recent literatures suggest that mindfulness training can lead to improvements in psychological functioning in a wide range of populations. Mindfulness Meditation have been shown to provide significant benefits for health and well-being particularly in stress, depression and anxiety [1], [2], [3].

Achieving mindfulness state requires an individual to focus his attention to the present moment. The skill of maintaining one's attention to the present moment is fundamental to mindfulness meditation. Mindfulness state is not only difficult to achieve and maintain, it is also difficult to be explained and described. Due to the lack of external cues, it is challenging to assess and monitor mindfulness state. While achieving mindfulness can be trained, there is a lack of methods and tools to monitor mindfulness, provide real-time feedback on the current meditation session.

At this point of writing, Muse [4] is the only device that provides real-time feedback for meditation. It provides feedback by measuring the brain signals and converting it into sounds of wind. When the mind is calm, calm and settled winds are given as feedback while when the mind is active, the winds will pick up and blow. However, Electroencephalogram (EEG) measurements are not always accurate due to ambient noise which may result in providing the wrong feedback. The real-time feedback provided by Muse is intrusive and when detected inaccurately, not only will it disrupt the user, it also reduces the user's sense of satisfaction of using Muse.

1.2 Objective

This project aims to propose a novel algorithm to quantify mindfulness level and to provide non-intrusive feedback to remind users when they are distracted as well as to reward when the user is in mindfulness state.

1.3 Scope

An Android application, "EEG Data Collection", will be developed to collect the EEG of Mindfulness state and the Active state of the participants using Muse. The participants are required to perform guided meditation and arithmetic stressor to build generic model of Mindfulness and Active state respectively.

Features are extracted from the collected EEG data and are used as training dataset to train and generate classification model using Support Vector Machine (SVM). The generated model is then used for real-time prediction of Mindfulness state or Active state.

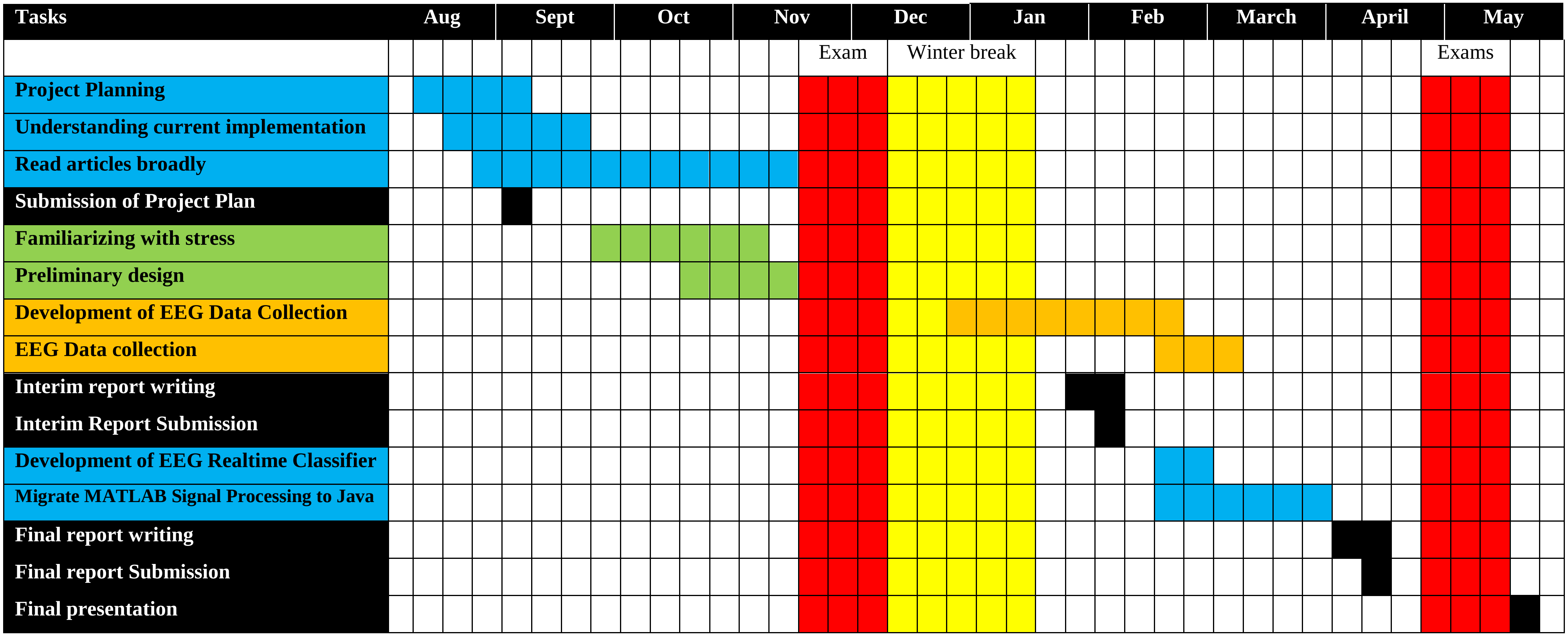
A second Android application, "EEG Realtime Classifier" will be developed to predict the state of the user in real-time by using the SVM model.

1.4 Report Organization

This document is organized as follows:

- **Chapter 1:** The introduction of this project
- **Chapter 2:** Literature review will introduce the various concepts and on EEG and highlight other works and devices which are created to improve mental health
- **Chapter 3:** Introduces the software and hardware tools which are required and used for the development of this project
- **Chapter 4:** Establishes proposed approach and explains the technique and methods used in this project
- **Chapter 5:** Describe in detail the process of EEG data collection
- **Chapter 6:** Establish the procedures in EEG data processing and the generation of the SVM model
- **Chapter 7:** Showcase the initial implementation of real-time prediction of user state
- **Chapter 8:** Describe and explain the finding on the results
- **Chapter 9:** Conclude the findings and discuss what future work can be done

1.5 Project Schedule



Chapter 2: **Literature Review**

2.1 Electroencephalogram

EEG is the method to record electrical activity of the brain by using electrodes placed along the scalp. Measuring the electrical activity from the brain is useful as it reflects electrical impulses when the different neurons communicate with each other.

EEG signals can be typically described by rhythmic activity and transients. The activity can be divided into bands of frequency. Certain mental states can be described using these frequency bands. Table 2-1 lists the five frequency bands by frequency ranges and normal mental states in each frequency band.

Table 2-1: Comparison of EEG bands [5]

Bands	Frequency (Hz)	Normal Mental State
Delta	1 – 4	- Adult slow wave sleep - Found in continuous-attention task
Theta	4 – 8	- Drowsiness in adults and teens - Associated with inhibition of elicited response.
Alpha	8 – 12	- Closed eyes - Relaxed
Beta	12 – 25	- Active thinking - Focus - High alert - Anxious
Gamma	Above 25hz	- Cross-modal sensory processing (perception that combines 2 senses) - Short-term memory matching of recognized object, sound,

2.2 Bandpass Filter

A bandpass filter is a function that combines both low pass filter and high pass filter. It removes unwanted frequencies from the received signal and allows the selected range of frequencies to be decoded.

2.3 Artifact removal

Muscle activity generates currents that can be picked up by the electrodes. The closer the muscle is to the electrode, the stronger their impact will be on the recording. These unwanted currents are called the artifacts. The most apparent offenders are the facial muscle, neck muscle and jaw which have severe effects on EEG recording [5].

These muscle movements are serious contamination of EEG activity cause problems for EEG interpretation and analysis. A naive rejection of the contaminated EEG results can cause considerable loss of the collected information. Therefore, it is important to have a good algorithm for artifacts removal.

2.4 Features Extraction

Features extraction starts from an initial set of measured data and builds derived values (features) intended to be the most informative and non-redundant to facilitate the subsequent machine learning for classification.

Feature extractions are done when the input data are too large to be processed or are suspected to be redundant. It can be processed and transformed into a reduced set of features. This process is called feature selection [6]. These selected features are expected to contain the most relevant information so that the desired task can be performed with these reduced data set instead of the complete original data.

2.5 Support Vector Machine

SVM [7] is a supervised machine learning algorithm which can be used for classification problems. It uses a technique called the kernel trick which transforms the data into high dimensional feature spaces allowing it to find a boundary between the possible states. The new data are then mapped into that same space and predicted to belong to a state based on a threshold to determine which side of the boundary they belong to.

2.6 Alternative devices for health and mental wellness

In present times, there are many devices which aim to provide wellness using different techniques. One example is Thync [8], which provides electromagnetic impulses to simulate cranial nerves on the face and back of the head to improve mood.

Spire [9] is another device which measures respiration and classifies respiratory patterns into cognitive or emotional state. Monitoring respiration allows the device to detect stress and remind the users to take a deep breath and calm down.

Chapter 3: **Program and Tools**

3.1 Introduction

This chapter will provide a description of the hardware and software tools which are involved in the project development.

3.2 Hardware Tools

3.2.1 Muse Headset

Muse [4], is a consumer-grade EEG Device. It records EEG signals using dry electrodes and sends data back wirelessly thus making it simple and easy to use. They are also relatively inexpensive as compared to research grade EEG device such as Emotive EPOC+ [10]. The Muse headset was tested, and it demonstrated that it has the potential to be used as a research tool [11] although it would have more error as compared to research grade devices due to simplified design and lower number of sensors.

3.2.2 Samsung Galaxy Tab S2 9.7 inch

The Samsung Galaxy Tab S2 9.7 inch is employed as a device to record EEG signal using Muse and to display real-time EEG state classification. The Tab S2 has a large display which allows the user to read instructions as displayed by the Android application. It also has a sufficiently fast processor to perform real-time EEG data pre-processing for state classification.

3.3 Software Tools

3.3.1 Android Studio IDE

Android Studio IDE was used as a development platform for the android applications to be developed namely, EEG Data Collection and Real-time EEG Classifier. At the point of writing, the version used is 3.1.1.

3.3.2 MATLAB

MATLAB R2017b is used to do the EEG pre-processing, training and classification testing. It is chosen for its ease of use in signal processing and existing libraries of useful algorithms.

3.3.3 Muse Player

MusePlayer [12] is a utility used for converting EEG signals from the native Muse file format into other file formats such as Comma Separated Values (CSV) or MATLAB. It is used to convert the collected EEG signals to CSV for the ease of processing the data.

3.3.4 Python

Python scripts are used to pre-process the EEG signals in CSV file to make the data easier to work with in MATLAB. Python is chosen as it is easy to use, powerful and versatile which allow for rapid development of a simple script to pre-process CSV into a simple to work with format.

Chapter 4: **Proposed Approach**

This chapter discusses the author's approach to develop brain computer interface (BCI) for mindfulness training. The step-by-step procedure to create real-time mindfulness classification is implemented with three phrases as seen in Figure 4-1.

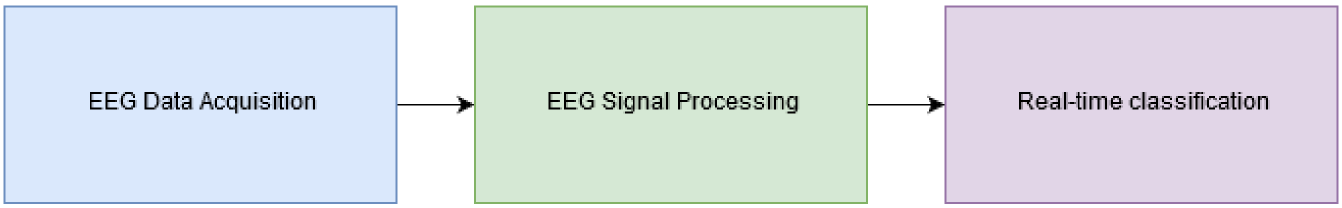


Figure 4-1: Overview of Proposed Approach

Active state and Mindfulness state will be collected during the EEG Data Acquisition phrase. Next, the raw EEG data is pre-processed during the EEG Signal Processing phrase to remove artifacts and extract features. The extracted features are then labelled as Active state and Mindfulness State and are used as training data for SVM training algorithm to generate a model. Finally, he generated SVM model allows for SVM to classify new data into one of the two states during the real-time Classification phrase.

4.1 EEG Data Acquisition

4.1.1 Overview

The data acquisition phrase requires two types of mental states namely, the Active state and the Mindfulness state. The Active state is associated with brain activity whereas the Mindfulness state is associated with meditation. In this project, an arithmetic stressor Montreal Imaging Stress Task (MIST) [13] and Guided Meditation is used to induce Active states and Mindfulness states respectively.

The EEG signal data will be collected using an Android application, "EEG Data Collection" for the purpose of this task.

4.1.2 Arithmetic Task for stress state

The participant is required to perform an arithmetic stressor, MIST, for the collection of Active state data. MIST is an arithmetic stressor task which generates mathematical expression of with three numbers and two operands ("+" for addition, "-" for subtraction, "*" for multiplication and "/" for division). The algorithm has been designed to generate mathematical expressions in which the solution will be an integer from 0 to 9. Hence only a single input will be required to submit as an answer.

This task is spilt into two sections, control and experimental. During the control section, a series of mental arithmetic tasks are displayed on the screen and the participants will submit their answers by tapping on the response interface.

In the experimental condition, the time limit of the task is set to be beyond the ability of the participants. The timeout of each arithmetic task will be set as 10% lesser than the average time achieved during the control condition. Additionally, the algorithm will constantly measure the participants' response and it will reduce or increase the timeout by 10% if the participants answered correctly or wrongly consecutively for three times. It is expected that this will keep the percentage of correct answers to be between 20% – 45% and will enforce a constant stress on the participant.

In addition, the participant is pre-informed that the average percentage of correct answers 80%-90% and are told that their performance should be close to the average for their data to be considered in the project. This will create additional stress on the participant due to their inability to hit the average score.

4.1.3 Guided Meditation for mindfulness state

Guided meditation has been shown to produce relaxation response in young adults with no meditation experiences [14]. As such, guided meditation can be used for guiding participants to enter mindfulness state.

The participants are provided with Guided Mindfulness Meditation Track by The Honest Guys [15] which will provide instructions to reach the state of mindfulness. The instructions come in the form of concentrating on their breath followed by suggestions to peace, stillness and thoughtless or peaceful stage. Earphones are to be worn by the participants during process of the guided mediation.

