EEG-BASED EMOTION RECOGNITION USING MACHINE LEARNING ALGORITHMS

BY

LAM YEE WEI

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONOURS)

Faculty of Information and Communication Technology

(Kampar Campus)

JUNE 2024

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: EEG-Based Emotion Recognition using Machine Learning Algorithms

Academic Session: JUNE 2024

I LAM YEE WEI ,

(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

(Author's signature) (Supervisor's signature)

Address:
3, Hala Keledang emas 11A,
Taman Keledang Emas,
30100 Ipoh, Perak.

DR NUR SYAHIRAH BINTI ROSLAN
Supervisor's name

Date: 13/09/2024 **Date**: 13/09/2024

FACULTY/INSTITUTE* OF Information and Communication Technoogy

UNIVERSITI TUNKU ABDUL RAHMAN

Date: 13/09/2024

SUBMISSION OF FINAL YEAR PROJECT /DISSERTATION/THESIS

It is hereby certified that Lam Yee Wei (ID No: 21ACB00138) has completed this final year project/ dissertation/ thesis* entitled " EEG-Based Emotion Recognition using Machine Learning Algorithms " under the supervision of Dr Nur Syahirah Binti Roslan (Supervisor) from the Department of Information Systems , Faculty/Institute* of Information and Communication Technology , and - (Co-Supervisor)* from the Department of - , Faculty/Institute* of - .

I understand that University will upload softcopy of my final year project / dissertation/ thesis* in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

(*Lam Yee Wei*)

*Delete whichever not applicable

DECLARATION OF ORIGINALITY

I declare that this report entitled "**METHODOLOGY, CONCEPT AND DESIGN OF A 2-MICRON CMOS DIGITAL BASED TEACHING CHIP USING FULL-CUSTOM DESIGN STYLE**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : ___________________________

Name : LAM YEE WEI

Date : 13/09/2024

ACKNOWLEDGEMENTS

I extend my sincere thanks to my supervisor, Dr Nur Syahirah Binti Roslan for her invaluable guidance and advice throughout this project journey. Her expertise and encouragement have been instrumental in shaping the direction of my work. I am truly thankful for her dedication and mentorship. Under her guidance, I am getting opportunity to explore on fascinating field which is Electroencephalogram (EEG) signals as my final year project. I am grateful for the opportunity to learn and grow under her supervision.

Besides, I would like to express my heartfelt thanks to my family and friends for their support. Their words of encouragement act as a source of motivation for me. Whenever I need some mental support and talk, they are always be there. Their support meant a lot to me so I can stay focused on achieving my goals. I am filled with gratitude for everyone who has been part of this journey. I am looking forward to continuing as I pursue my academic endeavours.

ABSTRACT

Human emotions are very complex and hard to identify based on their facial expressions and appearance. Humans can hide their emotions with positive appearance and facial expression. Traditional emotion recognition techniques such as conducting questionnaires and facial recognition to analyse emotion is not reliable. The result is varied and it is hard to define a standard as different people have different emotional levels. However, researchers have found out that physiological signals such as brain signal can be used to identify emotion accurately. It is because physiological signals are hard to control and more reliable.

Thus, this project proposed an optimised machine learning algorithms to classify emotion by analysing brain activity using Electroencephalogram (EEG) signals. Throughout this research study, models like Support Vector Machine (SVM), K-Nearest Neighbours (KNN) and Adaptive Boosting (AdaBoost) will be explored. This machine learning model is aimed to be implemented in various industries to overcome real-world challenges. Industries such as medical industry, business analysis in customer interested level, lie detectors and even for future research. In this project, SEED dataset will be used for training and testing purposes. The Electroencephalogram (EEG) signals from SEED dataset will be pre-processed and extracted using feature extraction techniques. Training will be conducted so the model can learn and capture patterns of data. Moreover, fine-tuning of model will be applied to get the optimal performance in machine learning model. An evaluation of overall performance for each machine learning will be carried out accordingly.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

EEG	Electroencephalogram
SVM	Support Vector Machine
KNN	K-Nearest Neighbours
VR	Virtual Reality
HFO	High Frequency Oscillation
RBF	Radial Basis Function
PSO	Particle Swarm Optimization
PSD	Power Spectral Density
MSCE	Magnitude Squared Coherence Estimate
DWT	Discrete Wavelet Transform
PCA	Principal Component Analysis
DE	Differential Entropy
PDF	Probability Density Function
SFS	Sequential Forward Selection

Chapter 1
Introduction

Human Emotion is very complex and hard to determine based on facial expressions. In general, behavioural signals such as eyebrow, angle curve of mouth, cheek, jawline is the parameter in identifying human emotion. It also analysed various aspects such as speech tone, body gesture and even eye blink. However, the features that we can identify from appearance can be challenging and not robust enough. A person who is feeling sad can intentionally hide their emotions by presenting a positive appearance. In contrast, a person who is introvert and happy is hard to identify whether they are happy or sad. It is because their facial expression does not actively present their positive emotion. In this scenario, it will lead to a false negative result in identifying negative emotion.

Other than behavioural signal, physiological signal also can be used in emotion recognition. For instance, Electroencephalogram (EEG) are examples of physiological signal collection. An EEG test will be conducted by attaching many electrodes placement on the hair scalp. EEG refers to electroencephalogram is a test conducted to measure frequency in human brain activity. Throughout the EEG test, the electrical activity of the human brain will be recorded and generated accordingly. In real-world situations, EEG has been applied in various industries such as neurophysiology, medical use, neurogaming in virtual reality (VR) and research purposes. The brain's neural activity serves as a powerful signal that can disclose significant information regarding human behavioural [1]. Therefore, developing a robust model to classify emotion states is necessary. In this project, we will make use of machine learning to classify emotions based on EEG result.

1.1 Problem Statement and Motivation

Emotion is fundamental in human interaction in daily life. Emotion is complex and hard to determine among people. Some techniques such as body language analysis, voice analysis and eye-tracking analysis are the common techniques to analyse human emotion. Alternatively, the accuracy of the analysis result is questionable. There are some limitations of current emotion recognition such as performing questionnaires on human emotions. It is because the result of the questionnaire is very subjective and varies among individuals. Level of happiness and sadness can differ from one another. Some may easily feel happy, but some find it hard to find themselves in high spirits. Additionally, limitations such as facial recognition for human emotion are not accurate enough in identifying human emotions. Humans nowadays are more likely to conceal their real emotions behind a positive appearance. In contrast, Electroencephalogram (EEG) can overcome these limitations by capturing and interpreting brain activity. Using EEG result is more accurate because brain activity is challenging to manipulate. It truly reflects how individuals respond from different scenarios and situations. Nevertheless, EEG signals are complex and diverse of emotional level. Based on the above considerations, development of an efficient machine learning system for EEG-based emotion recognition will be advantage in classification human emotions. A robust and effective machine learning model gives different insights and better understanding of human emotions.

1.2 Motivation

The motivation of this research project is transforming the potential impact of EEG-based emotion recognition in adapting to real-world challenges. An efficient and mature emotion recognition system can be utilised in various domains. For instance, emotion recognition in EEG-based machine learning can improve business predictions on understanding customer behavioural. When a customer is wearing EEG device, an instant result of product interested level can be generated by interpreting EEG result of customer. It may help in enhancing user engagement and business marketing strategy. Furthermore, EEG-based emotion recognition in machine learning can also be implemented in lie detectors. In lie detectors, the common physiological responses measured heart rate, blood pressure and skin conductance. If EEG-based emotion recognition is implemented, the accuracy and performance of lie detectors will increase. As a result, the potential of developing EEG-based emotion recognition in machine learning is unlimited and beneficial to overcome real-world challenges.

.

1.4 Research Objectives

1. **To develop an optimized machine learning for classification on emotion recognition based on Electroencephalogram (EEG) signals.**

    (i) To classify human emotions status accurately based on Electroencephalogram (EEG) signals using machine learning algorithms.

    In this project, an optimised algorithm will be identified to enhance the overall performance of machine learning model and ability of emotion classification. Optimisation such as hyperparameter tuning is performed to identify the best combination of hyperparameter. Different combinations of hyperparameter will impact the model performance.

    (ii) To evaluate performance of different machine learning in classification task of emotion recognition.

    Model performance of each machine learning will be evaluated accordingly. Evaluation metrics such as accuracy, precision, recall and F1 score will be carried out. It consists of information that describes the classification performance from a dataset such as class of correctly identified positive for True Positive (TP), correctly identified negatives for True Negative (TN), incorrect identified positives for False Positive (FP) and incorrect identified negatives for False Negative (FN). Besides, F1 scores will combine information precision and recall metrics. Precision refers to the accuracy of true predictions and recall refers to the capability of model in identifying true prediction.