4.2 EEG Signal Processing

4.2.1 Overview

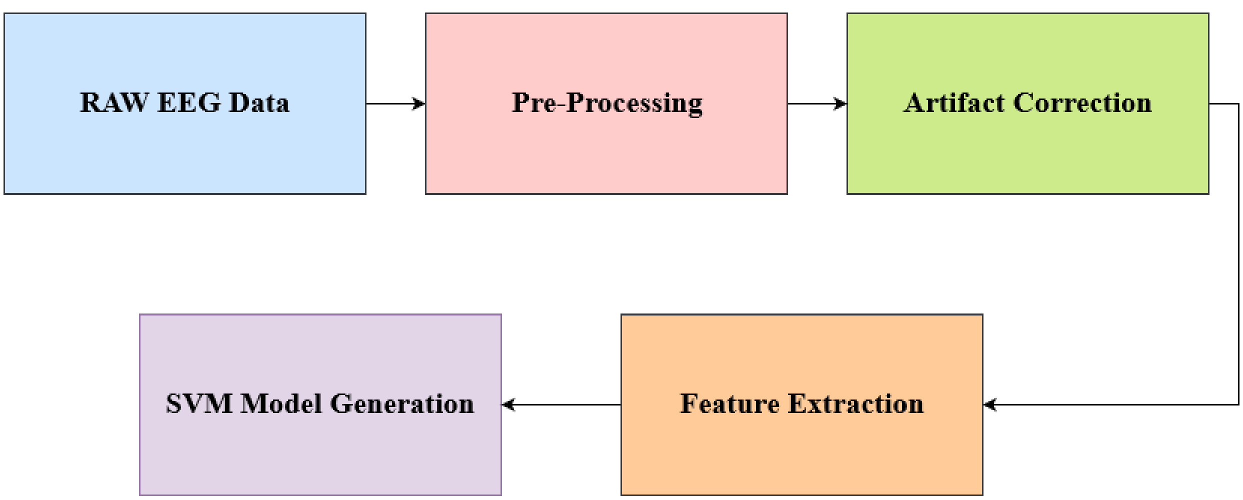


Figure 4-2: EEG Processing Sequence

Figure 4-2 shows the entire process of processing raw EEG signal data to the final state which will be use as dataset for SVM model training.

4.2.2 Pre-processing

EEG Signal data collected using the Muse device are in the Muse file format. The data file is first converted into CSV file using MusePlayer for the ease of processing. Important segments such as during Arithmetic Task and Guided Meditation are extracted out and labeled as “1” and “0” respectively. These data are then ready to be used as training data for the supervised learning machine learning using SVM.

4.2.3 Artifact Removal

The pre-processed EEG signal data are passed through a bandpass filter to filter the important frequency bands of the signals from 0.3hz to 45hz. Artifact removal is performed on the filtered data using multi-resolution moving average [16].

4.2.4 Feature Extraction

The artifact removed EEG signal data are then passed through another bandpass filter to separate the signal into different frequency bands.

Each feature is extracted from a two second window with one second overlap. For example, with three seconds worth of data, two features can be extracted. In Figure 4-3, it shows the first feature being extracted out from the first two seconds of data. In Figure 4-4, the second feature is extracted from the last two second of data.

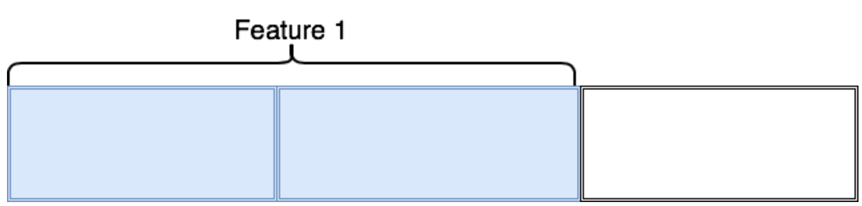


Figure 4-3: Example Feature Extraction 1

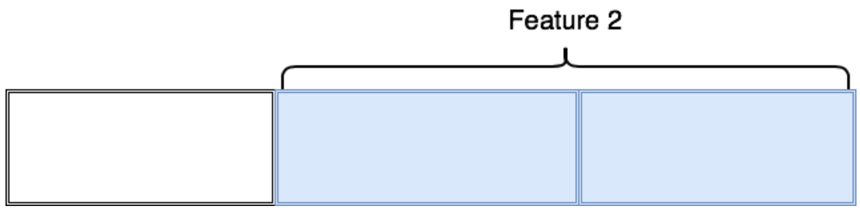


Figure 4-4: Example Feature Extraction 2

The features of each window are extracted by calculating the relative power of the entire window [17].

4.2.5 SVM supervised learning

The extracted features are used as training dataset for SVM to train to generate a SVM classification model.

The generated classification model is validated using the 10-fold cross-validation method [18]. The dataset is partitioned into 10 equal segments, 9 segments used for training and 1 segment used for validation. The cross-validation process is then repeated 10 times with each of the segments use exactly once to as validation data. The results are then averages to produce a single estimation score of the classification model.

4.3 Real-time EEG Classifier

The trained classification mode is ported over to the android application and used to predict the state of the participant in real-time.

The raw EEG received from the Muse headband is processed using artifact removal and feature extraction. The extracted features are used for the prediction of the current meditation state. A meter will be displayed to show the current meditation score. A graph will be displayed to show the scores over last 45 seconds. Refer to chapter 7 for the images of the meter and graph.

4.3.1 Ensure all functions from MATLAB are the same as in java

The signal processing such as artifact removal, bandpass filter, feature extraction is written in MATLAB. To ensure the implementation of the converted MATLAB function to Java function is exactly the same, unit test is written.

Input and outputs from each function of the MATLAB function is saved as a CSV file, with each value having to 16 decimals places. The test is then written to ensure the value of the output from the Java function is the same as the output from MATLAB. However, internal representation of floating point values varies slightly from each system. Hence, a leeway of +-0.001 must be given to verify the output.

Chapter 5: **EEG Data Collection**

5.1 Overview

This chapter will explain the entire process of EEG data collection using the Android application "EEG Data Collection".

Table 5-1: Overview of EEG Data Collection

Process	Actions
Connect to Muse	• Connect to select Muse headband • Ensure headband connection is stable and excellent
Arithmetic Training	• Series of arithmetic questions are displayed • Participants are to answer the questions correctly
Guided Meditation	• Guided meditation track will be played • Participants are to follow the instructions given by the meditation track enter the state of mindfulness
Arithmetic Test	• Participants are reminded for their data to be used, their results must be of 80% and above which is near the average • Series of arithmetic question are displayed with a timer which allows for 10% less than their average time taken during the Arithmetic Training. • Participants are to answer the questions correctly before timeout.

5.2 Setup

5.2.1 Prerequisites

You need to clone or install

- FocusDataCollection [19]

5.2.2 Program setup

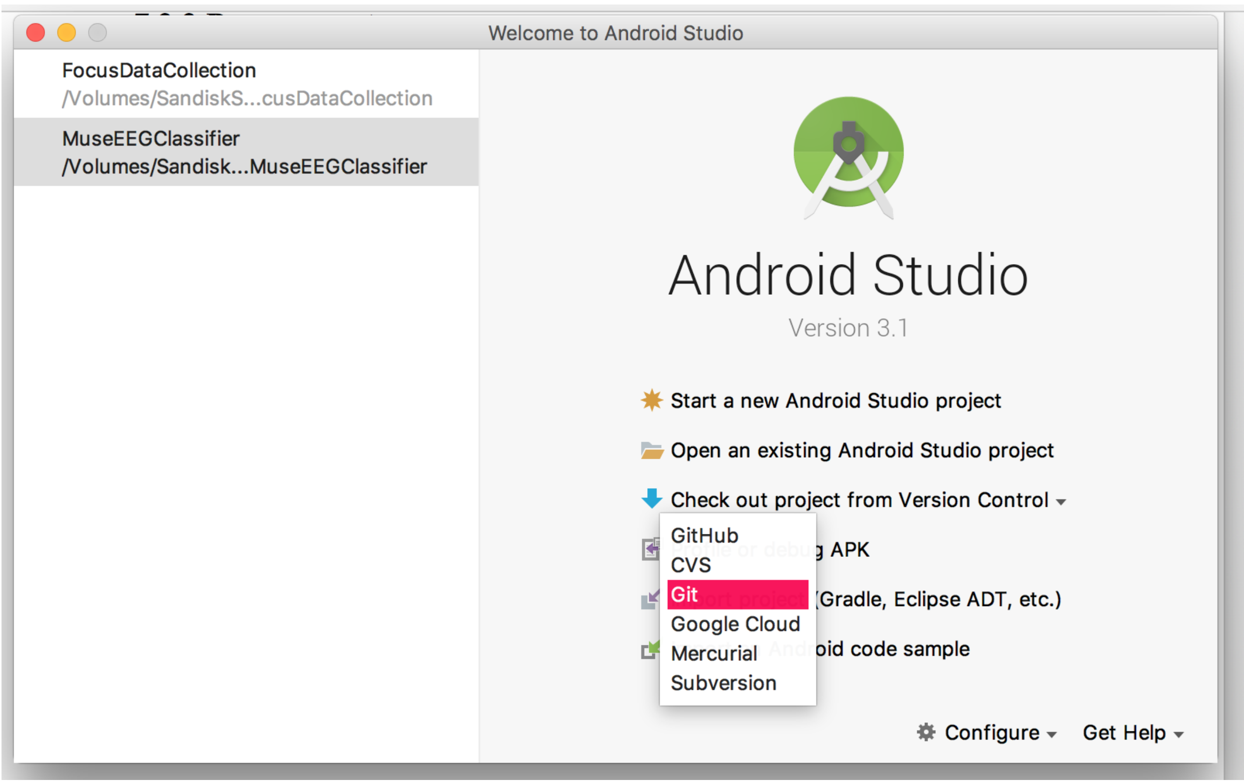


Figure 5-1: Android Studio git clone 1

Figure 5-1 shows where locate git clone using android studio.

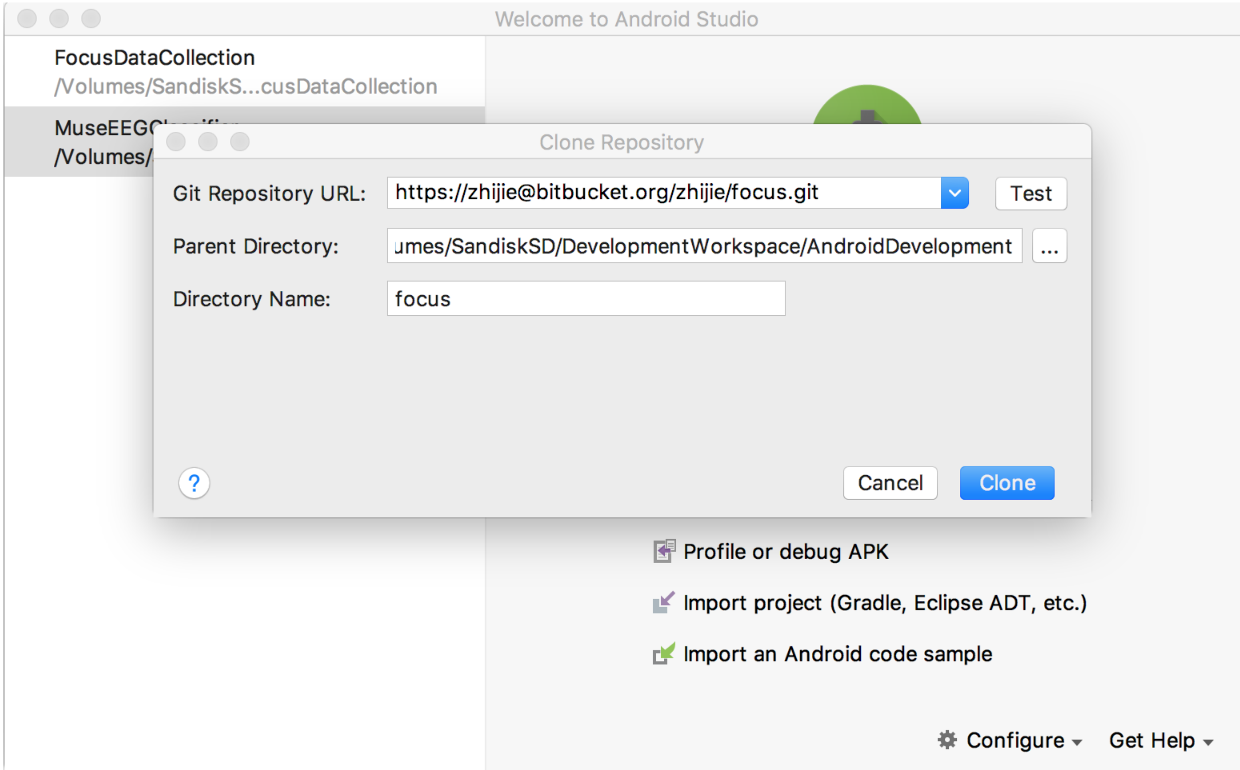


Figure 5-2: Android Studio git clone 2

Figure 5-2 shows how to perform git clone using android studio.

When the clone is completed, setup the android studio as default and install all required dependences. Once ready, you will be able to install on your android device.

5.3 Participants

The participants age ranges from 20 to 26 years old. All the participants are healthy male subjects who have no history of practicing meditation or any form of relaxation techniques.

5.4 Procedure

Each participant was briefed and given a walkthrough of the entire sequence of data collection allowing them to be familiarized with the systemic conduct of the experiment. To be familiarized, the participant was made to answer a few arithmetic questions. The participants are reminded before the start of the data collection to refrain from clenching their jaw muscle or moving any facial muscle to ensure that their EEG signal data are relatively clean from artifacts.

The participants were provided a Samsung Galaxy Tab S2 9.7 tablet, Muse headband and an earphone. They are guided through the pairing process of the EEG Data Collection application on the Android phone and Muse headband.

The EEG Data Acquisition will have several phrases. These phrases are to facilitate the collection of the different mental states of the participant. The following are the sequence of events.

Table 5-2: Timing of EEG Data Collection Sequences

Task	Time (Minutes)
Arithmetic Control Session	2
Guided Meditation	10
Arithmetic Test Session	2

Table 5-2 shows the time needed for each task. At the beginning and ending of each task, Muse annotations are added to facilitate the classification of Active and Mindfulness states.