1.3 Project Scope

The scope of this project is to study and classification EEG-based emotion recognition using different machine learning algorithms. For instance, such as Support Vector Machine (SVM), K-Nearest Neighbours (KNN) and Adaptive Boosting (AdaBoost). This research paper will use SEED dataset to study and develop machine learning model. For data preprocessing, dataset will be cleaned and perform feature extraction. This project will also explore machine learning optimisation. An optimised machine learning model will be developed to classify emotion states into positive, neutral or negative. This project aims to achieve robust emotion classification. Performance analysis of different machine learning will be carried out to evaluate their classification ability

1.5 Contribution

Throughout this research project, the primary contribution is to develop a highly efficient machine learning model to classify diverse human emotions. This model will be using advanced machine learning techniques and signal processing to study EEG data comprehensively. A model with high accuracy will be developed to maximise the overall performance in emotion recognition.

Chapter 2
Literature Review

2.1 Background Information

2.1.1 Electroencephalogram (EEG)

In international standards, there are two types of Electroencephalogram (EEG) system used around the world. 10-10 EEG system and 10 -20 EEG system are examples of standard systems used in experiments and recordings. The differences between the 10-10 system and 10-20 system are the number and distribution of electrode in EEG.

For 10-20 system EEG, the placement of electrode is 10% or 20% from the region of brain [2]. It will be placed in the Frontal, Central, Temporal, Posterior and Occipital brain region. Frontal (F) refers to the front region of the brain. It works on problem solving, speech, personality and emotional reaction of human. Central (C) of brain is helping human in identifying object and physical processing. Temporal (T) region is responsible in controlling human emotions, recalling memory and interpreting languages. Posterior (P) is taking part in decision making while Occipital (O) is playing a significant role in visual processing.

Figure 2.1 shows the electrode placement of 10-10 system and 10-20 system in an EEG. Black circles illustrate the 10-20 system while grey circles illustrate the 10-10 system. The label begins with Frontal (F), Central (C), Temporal (T), Posterior (P), Occipital (O) and followed by a number. Odd numbers are placed at left brain region and even numbers are placed at right brain region accordingly.

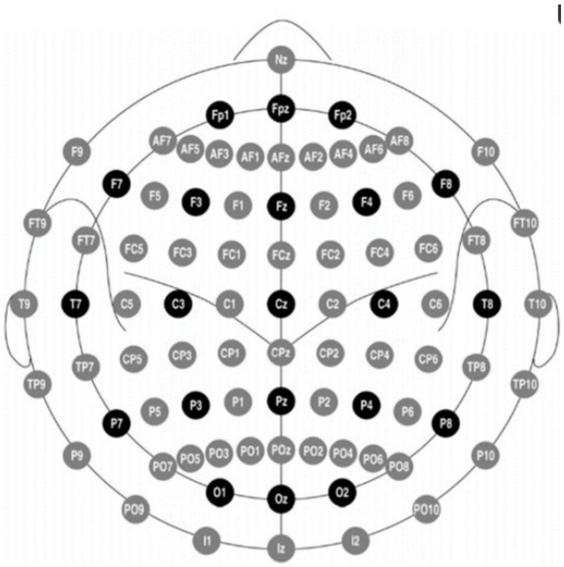


Figure 2.1 Electrode placement of 10-10 system EEG and 10-20 system EEG

Source: Adapted from [2]

2.1.2 Frequency band in Electroencephalogram (EEG)

Throughout the physiological signal collection, there are several types of band frequency that can be obtained in EEG result. This band frequency is used to category EEG waves using frequency. They are Delta waves, Theta waves, Alpha waves, Beta waves, Gamma waves, Ripple waves and Fast Ripple waves. According to [3], Delta waves are the smallest and slowest brainwave that captured in EEG. In general, its range is within 0.5 Hz to 4 Hz. This frequency mostly occurs during deep sleeping stage and physically observed through the frontocentral brain region. The frontocentral brain region is at the upper front of the brain. Brain and nerve cells remain active while sleeping. Thus, slowest brainwave still can be detected throughout the sleep. For Theta waves, the frequency range is from 4 Hz to 8 Hz. This stage is called the early sleep stages, such as people feeling sleepy and thinking phases.

Besides, the range from 8 Hz to 12 Hz is under Alpha waves. This frequency band indicates that a person is in relaxed form and awake. It can be seen through the occipital brain region at the back of the brain region. In EEG, Alpha waves are also defined as the general brain activity frequency for an adult. For Beta waves, it falls from 13 Hz to 30 Hz. In this stage, humans are more likely to interact with complex decision making and active brain activity. If frequency band is higher than 30 Hz, it will be categorized under High Frequency Oscillations (HFO) [3]. HFO includes Gamma waves ranging from 30 Hz to 80 Hz, Ripples waves from 80 Hz to 200 Hz, and Fast Ripples waves from 200 Hz to 500 Hz. For High Frequency Oscillations (HFO), the frequency waves usually will be observed through multiple brain regions. The most active brain region in HFO is located at the outer-side brain namely lateral temporal and upper-front brain namely prefrontal brain [4].

Figure 2.2 illustrates the weight distribution for five different frequency bands in from Delta waves to Gamma waves. In Beta and Gamma waves, lateral temporal and prefrontal brain is actively performing compared to Delta, Theta and Alpha waves.

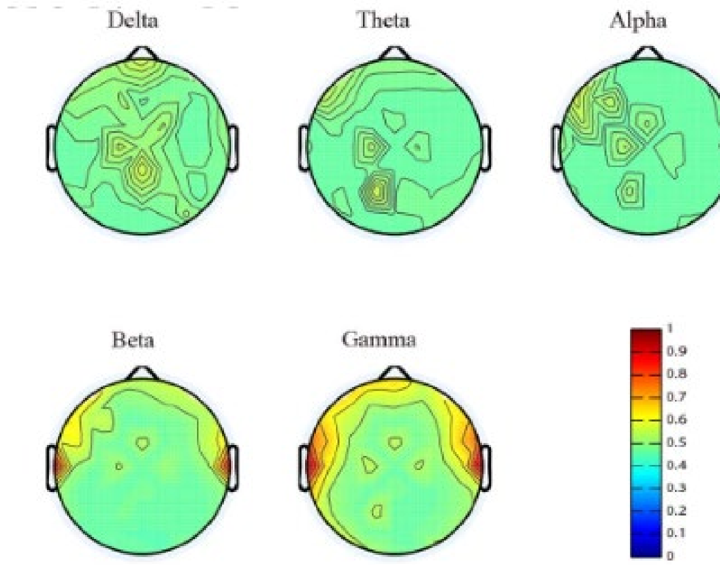


Figure 2.2 The brain region with weight distribution in different frequency band.
Source: Adapted from [4]

Figure 2.3 shows the different category of Electroencephalogram (EEG) waves included Delta, Theta, Alpha, Beta and Gamma. As the frequency band increases, the EEG waves become more active.

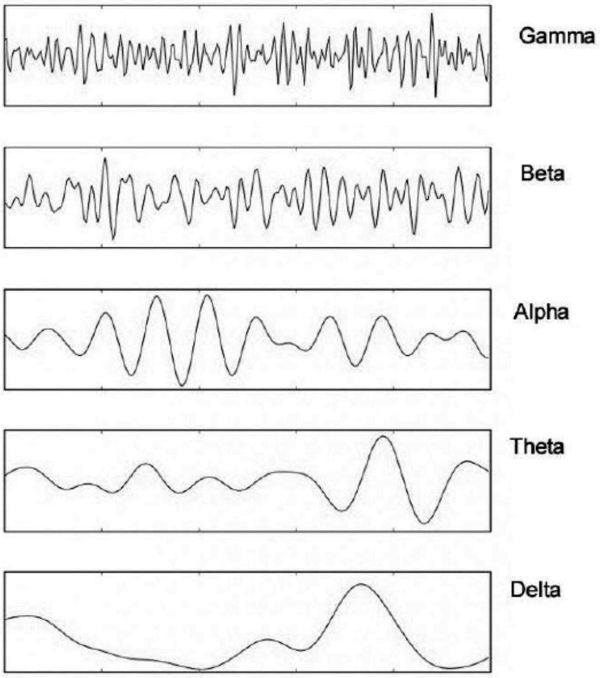


Figure 2.3 Electroencephalogram (EEG) waves for frequency band.
Source: Adapted from [5]

2.2 Machine Learning Algorithm in EEG-based Emotion Recognition

2.2.1 Support Vector Machine (SVM)

Support Vector Machine (SVM) is one of the supervised machine learning used in solving real-world problems. It is widely applicable in tasks that involving classification and regression tasks. In SVM, a hyperplane that finds the best boundary of multi dimension spaces into different category [6]. It is a straight line that segment input points/vector into a suitable class. A new input vector will be put into the correct category based on the hyperplane. Margin is a gap between vectors and the hyperplane. A vector/point closest to the hyperplane is known as support vector. It represents a maximized margin between two classes of the hyperplane. When the margin is maximized, an optimal hyperplane is generated for future training and testing. There is a unique way to define hyperplane for linear SVM and non-linear SVM.

Figure 2.4 shows a hyperplane for linear SVM. It maximises the margin between support vectors of different classes to obtain an optimal hyperplane. Optimal hyperplane is the line used to determine new vector into correct class in future.

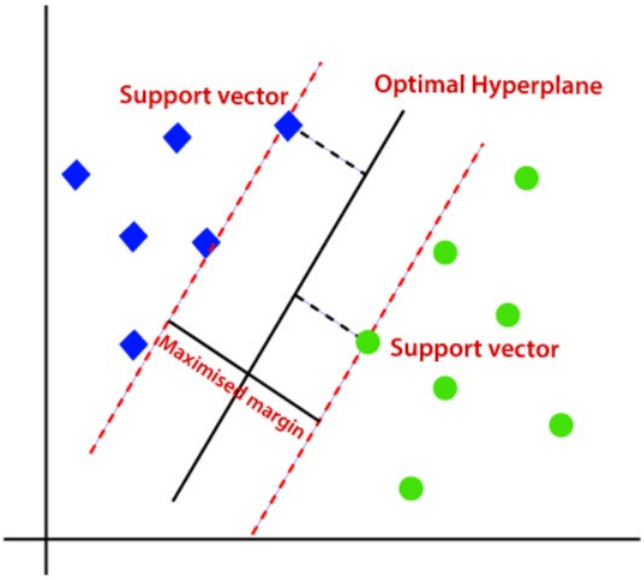


Figure 2.4 Hyperplane of linear Support Vector Machine (SVM)

Source: Adapted from [6]

Figure 2.5 shows a hyperplane for non-linear SVM. In non-linear SVM, a third dimension is added to calculate hyperplane. The circle represents a hyperplane for non-linear SVM in 2d dimension.

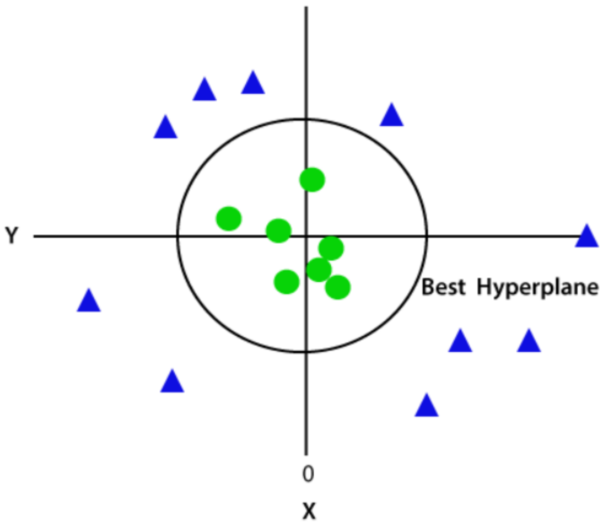


Figure 2.5 Hyperplane of non-linear Support Vector Machine (SVM)

Source: Adapted from [6]

2.2.2 K-Nearest Neighbours (KNN)

K-Nearest Neighbours (KNN) is one of the supervised machine learning algorithms widely applicable in classification and regression tasks too. In KNN, it will group a class based on their similarity features and characteristics [7]. It used variable k which indicates the number of neighbours around a new input point. Variable k is an important input parameter in this machine learning. Variable k is taken as a voting parameter to identify the class of the new input point. If k value is set to 5, the nearest 5 points among the new input point will take a vote to classify class for new input. In this point, Euclidean distance algorithm in (2.1) is used to get the 5 nearest points.

$$\sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2} \tag{2.1}$$

However, there are some rules in setting the value of k. Numbers in k value such as 1 and 2 are not encouraged to be used in classification task. If value k is too small, it will lead to unstable prediction as it only takes few nearest inputs to classify on new input. It may not capture the underlying patterns appropriately in a dataset.

Figure 2.6 illustrates the general concept of K-Nearest Neighbours (KNN) with k value. The red star represents a new input in the dataset. When k value is set to 3, two purple points and 1 yellow points will take a vote. If k value is set to 6, three additional points will participate in the vote. It is more robust and accurate as it captures the similarity underlying pattern among the majority.

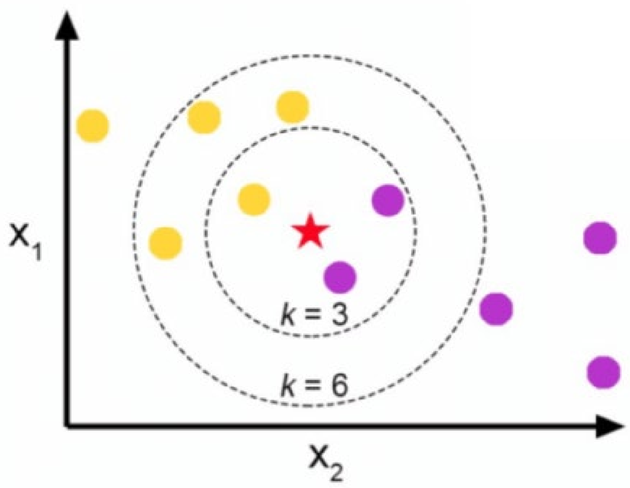


Figure 2.6 K-Nearest Neighbours (KNN) with k value 3 and 6.

Source: Adapted from [7]

2.2.3 Adaptive Boosting (AdaBoost)

Adaptive Boosting (AdaBoost) is machine learning algorithm apart from the ensemble methods. It usually used for classification and regression tasks. AdaBoost is aims to combine various weak classifiers to a strong classifier [21]. Firstly, the weak classifier is trained on the available dataset. It utilized decision tree as the weak learners in this machine learning algorithm. After the weak classifier is trained, AdaBoost will evaluate its training performance based on the training data. If instances classify incorrectly, it will receive higher weights. In contrast, instances which classify correctly will receive lower weights. It will continue to train and receive weights based on prediction results and focus on misclassification data iteratively. By combining predictions of various weak learner, it can achieve better generalization and performance compared to single classifier.

2.3 Literature Review on previous paper.

In [11], researcher has used DEAP dataset in Support Vector Machine (SVM). DEAP dataset has 32-channel of Electroencephalogram (EEG) with physiological result [12]. The dataset is down sampling to 128 Hz and 4Hz to 45Hz of bandpass filtering is applied in this dataset. For this paper, researchers used Russell's Circumplex Model to investigate power features from DEAP datasets. By implementing SVM, it has achieved a performance of 88.4% and 74% in Valence and Arousal. Valence is referring to the level of happiness in an event range from negative to positive while Arousal is the degree of physiological activity from relax to excited [13].

Figure 2.7 shows the Rusell's Circumplex Model. It includes two important parameters such as Arousal and Valence. Arousal is the level of energy range from calm to excited emotion state. For Valence, it represents the positiveness of emotion such as depression and delighted.

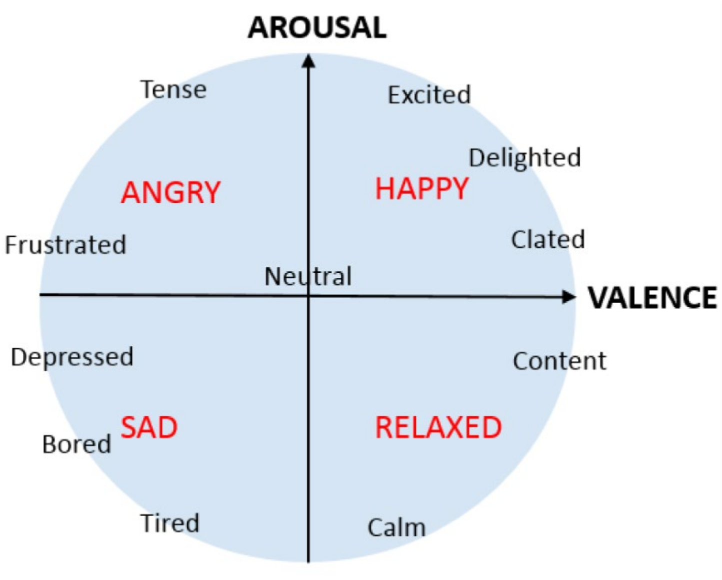


Figure 2.7 Valence and Arousal level in 2D form.