5.5 Application Walkthrough

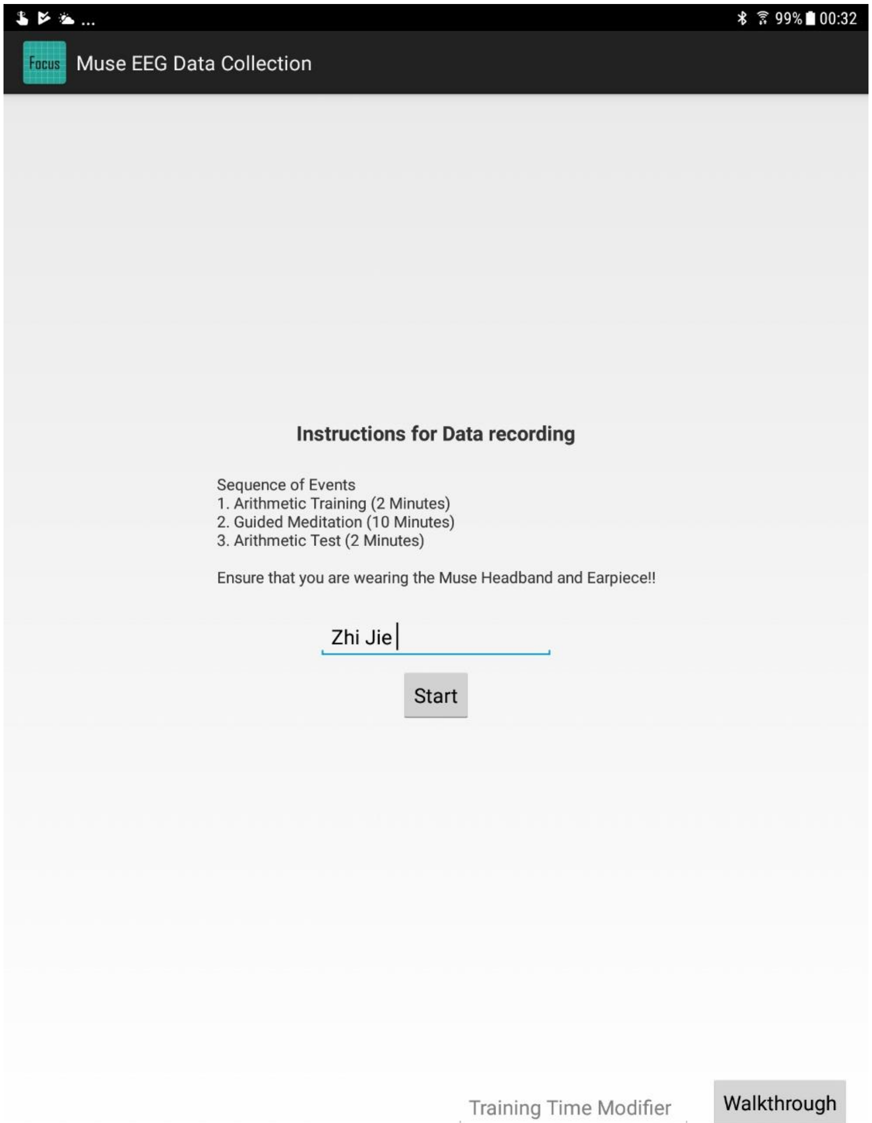


Figure 5-3: Landing page for EEG Data Collection

Figure 5-3 is the landing application screen when the application is first opened. It displays the sequence of events for the EEG Data Collection and reminds the participant to wear their Muse headband and earpiece. When the participant taps the start button, to the application will display the UI shown in Figure 5-4.

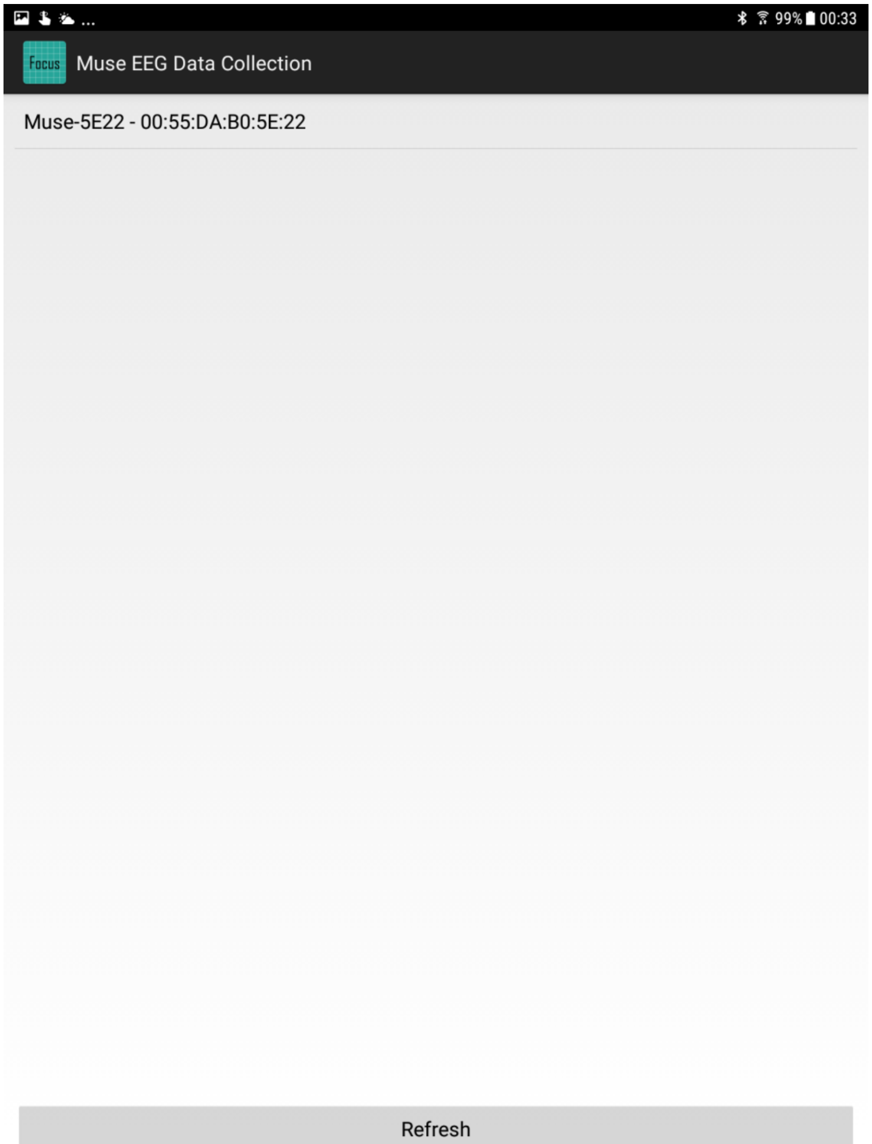


Figure 5-4: Select Muse Device

In Figure 5-4, the application displays a page to select Muse devices. The participant is to turn on their Muse headband and select the corresponding Muse device.

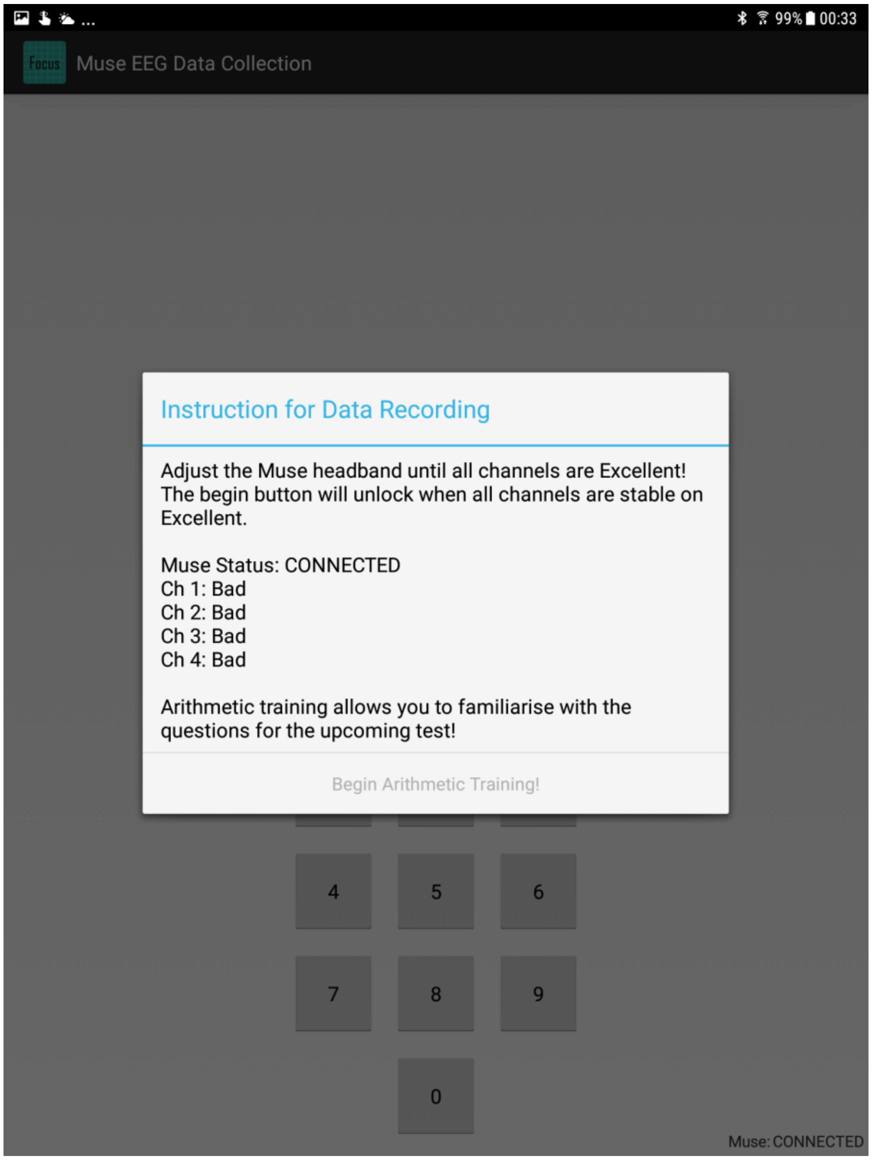


Figure 5-5: Muse headband connection 1

In Figure 5-5, the application displays the connection status and details about the channel quality. After the Muse device is selected, the Samsung Galaxy Tab S2 will connect to Muse device via Bluetooth. The “Begin Arithmetic Training” button will be disabled until Muse is connected and all four channels are stable on excellent quality.

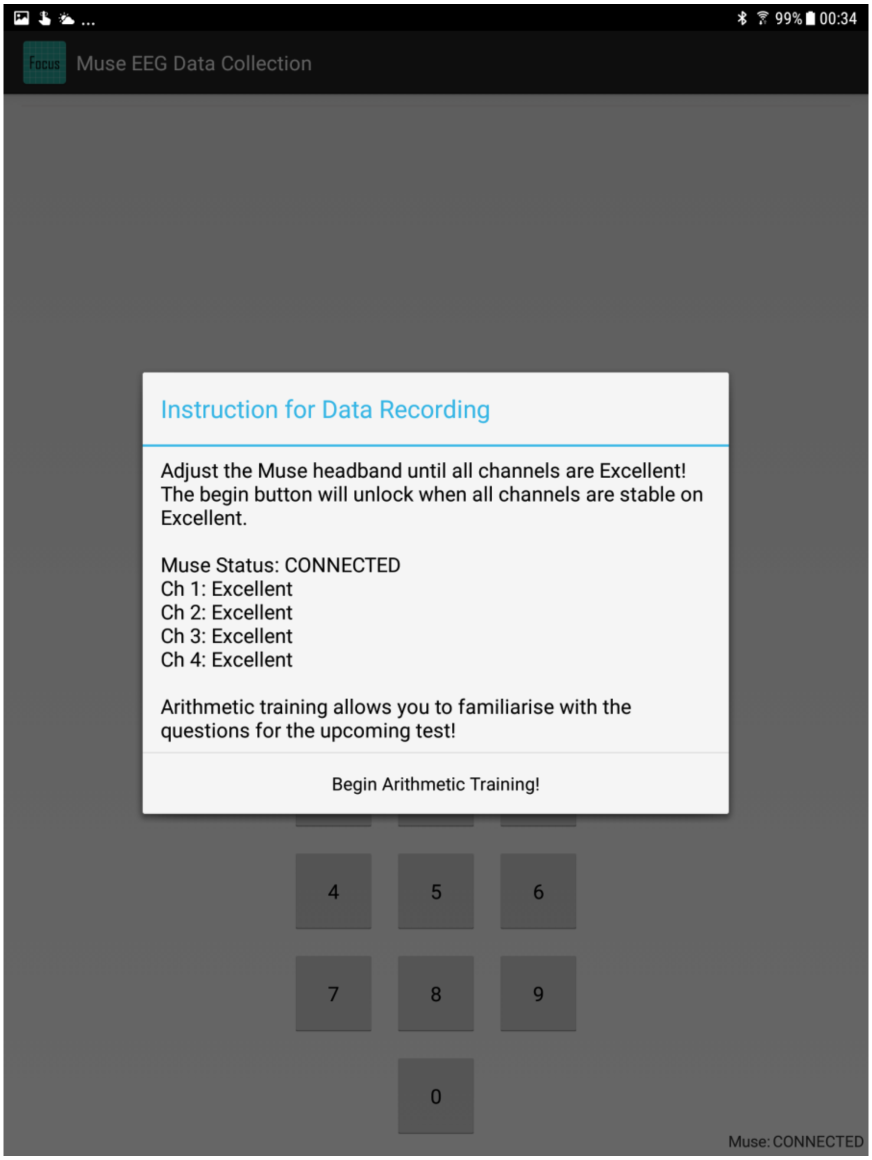


Figure 5-6: Muse headband connection 2

In Figure 5-6, “Begin Arithmetic Training” button will be enabled once all four channels quality are excellent for 3 seconds. The participant is to adjust their Muse headband until the all four channels are stable on excellent quality and tap “Begin Arithmetic Training” when ready.

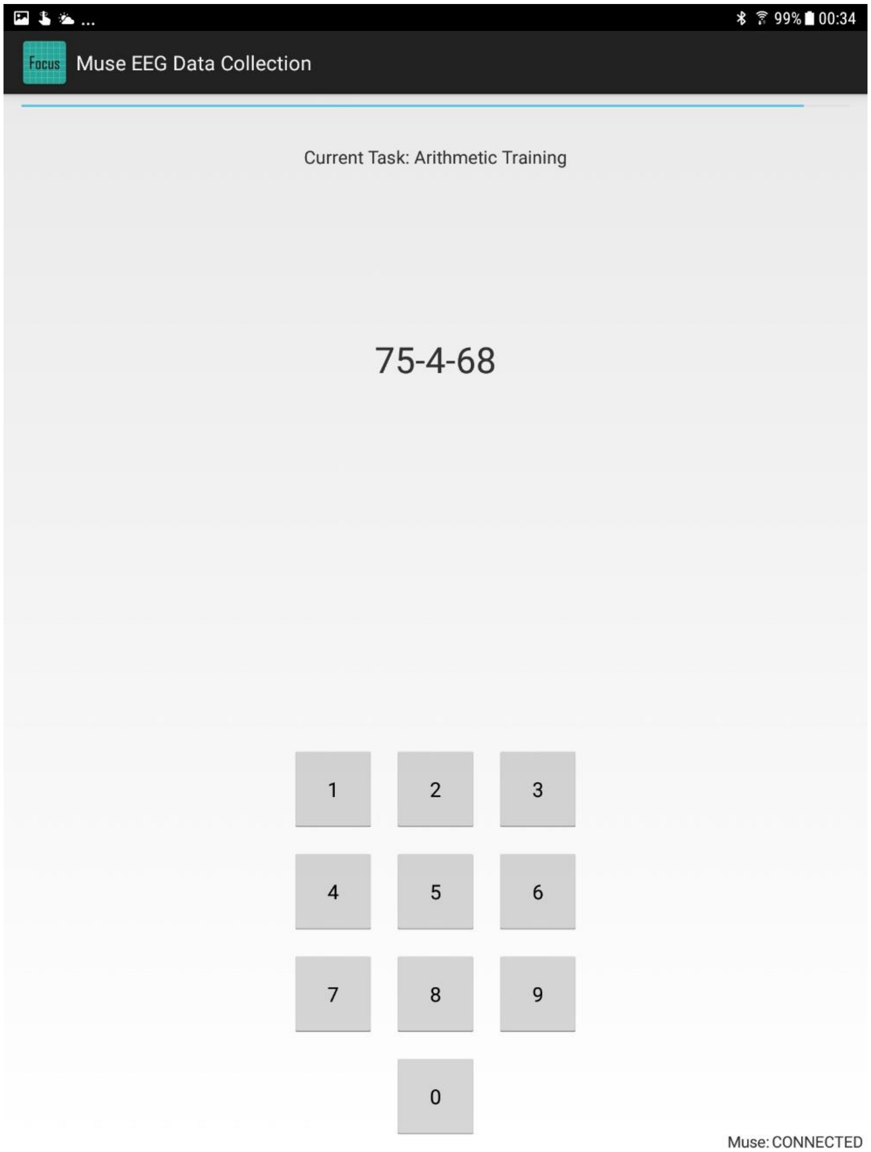


Figure 5-7: Arithmetic Training

In Figure 5-7, the application displays arithmetic training questions. There is a timer displayed as a blue line at the top of the application which lets the participant to know how much time is remaining. The mathematical expressions are displayed, and the participant are required to tap on the button with the correct answer. Participants are to answer arithmetic questions till the time runs out.