Source: Adapted from [8]

Besides, Radial Basis Function (RBF) kernel and Particle Swarm Optimization (PSO) is used to optimize the overall performance in SVM [14]. This paper has utilized different feature extraction such as Power Spectral Density (PSD) in (2.2), Magnitude Squared Coherence Estimate (MSCE) in (2.3) and Wavelet Sub-Band Energy and Entropy in (2.4). For PSD, PSD generates a mean frequency distribution of power density to monitor changes in maximal frequency. The use of PSD can improve the accuracy of classifier.

$$S(\omega) = \sum_{n=0}^{L-1} x(n)\omega(n)e^{-2\pi\left(\frac{\omega}{\omega_8}\right)n} \tag{2.2}$$

Moreover, Magnitude Squared Coherence Estimate (MSCE) is one of the features extractions. MSCE aims to highlight the frequency pattern of different signals in an activity. By using coherence estimation method, it is feasible to study functional relationship between different brain regions.

$$C_{xy}(f) = \frac{\left|P_{xy}(f)\right|^2}{P_{xx}(f)P_{yy}(f)} \tag{2.3}$$

For Wavelet Sub-Band Energy and Entropy, it can extract frequency band using Discrete Wavelet Transform (DWT). As per the abovementioned, frequency band has Delta, Theta, Alpha, Beta, Gamma etc. DWT will combine signal into scaling algorithms and wavelets accordingly. A signal is partitioned into specific coefficients using DWT algorithm [14].

$$DWT(x(t); a, n) = \int_{-\infty}^{\infty} x(t) \frac{1}{\sqrt{2^a}} \psi \left(\frac{t - 2^a n}{2^a} \right) dt \tag{2.4}$$

Among the three features extraction method, Particle Swarm Optimization (PSO) is used to select a suitable feature extraction. It operates by iteratively enhancing the position and velocity of each point to get the optimum solution [15]. In [14], it shows that machine learning of SVM with PSO algorithm has achieved accuracy of 80% and 85% in Valence and Arousal. It shows that PSO is successfully optimise the classification accuracy in SVM.

In [16], this paper applied energy in (2.5), entropy in (2.6) and Principal Component Analysis (PCA) in (2.7) features extracted to generate independent feature. DWT is applied to the chosen windowed EEG signals and 50% of overlapping to 2- and 4-seconds temporal windows is applied. Temporal window is a segmentation of time into small segments. Each window of frequency band will be used to extract energy and entropy. Entropy is used to measure the information of signal and uncertainty pattern in an EEG signal. Energy in every frequency band is calculated to ensure signal power is within the specific frequency range. The purpose of PCA is to maintain the same dimensionality of data. The output of PCA serves as the input parameters of machine learning classifier.

$$ENT_j = -\sum_{k=1}^{N} \left(D_j(k)^2 \right) \log\left(D_j(k)^2 \right) \tag{2.5}$$

$$ENG_j = -\sum_{k=1}^{N} \left(D_j(k)^2 \right) \, k = 1, 2, \ldots, N. \tag{2.6}$$

$$Z = X\phi \tag{2.7}$$

For classification, it used Radial Basis Function (RBF) kernel in (2.8) to transform input points into space of Gaussian. RBF kernel will assist SVM to find the optimum hyperplane to differentiate a class. By using PCA and RBF kernel in Beta band, performance of SVM achieved 91.3% and 91.1% in Valence and Arousal is performed in this study. It shows a tremendous improvement compared to the previous work.

$$K_{RBF}(x, x') = \exp \left[-\sigma \, \|x - x'\|^2 \right] \tag{2.8}$$

For K-Nearest Neighbours (KNN), this paper has tried out five different values of k from value three to seven. As a result, value k 0f 5 has achieved the optimum classification in this study. KNN has achieved a maximum of 79% in Arousal and 90.4% in Valence.

In [17], this paper used K-Nearest Neighbours (KNN) to classify EEG signal in DEAP dataset. The feature extraction used is wavelet transform (DWT), entropy and energy as per discussed above. Output of energy and entropy serve as the input parameter for KNN classifier. This paper has divided samples into 10 parts, 9 parts for training purpose while remaining parts for testing purpose. The value k in KNN is set to 3. As a result, Gamma frequency band has achieved a maximum of 95.70% of Valence and 95.69% of Arousal. This paper also highlight that Gamma frequency band is more suitable in classify emotion compared to other frequency band [17].

In [10], this paper used SVM and KNN machine learning throughout the study. It has introduced a new algorithm named differential entropy (DE) which is more powerful compared to the traditional energy spectrum. In DE, it refers to build features from the frequency band. Formula DE is formulated in (2.9). Besides DE, features such as DASM (2.10) and RASM (2.11) are also used to extract features from EEG signals.

$$h(x) = \frac{1}{2}\log(2\pi e\sigma^2) \tag{2.9}$$

$$DASM = h\left(X_i^{left}\right) - h\left(X_i^{right}\right) \tag{2.10}$$

$$DASM = h\left(X_i^{left}\right)/h\left(X_i^{right}\right) \tag{2.11}$$

Researchers have combined feature extraction such as DE, DASM, RASM to train their model. For dimension reduction, Principal Component Analysis (PCA) is used to improve the reliability and performance of model. It highlighted that Gamma frequency band is more suitable to apply in this proposed solution. As a result, DE has achieved an average accuracy of 84.22% while traditional energy spectrum only achieved accuracy of 76.56%. Thus, DE is more powerful in emotion recognition in EEG signals.

2.4 Comparison of previous paper.

Table 2.1 Comparison of previous work and techniques

Paper	Dataset	Technique Used	State classification	Machine Learning	Accuracy
[11]	DEAP	PSD, HOC	Valence and Arousal	SVM	88.4% in Valence, 74% in Arousal
[14]	DEAP	PSD, MSCE, Energy and Entropy	Valence and Arousal	SVM with PSO optimisation algorithm	80% in Valence, 85% in Arousal
[16]	DEAP	DWT, Energy and Entropy, PCA, Kernel RBF	Valence and Arousal	SVM with PCA	91.3% in Valence, 91.1% in Arousal
[16]	DEAP	DWT, Energy and Entropy, PCA, Kernel RBF	Valence and Arousal	KNN, k value of 5 with PCA	90.4% in Valence, 79% in Arousal
[17]	DEAP	DWT, Energy and Entropy	Valence and Arousal	KNN, k value of 3	95.70% in Valence, 95.69% in Arousal
[10]	Movie clips sample	DE, DASM, RASM, ES	Accuracy	SVM with PCA	84.22% of average accuracy

Chapter 3
System Model And Design

3.1 System Requirement

3.1.1 Hardware

Table 3.1 Specifications of laptop

Description	Specifications
Model	Dell Inspiron 5406 2n1
Processor	11th Gen Intel(R) Core(TM) i5-1135g7 @ 2.40GHz 1.38GHz
Operating System	Windows 10 Home
RAM	8GB
GPU	Intel (R) Iris (R) Xe Graphics

3.1.2 Tools and Software

Table 3.2 Tools and Software

Tools/Software	Descriptions
Programming Language	Python programming language
Software	Jupyter Notebook (Anaconda)
Machine Learning	Support Vector Machine (SVM), K-Nearest Neighbours (KNN) and Adaptive Boosting (AdaBoost)
Source	SEED experiment and SEED datasets

In this project, python programming language in Jupyter Notebook will be using throughout this study. Support Vector Machine (SVM), K-Nearest Neighbours (KNN) and Adaptive Boosting (AdaBoost) is used to develop machine learning models. SVM and KNN are well-known machine learning algorithms for classification and regression.

For dataset selection, SEED dataset will be used to develop an efficient machine learning model in classifying human emotion. In SEED experiment, there are fifteen short clips that extracted from Chinese movie. These short clips represent different emotions such as positive, neutral and negative emotions are selected. Each movie clip is around 4 minutes and trimmed appropriately to optimize human emotion. The clips must fulfil three requirements. First, the video clips should be short enough to avoid emotional exhaust. Secondly, trimmed video clips shall be able to understand for all experiment participants. Lastly, the video clips should represent a specific emotion among positive, neutral or negative emotion.

Figure 3.1 shows the details of video clips that were used in SEED experiment. Each clip is listed with emotion label accordingly. As a result, there are two negative video clips, three positive video clips and one neutral video clip.

No.	Emotion label	Film clips sources
1	negative	Tangshan Earthquake
2	negative	Back to 1942
3	positive	Lost in Thailand
4	positive	Flirting Scholar
5	positive	Just Another Pandora's Box
6	neutral	World Heritage in China

Figure 3.1 Examples of movie clips used in SEED experiment.

Source: Adapted from [18]

Figure 3.2 shows the process of experiments in detail. Firstly, 5 second hints will be displayed to the participants. After 5 seconds, participants will watch the trimmed 4 minutes video clip. Then, participants will conduct a self-assessment regarding the video clips. After 15 seconds of rest, participants will watch another video and repeat the entire process.

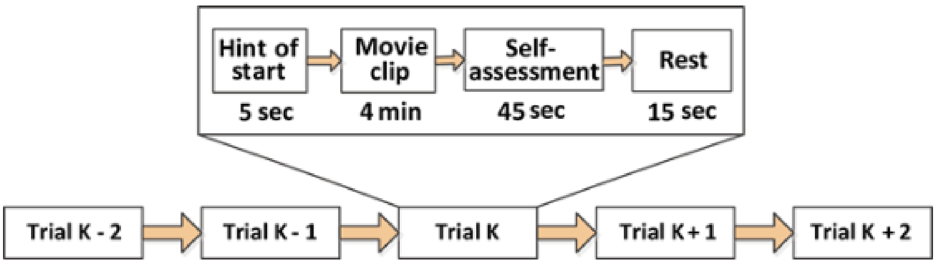


Figure 3.2 Process of SEED experiment and data collection.