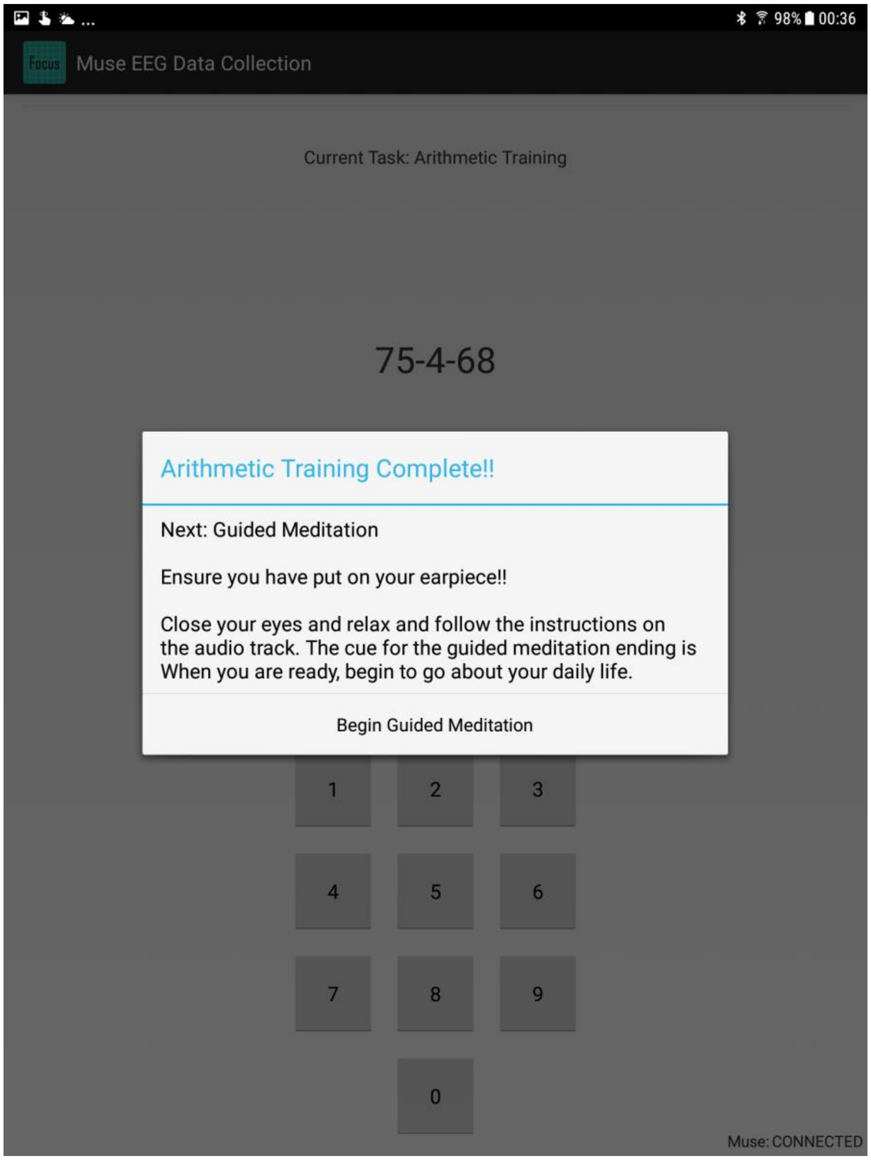


Figure 5-8: Arithmetic Training Complete

In Figure 5-8, the arithmetic training is complete. The application reminds the participant to put on the earpiece and displays instruction for guided meditation. It also instructs the participant on when the guided meditation will end.

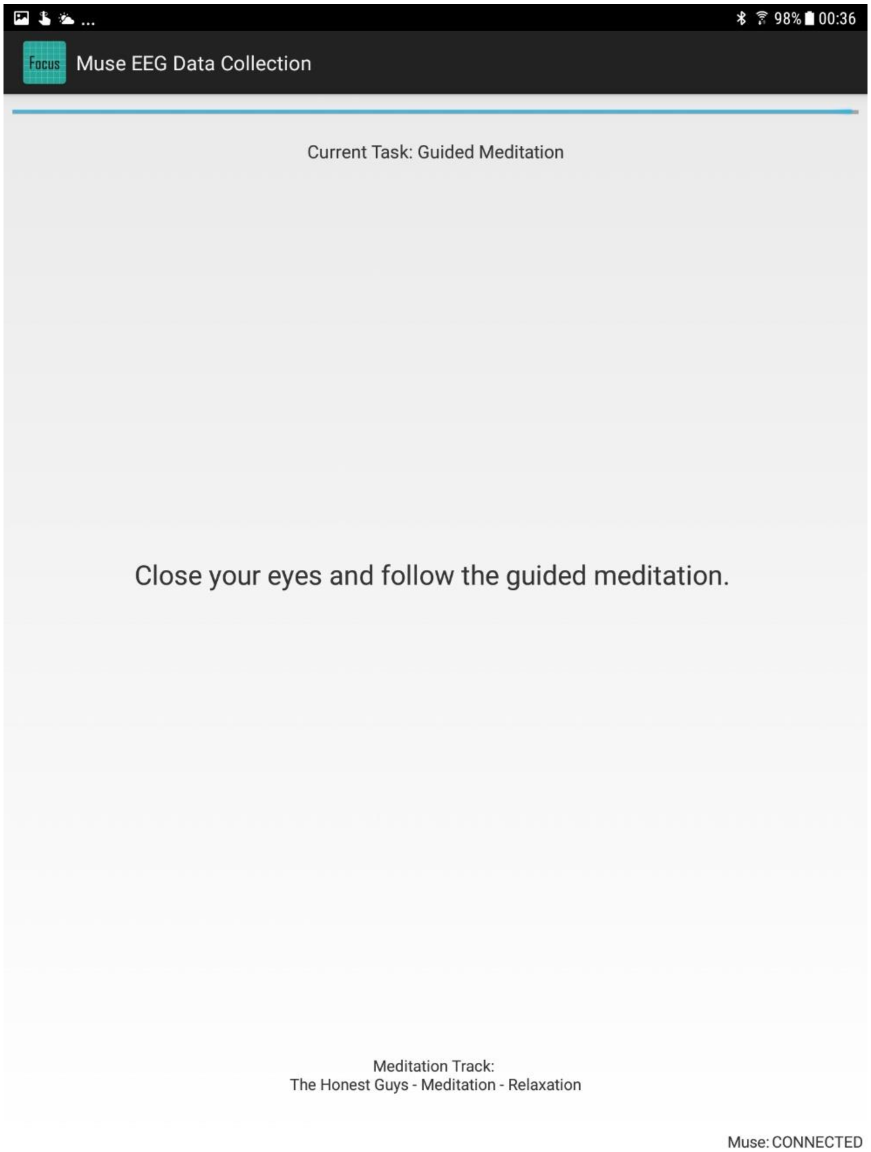


Figure 5-9: Guided Meditation

In Figure 5-9, the guided meditation track begins to play. The participant is once again reminded to close their eyes and follow the guided meditation. The guided meditation sound track which is played is displayed at the bottom of the screen.

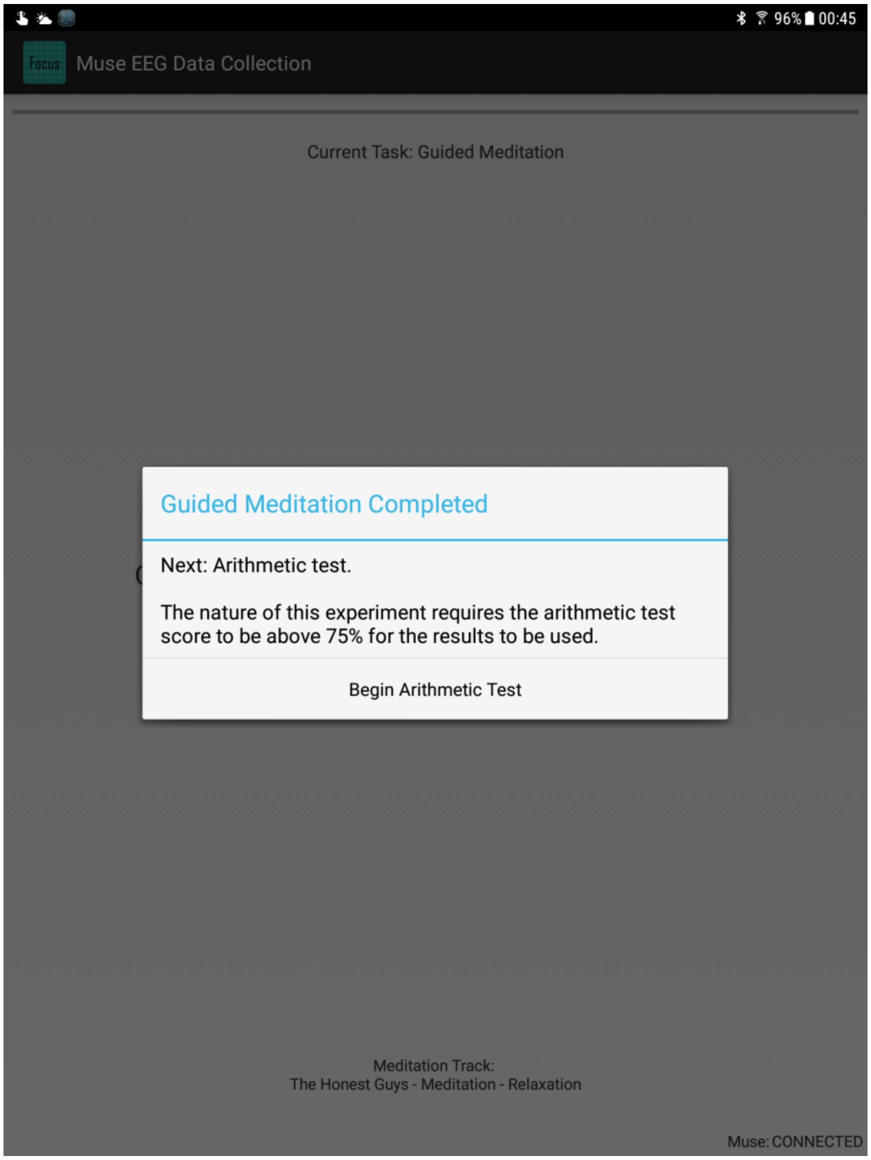


Figure 5-10: Guided Meditation completed

In Figure 5-10, the application displays the guided meditation is complete. The participant is reminded that for their score must exceed 80% for their data to be used in this study.

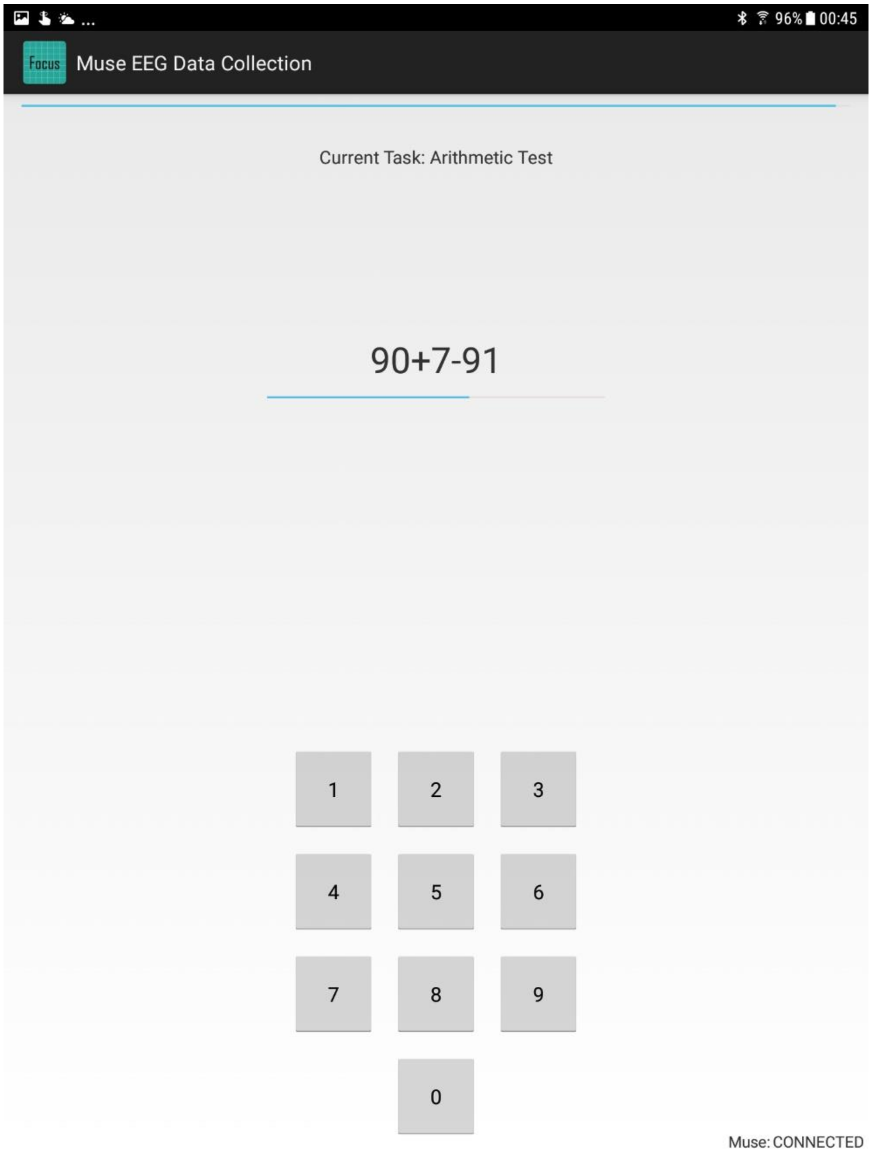


Figure 5-11: Arithmetic test

In Figure 5-11, the application displays the arithmetic test questions. There is an additional timer for each question. The participant must tap on the button with the correct answer before the timer ends.