Source: Adapted from [18]

For signal collection, SEED experiment used 10 – 20 system with 62 channels. **Figure 3.3** clearly illustrated the placement of 10-20 system with 62 channels.

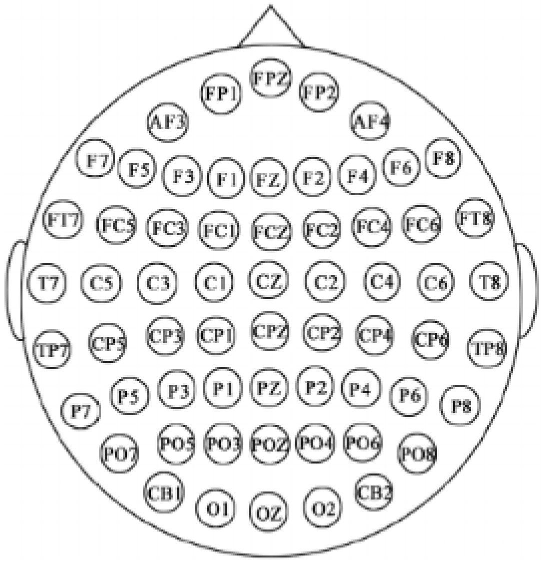


Figure 3.3 Layout of 10 – 20 with 62 channels.

Source: Adapted from [9]

3.1.3 System Performance

1. Accuracy

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{3.1}$$

Accuracy is measuring the accuracy correctly classified instances among all. It uses the proportion of True Positive plus True Negative divide by sum of other predictions class such as True Positive, False Positive, False Negative and True Negative. If the value of accuracy is near to value 1, which means that the model has high accuracy as the predictions are mostly correct for all.

2. Precision

$$Precision = \frac{TP}{TP + FP} \tag{3.2}$$

Precision in machine learning refers to the ratio of positive instances predicted. It takes the number of True Positive divide by True Positive plus False Positive. The value of precision is range from 0 to 1. Precision of 1 indicates that all the predicted instances with no false positive while precision of 0 indicates that all the predicted instances are false positive.

3. Recall

$$Precision = \frac{TP}{TP + FN} \tag{3.3}$$

Recall evaluates the ratio of predicted positive instances out of all actual positives. It aims to capture positive instances as much as possible. The value of recall is range from 0 to 1. Recall of 1 indicates that all the predicted instances with no false negative while precision of 0 indicates that all the predicted instances are false negative.

4. F1 Score

$$F1\,Score = 2 \times \frac{Precision\ \times Recall}{Precision + Recall} \tag{3.4}$$

F1 score refers to the overall performance of the model as it balances the value between precision and recall. The value of F1 score is range from 0 to 1. Best Value of 1 in F1 score indicates that the model has achieved perfect precision and recall while the worst case for F1 score will be value 0. Higher value reflects that the overall performance is high instead of comparing using accuracy rate only.

3.2 System Model

3.2.1 Overview Model

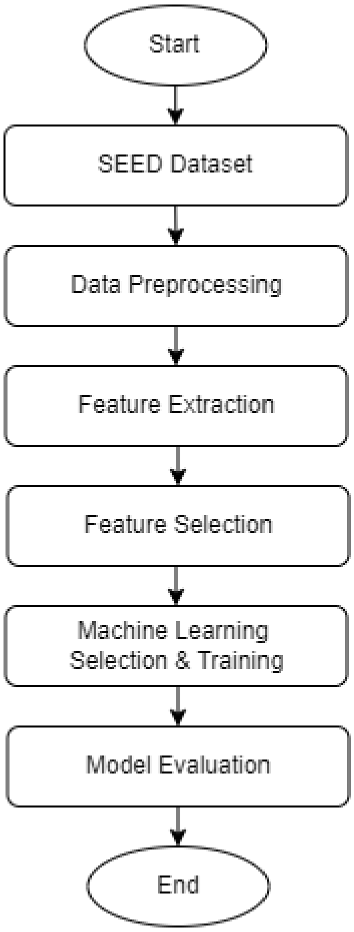


Figure 3.4 General flowchart of proposed model.

A general approach for the proposed system is illustrated above. According to the flowchart above, SEED datasets will be loaded into the system. Data preprocessing will be performed on the SEED datasets to extract useful data and eliminate useless data. Throughout data preprocessing, raw data is pre-processed to remove noise and unwanted signal. Data preprocessing is necessary as it can highly improves data consistency and usability for future training and testing. The original SEED dataset has been pre-processed and downsampled to 200 Hz. The only thing to do in preprocessing part is to trim a specific data time frame from the original 4 minutes data.

Feature extraction will be done to extract the key features. It captures relevant features and transforms features into representative features. Feature extraction techniques such as Differential Entropy (DE) will be studied in this project. After feature extraction, feature selection will be carried out to identify the combination pairs of electrodes among all electrodes. This process is optional and plays a crucial role in improving model performance. Then, the pre-processed dataset will be split into training and testing set. For machine learning selection, Support Vector Machine, K-Nearest Neighbours (KNN) and Adaptive Boosting (AdaBoost) will be implemented. Data will be trained and tested accordingly. At the end of the project, model evaluation will be carried out to analyse model performance for each machine learning. The model performance is measured using accuracy, precision and recall, F1 score and other measure metrics.

3.2.2 Data Preprocessing

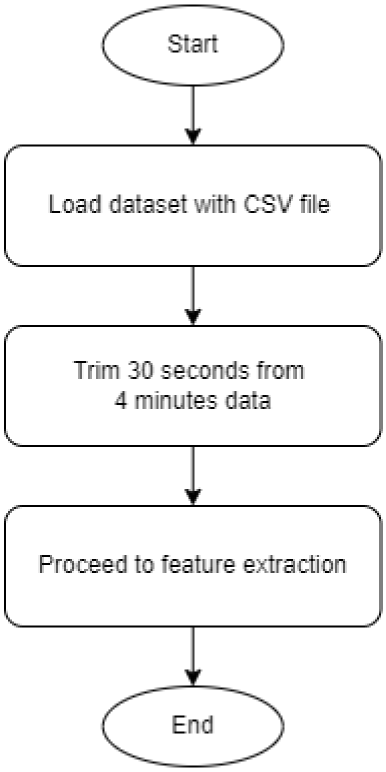


Figure 3.5 Flowchart of data preprocessing part.

The original SEED dataset consists of samples with 3 to 4 minutes each. Dealing with the full 4 minutes of data is very time-consuming and not practical. Therefore, some period adjustment has to be conducted in this dataset. In the first minutes, participants might prepare themselves to watch the video clips. At this stage, the EEG signal is not stabilised and informative enough. Besides, the EEG signal during initial phase is less consistent and their emotions are still settling. Less consistency will lead to fluctuations and noises in the EEG data. Moreover, the variety in video content also impacts the initial EEG readings. The variety of contents increases the complexity of the signal during the first minutes. In this case, the data will be trimmed and the 30 seconds will be extracted after 1 minute.

As SEED dataset is downsampled to 200Hz per second, which means there are 200 samples data per second. The start point will be 60 seconds x 200 samples/second which is 12,000 samples. The end point will be 12,000 samples + 30 seconds x 200 samples which is 18,000 samples. As the SEED EEG signal have been saved in CSV file format, the horizontal axis represents 62 electrodes while the vertical axis represents the time series. Therefore, the extraction starts from row 12,000 until row 18,000, which corresponding to 30 seconds after 1 minutes. When the specific time is extracted, the data is passing for feature extraction step.

3.2.3 Feature Extraction

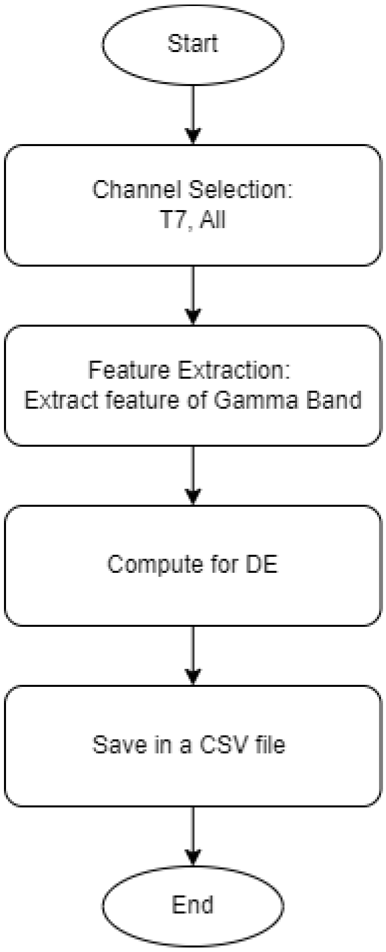


Figure 3.6 Flowchart of feature extraction process.

Figure 3.6 illustrates the detailed process of feature extraction before performing machine learning. In [10], researcher claimed that Gamma Band frequency is working closely and highly related with personal emotions. In **Figure 1.2**, it shows that our lateral temporal and prefrontal brain is highly active in Gamma waves. Among 62 electrode channels, this model has chosen channel T7 which is located at frontal temporal. Figure 3.6 has illustrated the Gamma band distribution level with the electrode placement. The red circle has highlighted the selected electrode of T7, which is located at frontal temporal and actively performing in Gamma frequency band.

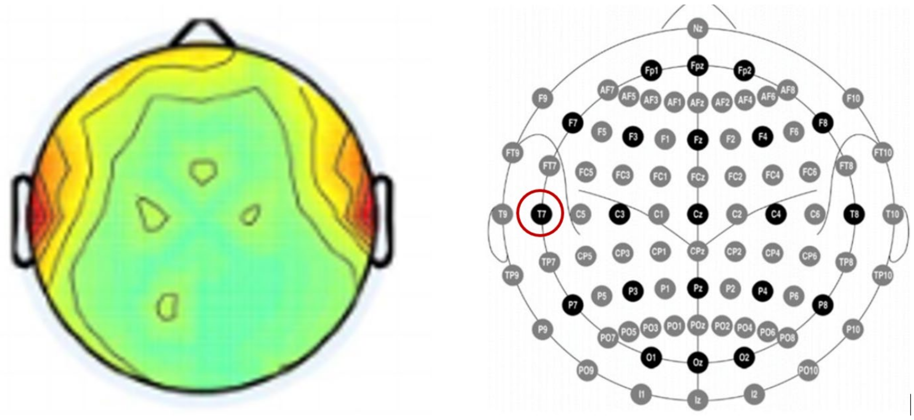


Figure 3.7 Gamma band distribution and electrode placement with selected electrode, T7.

Source: Adapted from [4] & [2]

For feature extraction, Gamma frequency will be extracted from channel T7. Besides channel T7, all electrodes will be extracted for future experiment use. By applying bandpass filter, this method decomposes EEG signal into different frequency bands such as delta, theta, alpha, beta and gamma. It will extract its coefficients and returns the gamma frequency signal. For Gamma frequency band, Gamma band is between 30 Hz to 48 Hz. Once the gamma frequency band is extracted, calculation of Differential Entropy (DE) will be conducted. DE value is used to qualify the randomness in the specific signal distribution. In [19], researcher claimed that Differential Entropy feature is more suitable than other features in emotion classification in EEG. The computation of DE derives the Probability Density Function (PDF). As a result, a list of entropy value will be calculated corresponding to different segments of feature. After getting ready with all the computed DE value, the DE value will be saved in a csv file for future use. For instances, it may use for feature selection process and machine learning process.

3.2.3 Feature Selection

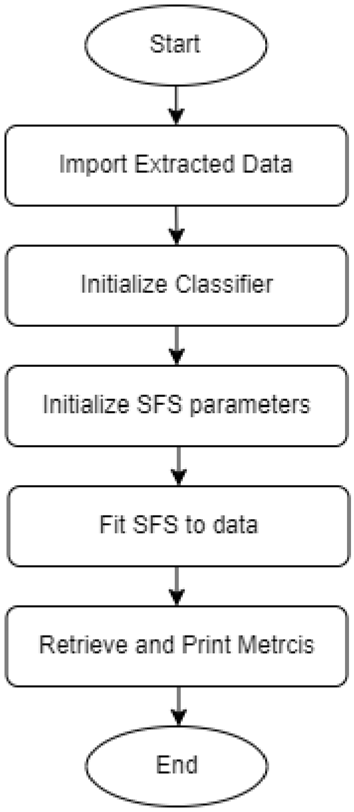


Figure 3.8 Flowchart of feature selection process.

In this study, Sequential Forward Selection (SFS) is implemented to select electrodes that are highly relevant for the classification task. According to [21], SFS selects or removes features through iteration process. The main benefit of using SFS is reducing the model complexity by using relevant features. It does improve the overall model performance in prediction tasks. Initially, the extracted data is loaded. Machine learning such as K-Nearest Neighbours (KNN), Support Vector Machine (SVM) and Adaptive Boosting (AdaBoost) is used as the model classifier. SFS provides two types of selection such as forward selection and backward selection. Forward selection is adding features while backward selection is removing features. Based on this study, forward selection is selected to select the optimal electrode combination among all electrodes. The SFS parameters are set to select 20 features using forward selection method and scoring the features with accuracy metric. After the feature selection, all selected features are retrieved for future model training.

3.2.4 Machine Learning (Training & Testing)

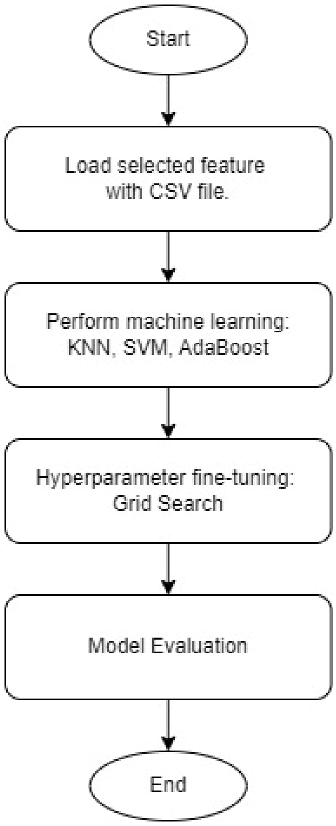


Figure 3.9 Flowchart of building machine learning model.

In the machine training process, the selected features loaded from a CSV file, it is necessary input data throughout the training phase. The data will split into 8:2 for training and testing purpose. Three machine learning will conduct in this project. The first algorithm is K-Nearest Neighbours (KNN), the second algorithm is Support Vector Machine (SVM) and the third algorithm will be Adaptive Boosting (AdaBoost). Each algorithm. Each machine learning algorithm will undergo training process to identify the relationship and learning the patterns of EEG signals for future prediction. After completed training process, it will go through testing for the dataset provided. Besides, hyperparameter tuning using Grid Search to optimize overall performance of the model. In [20], Grid Search responsible in finding the best hyperparameter for machine learning by using different combination ways. In Grid Search, it identifies model performance with cross-validation for consistent results. Subsequently, conducting model evaluation to analyse the performance for each algorithm. The evaluation phase will employ different metrics including accuracy, precision, recall and F1 score. Based on these performance metrics, it will serve as a guide for optimization of the machine learning models.

3.3 System Design

3.3.1 Implementation of Data Preprocessing

Code Snippet for Data Preprocessing

1. Extract 30 seconds after 1 minutes

   Description:

   - Define starting row of 12,001, which refers to 1 minute
   - Define ending row of 18,000, which refers to 30 seconds after 1 minute
   - Applying selected row into SEED EEG data for each sample.
   - Output: Data frame with 30 seconds duration after 1 minute.

```python
def process_eeg_folder(folder_path, fs, bands):
    all_de_values = []
    start_row = 12001  # Start row (1 minute, 200 Hz -> row 12001)
    end_row = 18000    # End row (30 seconds after 1 minute, row 18000)
```

Figure 3.10 Code snippet for extracting specific time frame.

3.3.2 Implementation of Feature Extraction

Code Snippet for Feature Extraction

1. Channel Selection: T7, All electrodes

   Description:

   - Utilized 'pd.read_csv' to extract EEG signal from 24th column.
   - In pandas data frame, the column is start from 0 (zero-based indexing). Thus, column 23 is the channel for T7.
   - Output: Data frame for channel T7.

```python
eeg_data = pd.read_csv(file_path)
eeg_signal = eeg_data.iloc[:, 23]
```

Figure 3.11 Code snippet for reading CSV files in model.

2. Extract Gamma Frequency Band

   Description:

   - Set for Gamma frequency band of 30-48 Hz.
   - Apply bandpass function using butter method from SciPy.
   - Output: Data which lied under Gamma band of 30 Hz to 48Hz.

```python
# Configuration: Define frequency bands (Gamma)
frequency_bands = {
    'Gamma': (30, 48)
}
fs = 200  #  sampling frequency
```

```python
# Function to apply bandpass filter
def bandpass_filter(data, lowcut, highcut, fs, order=4):
    nyquist = 0.5 * fs
    low = lowcut / nyquist
    high = highcut / nyquist
    b, a = butter(order, [low, high], btype='band')
    return lfilter(b, a, data)
```

Figure 3.12 Code snippet for bandpass filter Gamma frequency (30Hz to 48 Hz)

3. Differential Entropy

   Description:

   - Calculate entropy of each Gamma feature.
   - Compute the probability density function (PDF) using Kernel Density Estimation (KDE).
   - Calculate the entropy value by using formula of Differential Entropy(DE)

```python
def calculate_differential_entropy_kde(signal):
    kde = gaussian_kde(signal, bw_method='silverman')
    x = np.linspace(min(signal), max(signal), 1000)
    pdf = kde(x)

    pdf = pdf[pdf > 0]  # Remove zero probabilities to avoid log(0)
    diff_entropy = -np.sum(pdf * np.log2(pdf))

    return round(diff_entropy, 4)
```

Figure 3.13 Code snippet for computing Differential Entropy (DE) for Gamma frequency.