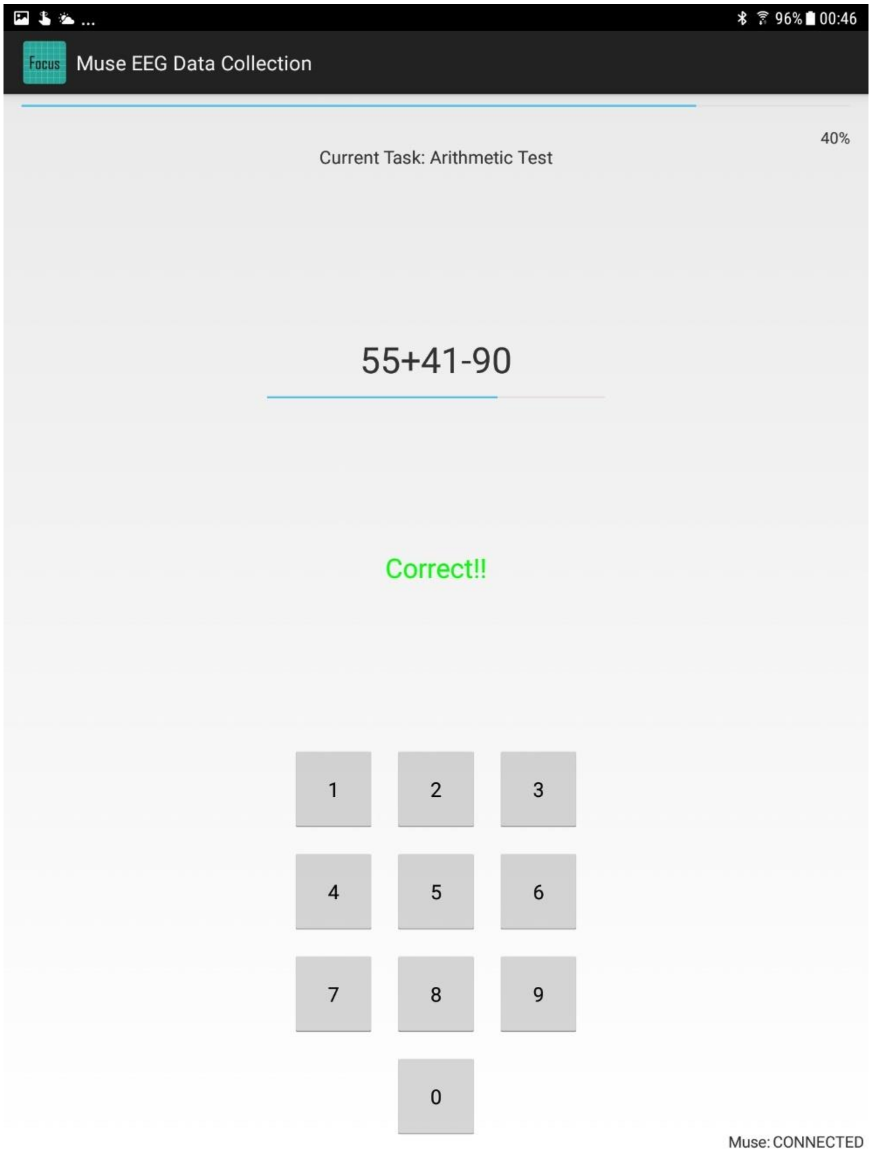


Figure 5-12: Correct Answer

In Figure 5-12, the application displays "Correct!!" in green under the question when the participant taps on the correct answer. A percentage of correct answers will be displayed at the top right-hand corner when the participant has answered at least one question.

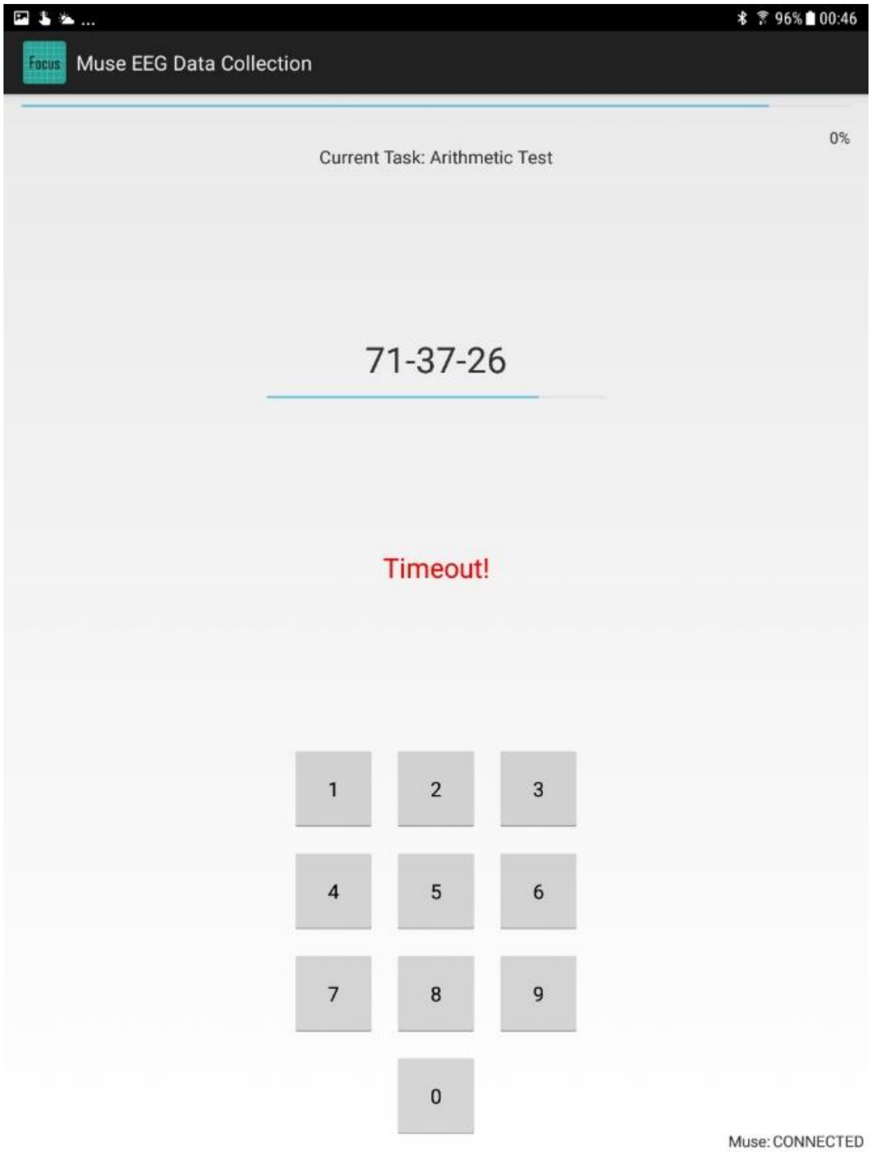


Figure 5-13: Question Timeout

In Figure 5-13, the application displays a "Timeout!" under the question as the participant has not managed to answer the question before the timer ends.

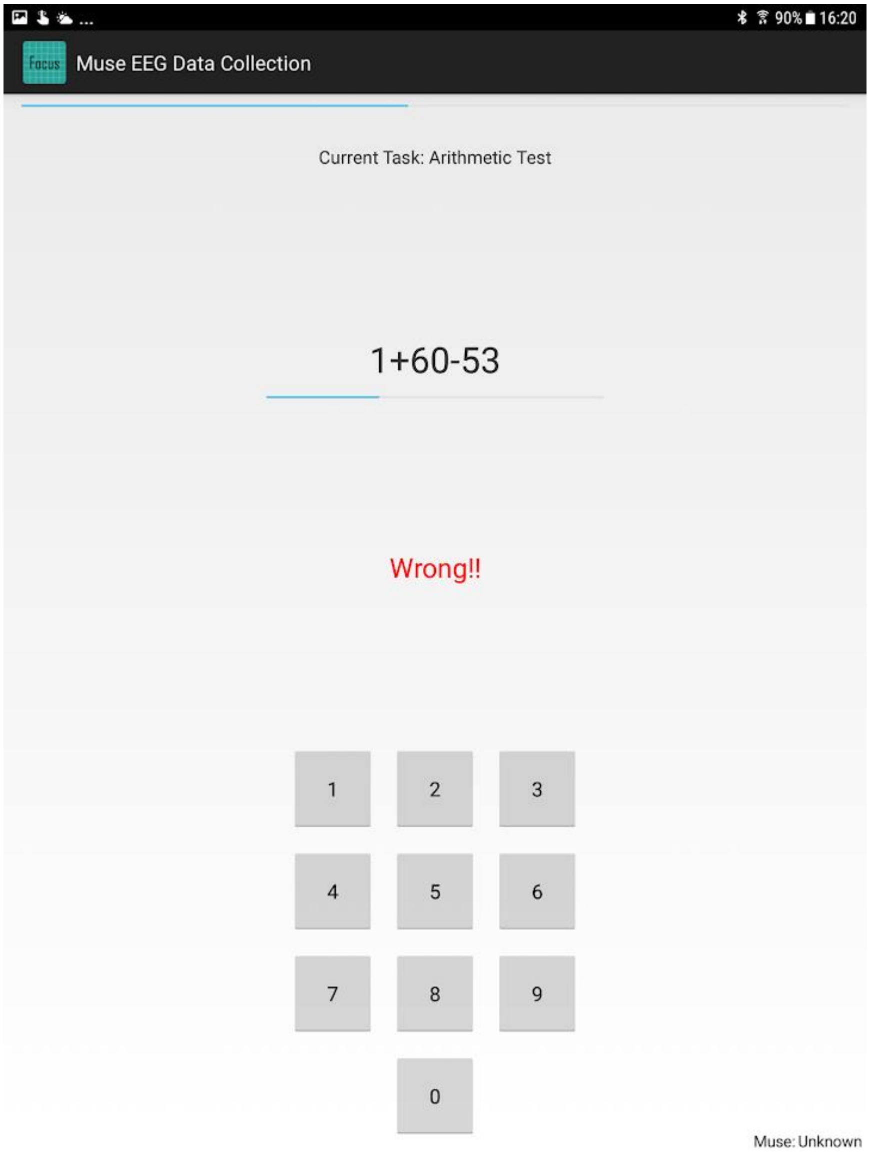


Figure 5-14: Arithmetic Test, Answered wrongly

In Figure 5-14, the application display “Wrong!!” in red under the question when the participant taps on the wrong answer. The arithmetic question remains unchanged until the question has been correctly answered or when the timer for each question ends.

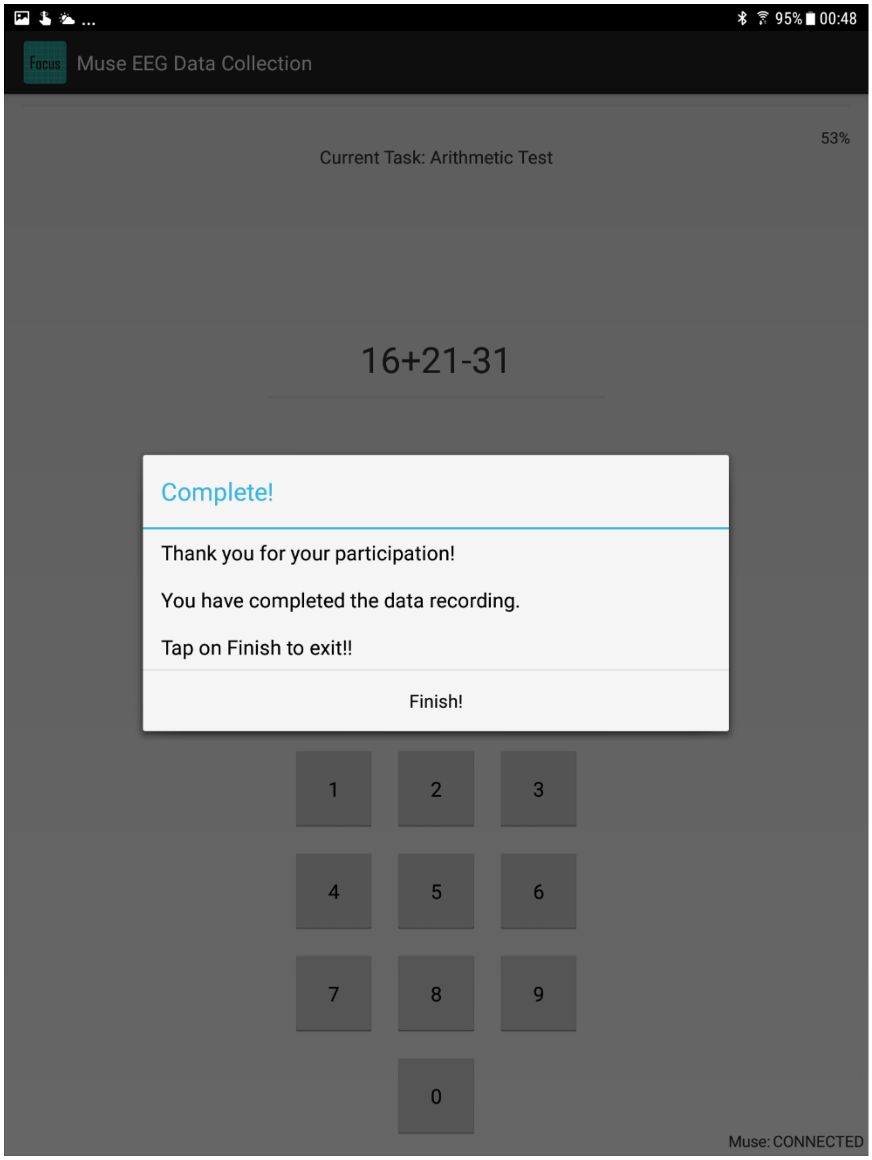


Figure 5-15: Data Collection Complete

In Figure 5-15, the application displays a thank you dialogue box when the arithmetic test is complete. The EEG signal data is saved on the Samsung Galaxy Tab S2 device under the folder "Internal storage/Android/data/com.zhijie.musefocus/files/Download/". The file is named with the following format ***"name_dd_mm_yyy_hh_mm_ss.muse".***

Chapter 6: EEG Data Processing and Model Generation

6.1 Overview

This chapter will discuss the conversion of the MUSE file into CSV file and the generation of the SVM model.

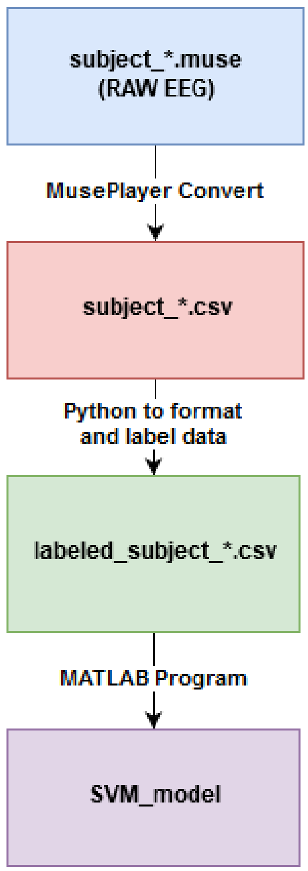


Figure 6-1: Raw EEG to SVM Model Overview

Figure 6-1 shows the overview of the process of generating a SVM model using raw EEG recorded using Muse. The EEG signal is recorded in a file format Muse. This format is propriety to Muse and is not easy to process. Hence, MusePlayer is used to convert the MUSE file to CSV file. The CSV file is then further processed removing unnecessary EEG signals and labeled with mindfulness and stress. The labeled CSV file is used to train and generate the SVM model.

6.2 Program Setup

6.2.1 Prerequisites

You need to clone/install:

- Muse Tools [20]
- MuseEEGClassifier [21]
- MATLAB
- MATLAB_SVM_Model_Generator [22]

The script to convert MUSE to CSV format is written and tested on MacOS High Serria. It should also work on Linux. Windows platform is not supported. The convert MUSE to CSV format python script can be found in the folder “MuseEEGClassifier/MuseToCSV/scripts/”.

6.2.2 Running the program

main_batch_convert

The first operation is to convert all *.Muse file in folder to output_folder/original_csv/*.csv

The second operation is to format the original_csv into labelled CSV that can be used for training in SVM. It formats files from output_folder/original_csv/*.csv to output_folder/*.csv .

```
./main_batch_convert.py -i <input folder> -o <output folder>
```

main_muse_to_csv

Convert a *.muse file into CSV, then format and label the CSV file.

```
./main_muse_to_csv.py -i <inputfile> -o <outputfile>
```

format_csv.py

Format *.CSV file removing redundant EEG data and append labels for meditation and stess.

```
./format_csv.py -i <inputfile> -o <outputfile>
```

Figure 6-2: Running the python script

Figure 6-2 shows the various commands that the script can be used.

6.3 Procedure

6.3.1 Converting “MUSE” to “CSV”

The recorded EEG signal data can be found in the Android Tablet in the following location. ‘‘Internal storage/Android/data/com.zhijie.musefocus/files/Download/”. Once the MUSE file has been retrieved from the Android tablet, the script “main_batch_convert.py” can be used to convert the MUSE file to CSV file.

Clone the MuseEEGClassifier to get the python scripts. Then, copy and paste all the muse files in the folder “novice_muse_recording” under “MuseEEGClassifier/MuseToCSV/scripts/”. The following command is used to convert all the MUSE file in the folder “novice_muse_recording” and output the formatted CSV file to “novice_muse_recording_csv”.

```
./scripts/main_batch_convert.py -i novice_muse_recording -o
novice_muse_recording_csv/
```

Figure 6-3: Bash code to batch convert folder

Figure 6-3 shows the bash code to batch convert all the Muse format file into labelled CSV files which are ready to be used for SVM model training.

```
Ohs-MacBook-Pro:MuseToCSV zhijie$ ./scripts/main_batch_convert.py -i novice_muse_recording -o novice_muse_recording_csv/
Input folder is novice_muse_recording
Output folder is novice_muse_recording_csv/

Muse Player 1.8.4
Input:
  * Muse file(s): ['novice_muse_recording/eug_28_02_2018_13_29_30.muse']

Output:
  * CSV file: novice_muse_recording_csv//original_csv/eug_28_02_2018_13_29_30.csv
Playback Time: 903.1s : Sending Data
 '{"annotation_type": "recorder_info"
 'arithmetic_training_begin'
 'arithmetic_training_ended'
 'guided_meditation_begin'
 'guided_meditation_ended'
 'arithmetic_test_begin'
 'arithmetic_test_ended'
 'Disconnected'
Operation successfully completed!
Completed 1 of 6


Muse Player 1.8.4
Input:
  * Muse file(s): ['novice_muse_recording/Yijie_02_03_2018_14_42_14.muse']

Output:
  * CSV file: novice_muse_recording_csv//original_csv/Yijie_02_03_2018_14_42_14.csv
Playback Time: 88.7s : Sending Data
```

Figure 6-4: Convert MUSE to CSV

Figure 6-4 displays the terminal output when each MUSE file has been processed and formatted into CSV file. The program will check in the output folder if the file generated file exists. If the file exists, the program will skip and start converting the next Muse file.

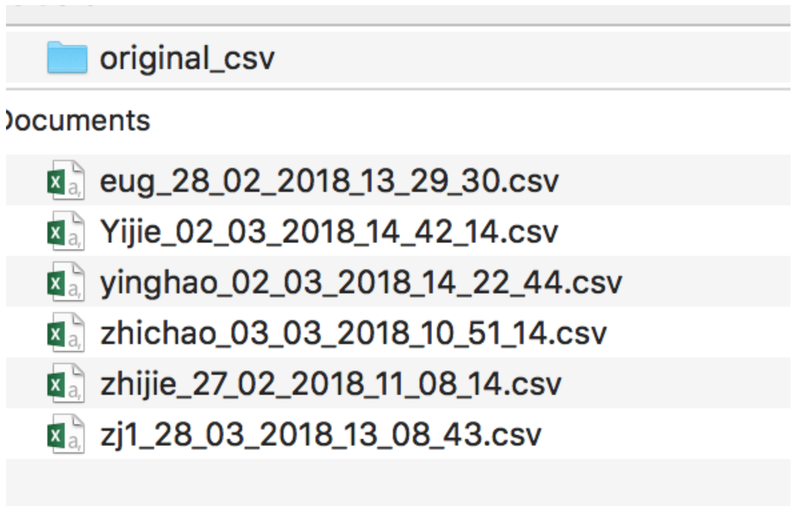


Figure 6-5: Files generated

Figure 6-5 shows the files generated when the script is executed. The folder "novice_muse_recording_csv/original_csv" contains the unformatted CSV files directly converted from the Muse format and the output folder "novice_muse_recording_csv" contains the formatted and labelled file which can be used in MATLAB.

6.3.2 Generate SVM Model using MATLAB

First, clone MATLAB_SVM_Model_Generator into the MATLAB working directory. Next, to obtain the SVM model, place the formatted CSV files into the **data** folder and run the file “genFullModelMain.m”. A model file “svm_model.txt” will be generated with the steps as mentioned in EEG Signal Processing earlier on in the chapters. The classification model can be now loaded into LibSVM in Java.

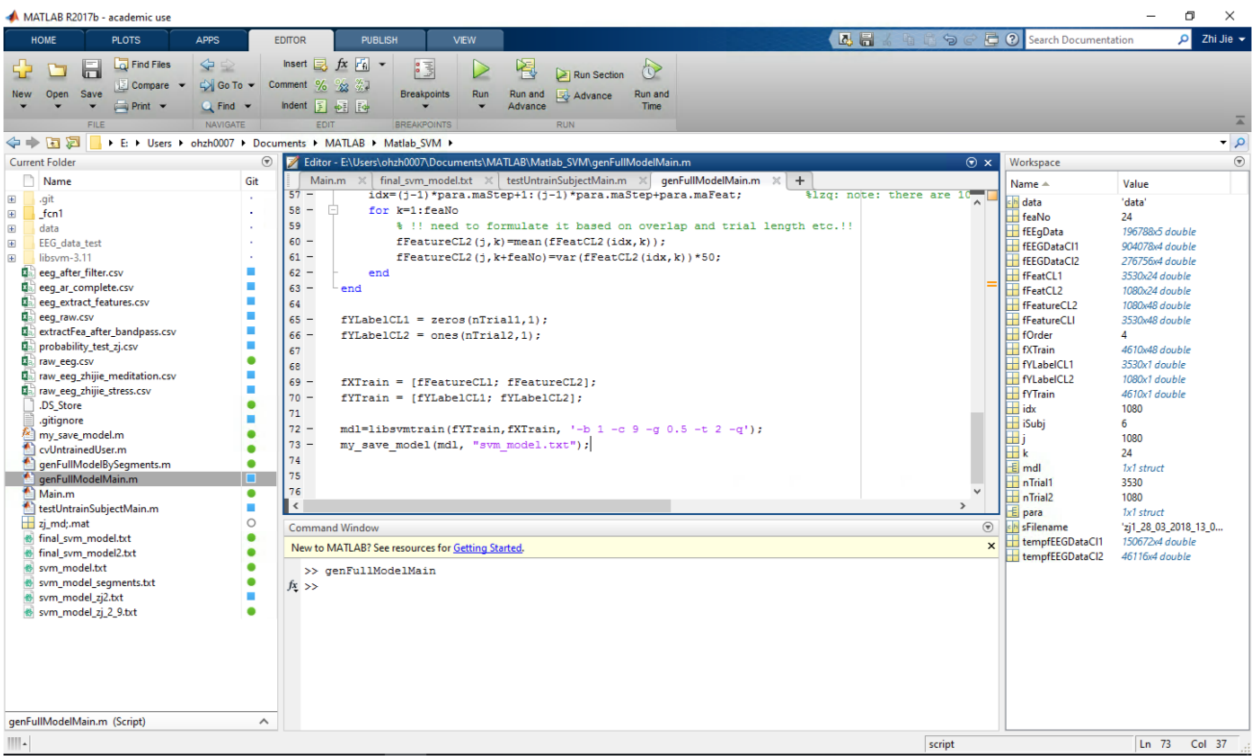


Figure 6-6: MATLAB generate classification model

In Figure 6-6, shows the state MATLAB after running the main program.

Chapter 7: **Real-time EEG Classifier**

7.1 Overview

The original goal for this project was to create an Android application which will be able to give real-time feedback for meditation. However, due to the lack of time this project has been halted. The current application can display the real-time state and history of the user state via a line graph.

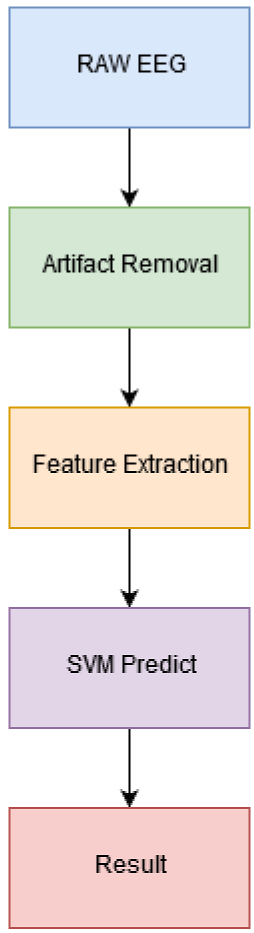


Figure 7-1:Overview on real-time EEG prediction

Figure 7-1 shows the overview on how the prediction is generated. The raw EEG received is first split into windows of samples. These windows are then processed using the artifact removal function after which the features are extracted. The extracted features are passed into SVM predict and SVM will return the classification result.

7.2 Program Setup

7.2.1 Prerequisite

You need to clone or install

- MuseEEGClassifier [21]

7.2.2 Program setup

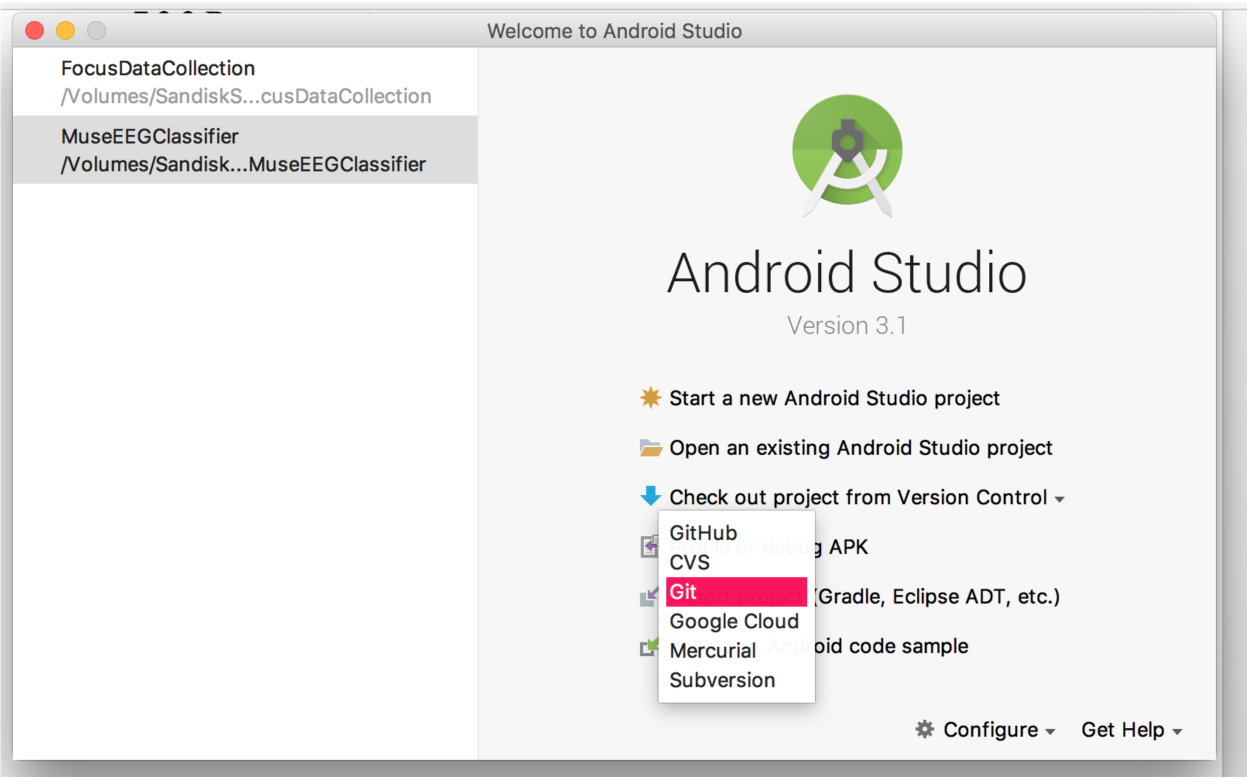


Figure 7-2: Android Studio git clone 1

Figure 7-2 shows where to perform git clone using android studio.