4. Save Selected Feature

   Description:

   - Save the selected feature in CSV file for further machine learning.

```python
def save_to_csv(data, output_file):
    df = pd.DataFrame(data)
    df.to_csv(output_file, mode='a', index=False, header=False)
```

Figure 3.14 Code snippet for saving targeted EEG signal into CSV file.

3.3.3 Implementation of Feature Selection

Code snippet for Feature Selection.

Import library of Sequential Forward Selection (SFS)

```
from mlxtend.feature_selection import SequentialFeatureSelector as SFS
from sklearn.metrics import accuracy_score
```

Figure 3.15 Code snippet for import library of Sequential Forward Selection (SFS).

1. K-Nearest Neighbours (KNN)

   Description:

   - Initialize KNN classifier for Sequential Forward Selection (SFS)
   - Initialize parameters for SFS with k_features of 20, forward selection and scoring by accuracy with 5 cross validations.
   - Fit the initialized SFS into KNN classifier.
   - Print the features combinations and performance metrics.

```
# Initialize the KNN classifier
knn = KNeighborsClassifier()

# Initialize Sequential Forward Selection
sfs = SFS(knn,
          k_features=20, # Number of features to select
          forward=True,
          floating=False,
          scoring='accuracy',
          cv=5)

# Fit SFS
sfs = sfs.fit(X_train, y_train.values.ravel())

# Get the metric dictionary to print results
metrics_dict = sfs.get_metric_dict()

# Print feature combinations and their corresponding accuracies
for step, metrics in metrics_dict.items():
    feature_indices = metrics['feature_idx']
    accuracy = metrics['avg_score']
    print(f"Top {len(feature_indices)} feature(s): {list(feature_indices)}
          with Accuracy: {accuracy:.4f}")
```

Figure 3.16 Code snippet for Sequential Forward Selection (SFS) using KNN.

2. Support Vector Machine (SVM)

   Description:

   - Initialize SVM classifier for Sequential Forward Selection (SFS)
   - Initialize parameters for SFS with k_features of 20, forward selection and scoring by accuracy with 5 cross validations.
   - Fit the initialized SFS into SVM classifier
   - Print the features combinations and performance metrics.

```
# Initialize the SVM classifier
svm = SVC(kernel='rbf', C=100, gamma=0.01, random_state=42)

# Initialize Sequential Forward Selection
sfs = SFS(svm,
          k_features=20, # Number of features to select
          forward=True,
          floating=False,
          scoring='accuracy',
          cv=5)

# Fit SFS
sfs = sfs.fit(X_train, y_train.values.ravel())

# Get the metric dictionary to print results
metrics_dict = sfs.get_metric_dict()

# Print feature combinations and their corresponding accuracies
for step, metrics in metrics_dict.items():
    feature_indices = metrics['feature_idx']
    accuracy = metrics['avg_score']
    print(f"Top {len(feature_indices)} feature(s): {list(feature_indices)}
          with Accuracy: {accuracy:.4f}")
```

Figure3.17 Code snippet for Sequential Forward Selection (SFS) using SVM.

3. Adaptive Boosting (AdaBoost)

Description:

- Initialize AdaBoost with Decision Tree classifier for Sequential Forward Selection (SFS)
- Initialize parameters for SFS with k_features of 20, forward selection and scoring by accuracy with 5 cross validations.
- Fit the initialized SFS AdaBoost classifier.
- Print the features combinations and performance metrics.

```
# Initialize AdaBoost with a base DecisionTreeClassifier
base_classifier = DecisionTreeClassifier(max_depth=1)
ada_classifier = AdaBoostClassifier(estimator=base_classifier, random_state=42)

# Initialize Sequential Forward Selection with AdaBoost
sfs = SFS(ada_classifier,
          k_features=20, # Number of features to select
          forward=True,
          floating=False,
          scoring='accuracy',
          cv=5)

# Fit SFS
sfs = sfs.fit(X_train, y_train.values.ravel())

# Get the metric dictionary to print results
metrics_dict = sfs.get_metric_dict()

# Print feature combinations and their corresponding accuracies
for step, metrics in metrics_dict.items():
    feature_indices = metrics['feature_idx']
    accuracy = metrics['avg_score']
    print(f"Top {len(feature_indices)} feature(s): {list(feature_indices)}
          with Accuracy: {accuracy:.4f}")
```

Figure 3.18 Code snippet for Sequential Forward Selection (SFS) using AdaBoost.

3.3.4 Implementation of Machine Learning

1. K-Nearest Neighbours (KNN)

   Description:

   - Normalize data with standard scaler function.
   - Dataset is splitting to 80% train set and 20% test set.
   - Utilized KNN Classifier to conduct machine learning.
   - Use Grid Search for hyperparameter tuning on the best k value from range 1-20.

```
# Standardize the input data using StandardScaler (Z-score normalization)
scaler = StandardScaler()
X_standardized = scaler.fit_transform(X_extracted)

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split
(X_standardized, y.values.ravel(), test_size=0.2, random_state=42)

# Hyperparameter Tuning using Grid Search (to find the best k)
param_grid = {'n_neighbors': range(1, 20)}

knn_classifier = KNeighborsClassifier()

grid_search = GridSearchCV(knn_classifier, param_grid, cv=5, scoring='accuracy')
grid_search.fit(X_train, y_train)  # Perform grid search using cross-validation

best_k = grid_search.best_params_['n_neighbors']  # Get the best k
best_knn_classifier = KNeighborsClassifier(n_neighbors=best_k)
best_knn_classifier.fit(X_train, y_train)  # Train with the best k
```

Figure 3.19 Code snippet for KNN training and testing.

   - Model evaluation using accuracy, precision, recall and F1 score.

```
# Evaluate the model
best_accuracy = accuracy_score(y_test, best_knn_classifier.predict(X_test))
best_precision = precision_score(y_test, best_knn_classifier.predict(X_test), average='weighted')
best_recall = recall_score(y_test, best_knn_classifier.predict(X_test), average='weighted')
best_f1 = f1_score(y_test, best_knn_classifier.predict(X_test), average='weighted')
```

Figure 3.20 Code snippet for model evaluation on KNN.

   - Perform cross-validation across k-value.

```
for k in k_values:
    knn_classifier = KNeighborsClassifier(n_neighbors=k)

    # Perform 5-fold cross-validation for each metric
    accuracy = cross_val_score(knn_classifier, X_train, y_train, cv=5, scoring='accuracy').mean()
    precision = cross_val_score(knn_classifier, X_train, y_train, cv=5, scoring=make_scorer
                                (precision_score, average='weighted')).mean()
    recall = cross_val_score(knn_classifier, X_train, y_train, cv=5, scoring=make_scorer
                             (recall_score, average='weighted')).mean()
    f1 = cross_val_score(knn_classifier, X_train, y_train, cv=5, scoring=make_scorer
                         (f1_score, average='weighted')).mean()

    # Append the cross-validated scores to their respective lists
    accuracies.append(accuracy)
    precisions.append(precision)
    recalls.append(recall)
    f1_scores.append(f1)
```

Figure 3.21 Code snippet for cross validation across each k-value.

2. Support Vector Machine (SVM)

   Description:

   - Normalize data with standard scaler function.
   - Dataset is splitting to 80% train set and 20% test set.
   - Use SVM classifier to conduct machine learning.
   - Perform Grid Search with cross-validation of 5.

```
# Standardize the input data using StandardScaler (Z-score normalization)
scaler = StandardScaler()
X_standardized = scaler.fit_transform(X_extracted)

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_standardized, y, test_size=0.2, random_state=42)

# Define the parameter grid to search
param_grid = {
    'C': [0.1, 1, 10, 100],           # Regularization parameter
    'gamma': [0.001, 0.01, 0.1, 1],    # Kernel coefficient
    'kernel': ['rbf', 'linear']        # Kernel type
}

# Initialize the SVM classifier
svm_classifier = SVC(probability=True)

# Perform grid search with cross-validation
grid_search = GridSearchCV(svm_classifier, param_grid, cv=5, scoring='accuracy', verbose=1)

# Fit the grid search to the data
grid_search.fit(X_train, y_train.values.ravel())

# Get the best parameters
best_params = grid_search.best_params_
print("Best Parameters:", best_params)

# Use the best parameters to train the final model
best_svm_classifier = SVC(**best_params, probability=True)
best_svm_classifier.fit(X_train, y_train.values.ravel())
```

Figure 3.22 Code snippet for SVM training and testing.