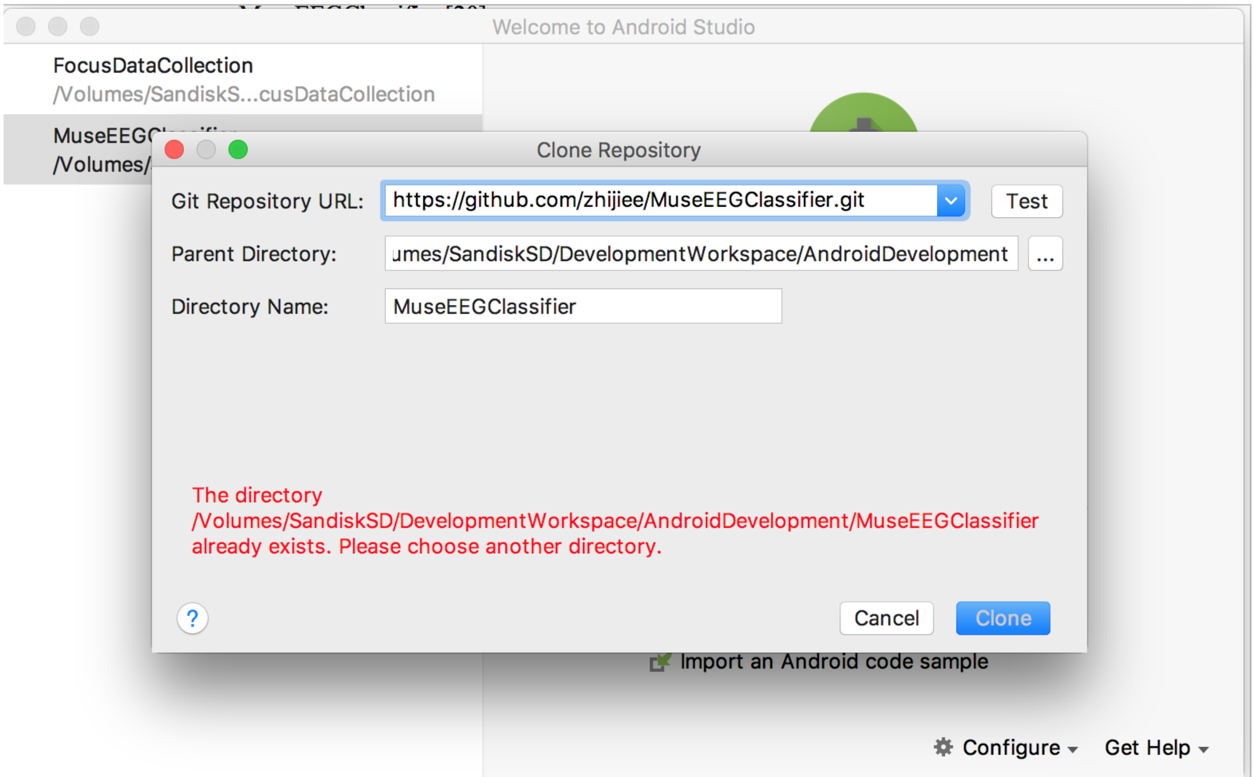


Figure 7-3: Android Studio git clone 2

Figure 7-3 shows how to perform git clone using android studio.

When the clone is completed, setup the android studio as default and install all required dependences. Once ready, you will be able to install on your android device.

7.3 Application

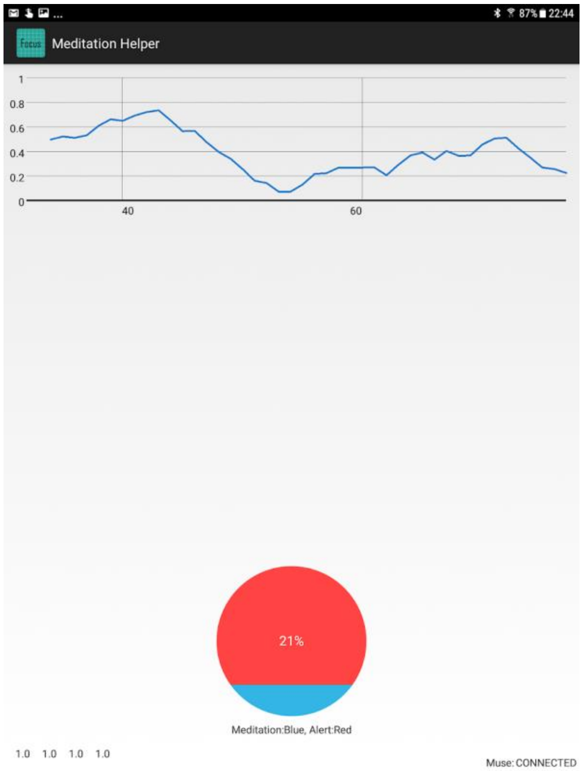


Figure 7-4: Real-time classification application

Figure 7-4 shows the current state via the circle progress bar and the past state via a line graph. The prototype application displays the mindfulness level of the user using the circle progress bar [23] . When the bar is blue, it is predicting the user being in Mindfulness state and when the bar is red, it is predicting that the user is in active state. It also displays a history of the last 60 seconds prediction using the line graph [24] as shown above.

Additional details such as the Muse signal quality indicator and the Muse connection status are also included in the bottom of the screen.

Chapter 8: **Results and Discussion**

8.1 Overview

In the results and discussion section, there are two main sub-topics. Chapter 8.2 describes SVM training which is performed on an individual data and then used for prediction of another part of the individual data using a technique called k-fold cross validation.

In Chapter 8.3 describes SVM training which is performed on four individuals, the SVM model generated is used to predict another individual who is not included in the training.

Table 8-1: SVM predict score example

Classification	Mindfulness	Stress
Stress	30%	70%
Mindfulness	70%	30%

Table 8-1 shows the results SVM predict. The SVM predict returns a classification (stress/mindfulness), and a score in percentage on the state classification. For example, the classification will return stress when the stress score value is higher than the mindfulness score. The sum of the two states should be 100%.

8.2 10-fold cross validation (Individual)

In this section, 10-fold cross validation is performed on each individual data. 9 segments of Mindfulness and Stress data from an individual are used as training set for the SVM model and the remaining segment of Mindfulness and Stress data are used to test SVM model. The segments are rotated such that each segment is used as validation data at least once.

The results generated are spilt into 2 sets. The first and second set shows the confidence level of the SVM model in predicting the Mindfulness state and Stress state respectively.

8.2.1 Mindfulness State Confidence

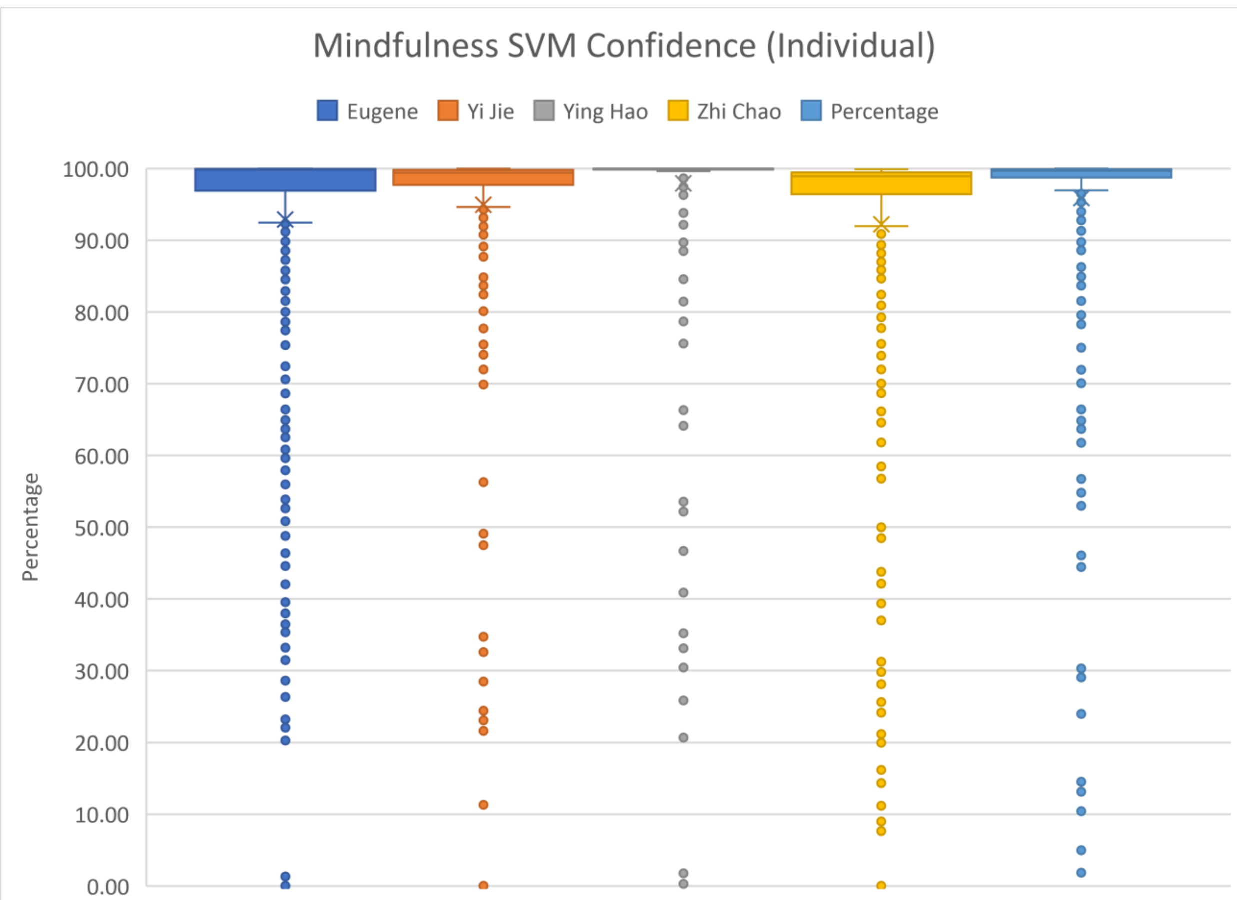


Figure 8-1: Mindfulness 10-fold Cross Validation (Individual)

Figure 8-1 is generated using only Mindfulness as input data. The y-axis shows the degree of confidence in which the SVM model predicts that it is of Mindfulness state. The dots below the box and whiskers are considered outliers. The results are very compact towards the top. This shows that the SVM model is very confident that it is correct when predicting the mindfulness state.

Table 8-2: Mindfulness SVM Confidence (Individual)

	Eugene	Yi Jie	Ying Hao	Zhi Chao	Zhi Jie	Average
Mean	92.77%	94.93%	**97.82%**	**92.21%**	95.78%	**94.70%**
Quartile 1	96.99%	97.70%	**99.84%**	**96.39%**	98.73%	**97.93%**
Median	99.83%	99.41%	**99.96%**	**98.90%**	99.65%	**99.55%**
Quartile 3	99.93%	99.80%	**99.99%**	**99.45%**	99.90%	**99.81%**

Table 8-2 shows the values of the different quartiles and the mean value of overall SVM's confidence on meditation data. The data from the table shows that even on the 1st quartile, it is 97.93% confident in its prediction of the state being mindfulness.

8.2.2 Stress state Confidence

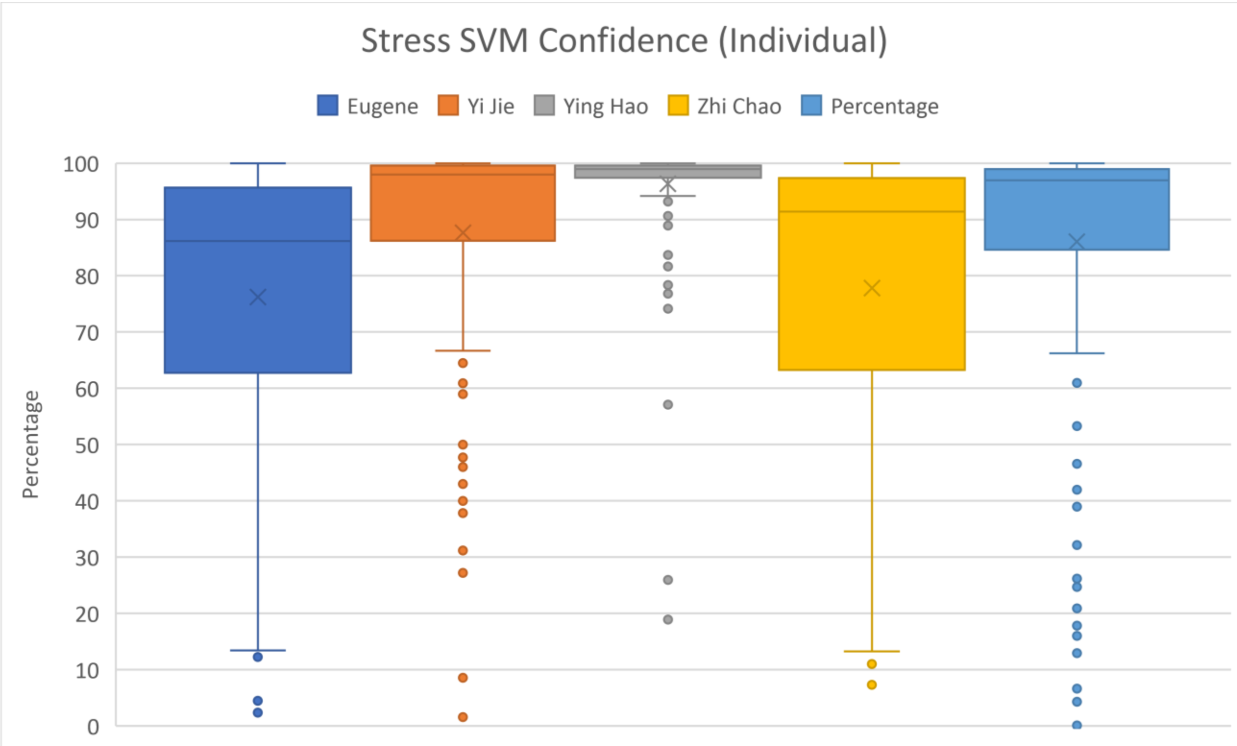


Figure 8-2: Stress 10-fold Cross Validation (Individual)

Figure 8-2 is generated using on Stress as input data. The y-axis display confidence level of SVM model predicting the input data is of stress state. The dots below the box and whiskers are considered outliers. The results are fairly compact between the range of 60% to 100%. This shows SVM although not as confident in predicting stress it is still able to high confidence in its prediction.

Table 8-3: Stress SVM Confidence (Individual)

	Eugene	Yi Jie	Ying Hao	Zhi Chao	Zhi Jie	Average
Mean	**76.24%**	87.65%	**96.32%**	77.79%	87.16%	**85.03%**
Quartile 1	**63.52%**	86.49%	**97.44%**	63.79%	84.70%	**79.19%**
Median	**86.17%**	97.99%	**98.97%**	91.39%	96.20%	**94.14%**
Quartile 3	**95.59%**	99.58%	**99.59%**	97.37%	98.94%	**98.21%**

Table 8-3 shows the values of the different quartiles and the mean value of overall SVM's confidence on stress data. The result from the table show that on quartile 1, SVM is on average 79.19% confident in its prediction on the stress state.

8.2.3 SVM Prediction accuracy

Table 8-4: Prediction accuracy (Individual)

Name	Accuracy
Eugene	92.00%
Yi Jie	**96.00%**
Ying Hao	92.00%
Zhi Chao	93.33%
Zhi Jie	**88.00%**
Average Accuracy	**92.27%**

Table 8-4 show the accuracy of SVM prediction per user. The result from the table shows the accuracy based on the number of predictions it gets correct using with the label predict set using the k-fold cross validation. Overall, the average prediction accuracy is 92.27%

8.2.4 Discussion on individual 10-fold cross validation

From the above results, SVM is capable of predicting mindfulness state with the average median of 99.55%. Stress prediction is also excellent with its median at 94.14%.

SVM model trained with the subject data can accurately predict the correct state of the subject himself. This is shown with the overall accuracy of 92.27%.

8.3 Untrained individual Validation

In this section, the SVM model are trained using k-1 individuals. The remaining untrained participant is used as validation data. This validation is performed for each of the participant. This validation is to simulate real-world scenarios whereby the subject using the model will not have their data included in the training of the model entirely.

8.3.1 Mindfulness Confidence (Untrained Individual)

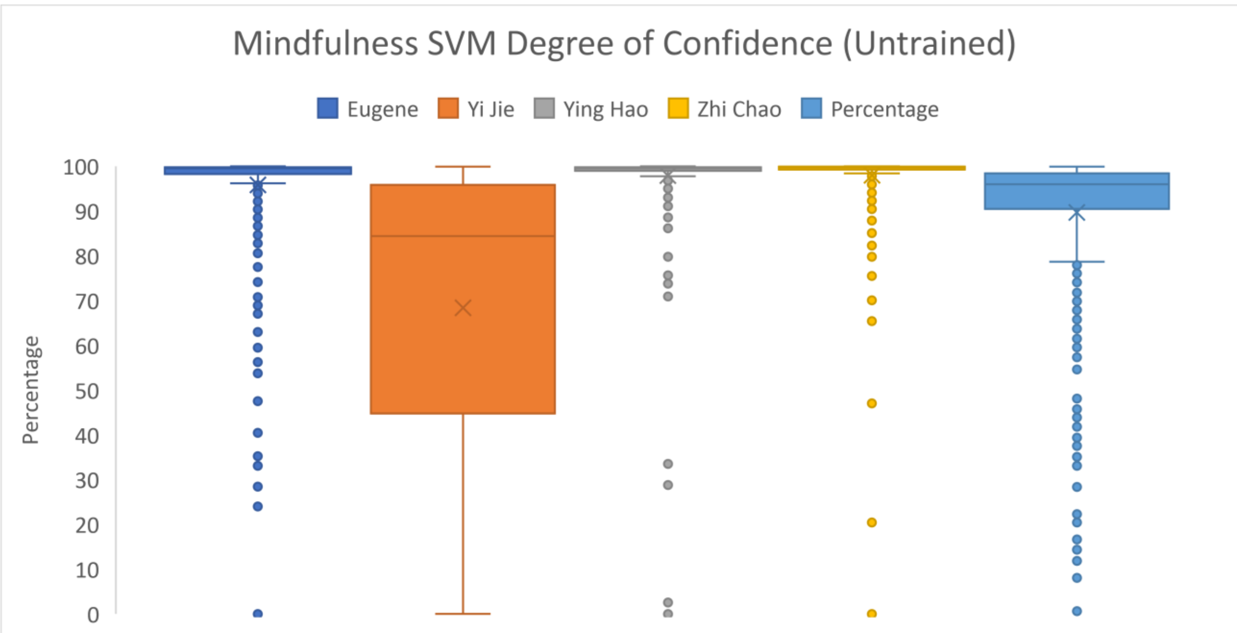


Figure 8-3: Mindfulness SVM Degree of Confidence (Untrained)

Figure 8-3 is generated with mindfulness data as input for each participant. Y axis shows the confidence level of each point. Other than participant "Yi Jie" the rest of the participant have high confidence of SVM recognizing the correct mindfulness data.

Table 8-5: Mindfulness results (untrained)

	Eugene	Yi Jie	Ying Hao	Zhi Chao	Zhi Jie	Average
Mean	96.13%	**68.38%**	98.09%	**98.30%**	89.64%	**90.11%**
Quartile 1	98.34%	**44.77%**	98.97%	**99.29%**	90.48%	**86.37%**
Median	99.57%	**84.73%**	99.56%	**99.79%**	95.98%	**95.93%**
Quartile 3	99.78%	**95.85%**	99.79%	**99.91%**	98.41%	**98.75%**

Table 8-5 shows the value of the different quartile of the mindfulness data. All the subject has prediction accuracy apart from "Yi Jie" having 20% less for mean as compared to the 2nd lowest.

8.3.2 Stress Confidence (Untrained Individual)

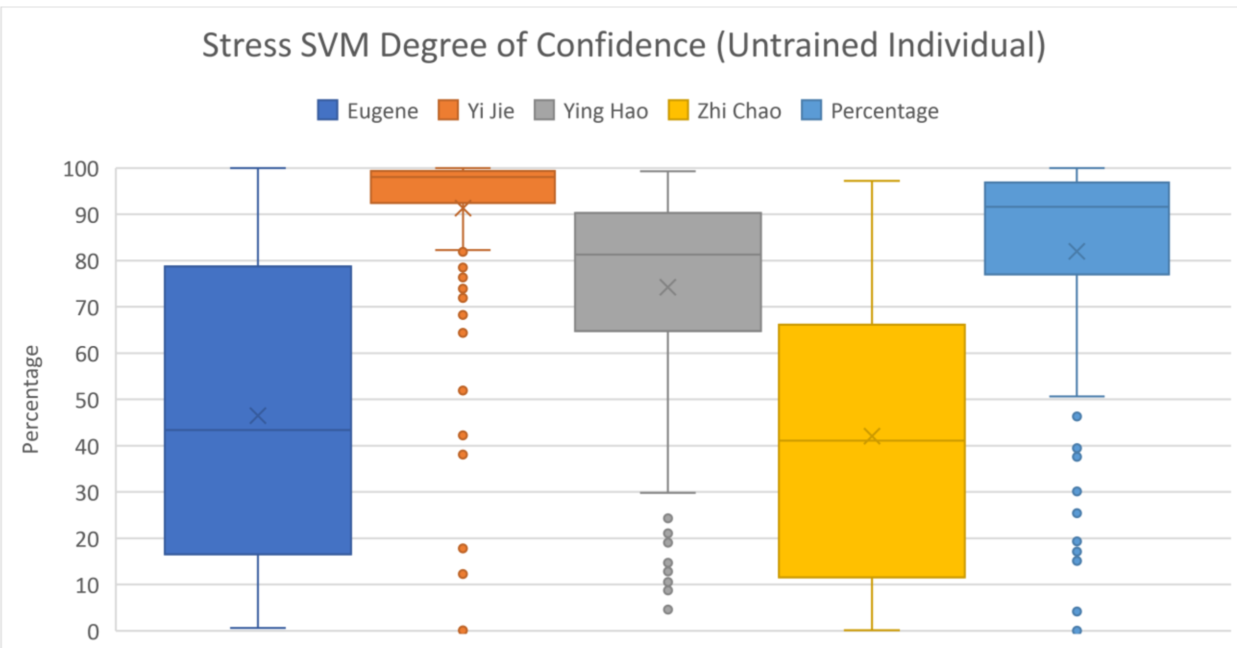


Figure 8-4: Stress SVM Degree of Confidence (Untrained Individual)

Figure 8-4 is generated with stress data as input for each individual. The y-axis shows the percentage which SVM was certain that the input data is stress. The results for each subject are quite diverse with some having low certainly and some have very high certainty.

Table 8-6: Stress results (untrained)

	Eugene	Yi Jie	Ying Hao	Zhi Chao	Zhi Jie	Average
Mean	46.47%	**91.32%**	74.23%	**42.06%**	81.96%	**67.21%**
Quartile 1	16.61%	**92.59%**	65.28%	**11.82%**	77.33%	**52.73%**
Median	43.40%	**98.06%**	81.28%	**41.09%**	91.59%	**71.09%**
Quartile 3	77.86%	**99.34%**	90.26%	**65.75%**	96.83%	**86.01%**

Table 8-6 shows the value of the different quartile of the stress data. It shows an average of median of 71.09%.

8.3.3 Overall Result

Table 8-7: Overall Accuracy (untrained)

Name	Accuracy
Eugene	85.7516
Yi Jie	77.8357
Ying Hao	95.8225
Zhi Chao	86.4052
Zhi Jie	93.4811
Average Accuracy	87.86

Table 8-7 shows the overall accuracy of prediction per subject when the subject is not part of the dataset.

8.3.4 Discussion

SVM model training without the subject results are not as good when compared to training with the subject's own data. This is to be expected as each individual data is slightly different. The average overall accuracy of the of SVM prediction when trained without subject's data is at 87.86% which means that implementation of an accurate real-time EEG is very possible. As of now the training is done with 4 subjects and the untrained subject is used to test for prediction. As the number of subject used to train increase, the overall accuracy is expected to increase.

However, it must be noted that when not trained with the subject data, the stress prediction is unreliable with an average of 71.09%. Some predictions are almost completely unreliable however this may be mitigated if the number of subject use to train is increased.

Overall, the results show that it is promising, and creating an accurate real-time meditation helper is highly possible.

8.4 Challenges faces

8.4.1 MATLAB SVM model to JAVA LibSVM

One of the unexpected challenges faced during the development of this project was that MATLAB SVM model could not be easily exported. There is no in-built function to export the SVM model as a text file. Hence, an additional script was written following the format of how Java LibSVM loads the model. However, this is not a complete fix as some of the details are hardcoded. As such, if different parameters are used for SVM training, the script to save the SVM model must be edited.

8.4.2 Real-time EEG processing

The EEG artifact removal in MATLAB is processed with the entire data sample unlike on the Android device which feeds the artifact removal every second. Due to the fact that the artifact removal is implemented using moving average, the end results of each window being processed with artifact removal ends up with 11 zeros at the end of the processing. This has greatly impacted the results negatively during testing.

The current solution used is to use 3 raw EEG signal sample windows with the middle window being the data that is supposed to be processed with artifact removal. This resolves the issue having zeros before being processed for features. However, this solution comes with a delay of one second.

8.4.3 Setting up android studio with Muse library

Setting up the android studio project with Muse library ran into some unexpected issues. The project is setup as per the instructions from the Muse setup guide, however, the library is unable to detected by the Android Studio. The error message generated when using the Muse library is vague and no known solutions was found.

To resolve this issue, the Muse sample project was downloaded, and the project parameters changed to fit this project's requirement.

8.4.4 SVM Active states model tainted by artifacts

The difference between meditation and doing the arithmetic test is that meditation is completely still while performing arithmetic test, the participants will undeniably shift their eyes and blink. These actions create artifacts and will cause taint the SVM model. The consequence in such artifacts causes general movement of any facial muscle to tilt the scale towards the active state.

Chapter 9: **Conclusion and Recommendations for Future Work**

9.1 Conclusion

To reiterate, the purpose of this project is to build a brain computer interface (BCI) application for mindfulness training. Although the purpose of this project was not achieved entirely, this project has sufficiently demonstrated even with limited subjects that accurate real-time prediction of mindfulness state is highly possible. This project has also lay down the ground work mindfulness feedback application with the real-time classification of mindfulness and stress state.

9.2 Recommendations for Future work

9.2.1 Multi-class SVM

Multi-class SVM model should be explored to allow for multiple states detection. States such as non-mindfulness relaxation can be included to further increase the correct detection of mindfulness.

Furthermore, emotions such as anxiety, depression and fatigue can be included. These additional states allow for more customized feedback for the participant which will better serve the participant in their quest for deeper mindfulness.

9.2.2 Increase the number of dataset of SVM training

Due to limited time of the project, only a limited number of dataset was recorded and used for the SVM model generation. Even though the results are quite positive, the increase of dataset should directly lead to the increase in accuracy of detecting stress and mindfulness.

References

[1] R. A. Baer, "Mindfulness Training as a Clinical Intervention: A Conceptual and Empirical Review," *Clinical Psychology: Science and Practice*, vol. 10, no. 2, pp. 125-143, 2003.

[2] H. A. Slagter, A. Lutz, L. L. Greischar, A. D. Francis, S. Nieuwenhuis, J. M. Davis and R. J. Davidson, "Mental Training Affects Distribution of Limited Brain Resources," *PLoS Biology,* vol. 5, no. 6, p. e138, 2007.

[3] P. Grossman, L. Niemann, S. Schmidt and H. Walach, "Mindfulness-based stress reduction and health benefits. A meta-analysis," *J Psychosom Res*, vol. 57, no. 1, pp. 35-43, 2004.

[4] "Muse: the brain sensing headband," [Online]. Available: http://www.choosemuse.com/. [Accessed 10 Janurary 2018].

[5] P. Mahler, "The Complete Pocket Guide to EEG - iMotions," iMotions, [Online]. Available: https://imotions.com/eeg-guide-ebook/. [Accessed 22 March 2018].

[6] E. Alpaydin, Introduction to Machine Learning, MIT Press, 2014.

[7] C. C. Chih and J. L. Chih, "LIBSVM -- A Library for Support Vector Machines," [Online]. Available: https://www.csie.ntu.edu.tw/~cjlin/libsvm/. [Accessed 1 April 2018].

[8] "Thync Global Inc. Bioelectronic Devices, Neuromodulation, Bioelectronics," [Online]. Available: https://www.thync.com/. [Accessed 10 January 2018].

[9] "Spire - Live Better," [Online]. Available: https://www.spire.io/. [Accessed 15 January 2018].

[10] "EMOTIV EPOC+ - 14 Channel Wireless EEG headset," [Online]. Available: https://www.emotiv.com/epoc/. [Accessed 28 March 2018].

[11] D. Surangsrirat and A. Intarapanchi, "Analysis of the meditation brainwave from consumer EEG device," *SoutheastCon 2015*, pp. 1-6, 2015.

[12] InteraXon, "MusePlayer," [Online]. Available: https://bitbucket.org/interaxon/museplayer. [Accessed 28 March 2018].

[13] M. Martens, E. Tunbridge and P. Harrison, "Physiological and psychological effects of the Montreal Imaging Stress Task," *European Neuropsychopharmacology,* vol. 27, pp. S64-S65, 2017.

[14] A. Mohan, R. Sharma and R. L. Bijlani, "Effect of Meditation on Stress-Induced Changes in Cognitive Functions," *The Journal of Alternative and Complementary Medicine,* vol. 17, no. 3, pp. 207-212, 2011.

[15] "Guided Mindfulness Meditation," The Honest Guys, 2015. [Online]. Available: https://store.cdbaby.com/cd/thehonestguys40. [Accessed 20 March 2018].

[16] J. L. Ferreira, R. M. Aarts and P. J. M. Cluitmans, "Optimized moving-average filtering for gradient artefact correction during simultaneous EEG-fMRI," *5th ISSNIP-IEEE Biosignals and Biorobotics Conference (2014): Biosignals and Robotics for Better and Safer Living (BRC),* 2014.

[17] Y. Wang, S. Estate M., E.-B. Ayman S., L. Xiaoli, S. F. Lonnie and T. Allan, "Relative Power of Specific EEG Bands and Their Ratios during Neurofeedback Training in Children with Autism Spectrum Disorder," *Frontiers in Human Neuroscience,* vol. 9, 2016.

[18] J. Vanschoren, "OpenML: exploring machine learning better, together.," [Online]. Available: https://www.openml.org/a/estimation-procedures/1. [Accessed 2 April 2018].

[19] "FocusDataCollection," [Online]. Available: https://github.com/zhijiee/FocusDataCollection. [Accessed 10 Apr 2018].

[20] "Muse Tools," [Online]. Available: http://developer.choosemuse.com/tools. [Accessed 3 April 2018].

[21] "MuseEEGClassifier," [Online]. Available: https://github.com/zhijiee/MuseEEGClassifier. [Accessed 12 Apr 2018].

[22] "MATLAB SVM Classifier," [Online]. Available: https://github.com/zhijiee/matlab_svm_classifier. [Accessed 10 Apr 2018].

[23] "CircleProgress," [Online]. Available: https://github.com/lzyzsd/CircleProgress. [Accessed 10 March 2018].

[24] "GraphView," [Online]. Available: https://github.com/appsthatmatter/GraphView. [Accessed 1 April 2018].