   - Model evaluation using accuracy, precision, recall and F1 score.

```
# Calculate evaluation metrics
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred, average='weighted')
recall = recall_score(y_test, y_pred, average='weighted')
f1 = f1_score(y_test, y_pred, average='weighted')
```

Figure 3.23 Code snippet for model evaluation on SVM.

3. Adaptive Boosting (AdaBoost)

Description:

- Dataset is splitting to 80% train set and 20% test set.
- Using Decision Tree classifier as base estimator with depth of 1.
- Use Grid Search for fine tuning using predefined parameters grid of estimators and learning rate.

```
# Standardize the input data using StandardScaler (Z-score normalization)
scaler = StandardScaler()
X_standardized = scaler.fit_transform(X_extracted)

# Step 3: Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X_standardized, y, test_size=0.2, random_state=42)

# Define the base classifier (Decision Tree with max_depth=1)
base_classifier = DecisionTreeClassifier(max_depth=1)

# Define the parameter grid to search
param_grid = {
    'n_estimators': [50, 100, 150],      # Number of estimators
    'learning_rate': [0.1, 0.5, 1.0]     # Learning rate
}
# Initialize the AdaBoost classifier
ada_classifier = AdaBoostClassifier(estimator=base_classifier, random_state=42)

# Perform grid search with cross-validation
grid_search = GridSearchCV(ada_classifier, param_grid, cv=5, scoring='accuracy', verbose=1)

# Fit the grid search to the data
grid_search.fit(X_train, y_train.values.ravel())

# Get the best parameters
best_params = grid_search.best_params_
print("Best Parameters:", best_params)

# Use the best parameters to train the final model
best_ada_classifier = AdaBoostClassifier(estimator=base_classifier,
                                         n_estimators=best_params['n_estimators'],
                                         learning_rate=best_params['learning_rate'],
                                         random_state=42)
best_ada_classifier.fit(X_train, y_train.values.ravel())
```

Figure 3.24 Code snippet for AdaBoost training and testing.

- Model evaluation using accuracy, precision, recall and F1 score.

```
# Calculate evaluation metrics
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred, average='weighted')
recall = recall_score(y_test, y_pred, average='weighted')
f1 = f1_score(y_test, y_pred, average='weighted')
```

Figure 3.25 Code snippet for model evaluation on AdaBoost.

3.4 Implementation Issues and Challenges

There are some of the challenges faced during the implementation. One of the significant challenge multiclass classifications to identify emotion state between positive, neutral and negative emotions. The complexity and randomness of EEG signals increase the difficulty of machine learning model development. It is crucial to develop a model which able to capture the complexity and randomness of EEG signals. It often leads to model overfitting issue as the model memorize unwanted noise rather than capturing the underlying pattern for each emotion class.

In addition, identifying optimal feature for classifying between three emotion state is another dauting challenge. EEG signals is complex and vary significantly among individuals. As individuals may response differently to the same emotion, it will be challenging to feature extraction, feature selection and even model development. As a result, identifying a unique characteristics of EEG signals between different class is a must to enhance the model overall performance and robustness.

Chapter 4
Experiment/Simulation

4.1 Phase 1 Model

In phase 1 classification model, channel T7 is utilised as the input feature for both multi-class classification and binary classification. For multi-class classification, the model aims to classify emotions into three classes: -1 refers to negative class, 0 refers to neutral class and 1 refers to positive class. In binary classification, emotions are classified into classes which are -1 and 1. Machine learning algorithms such as K-Nearest Neighbours (KNN), Support Vector Machine (SVM) and Adaptive Boosting (AdaBoost) are developed to evaluate the effectiveness of channel T7.

4.1.1 Phase 1 Experiment Result

1. K-Nearest Neighbours (KNN)

Table 4.1 Model Evaluation for K-Nearest Neighbours (KNN)

	(-1, 0, 1)	(-1, 1)
Best K	3	18
Accuracy	0.43	0.62
Precision	0.43	0.62
Recall	0.43	0.62
F1 Score	0.42	0.61

Note: -1 refers to negative class, 0 refers to neutral class and 1 refers to positive class.

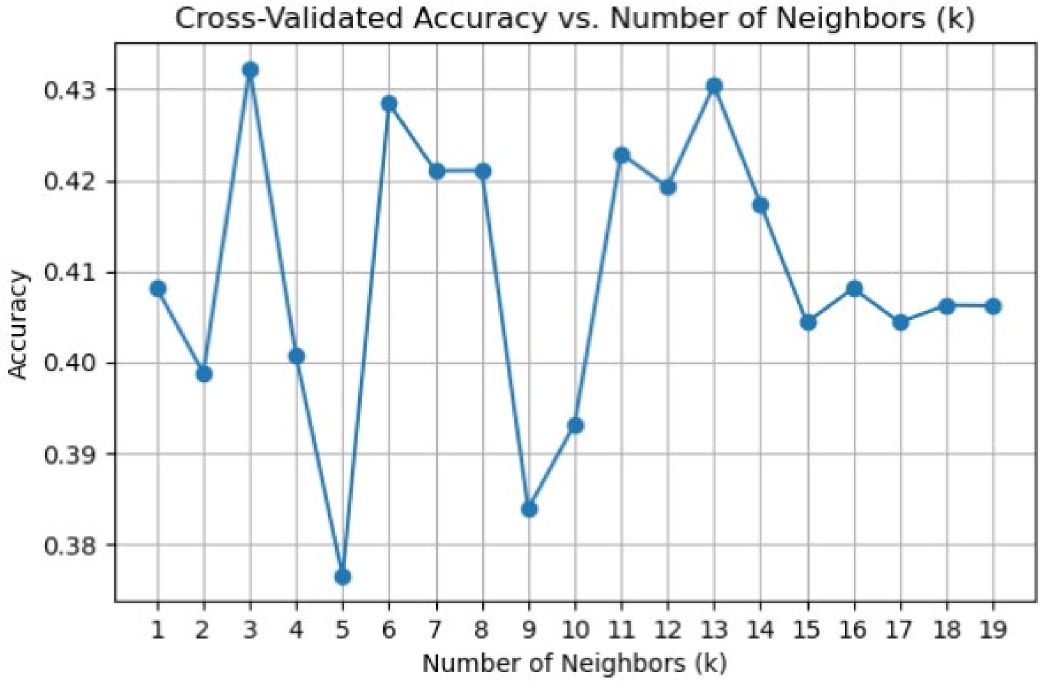


Figure 4.1 Accuracy of KNN machine learning in classifying all classes across value k of 1 to 20.

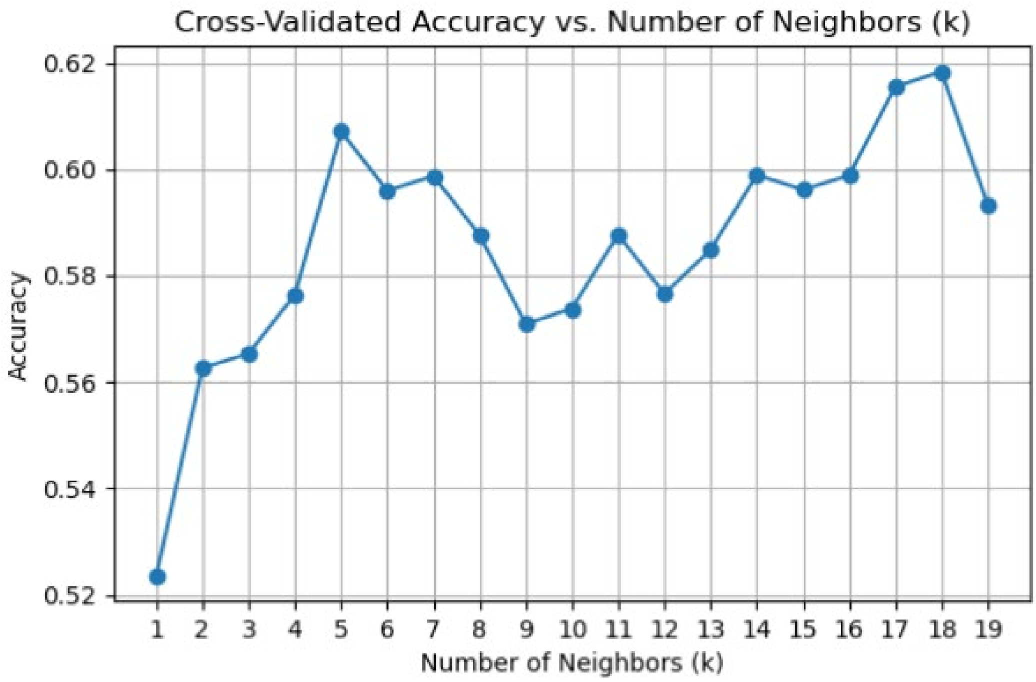


Figure 4.2 Accuracy of KNN machine learning in classifying binary classes across value k of 1 to 20.

2. Support Vector Machine (SVM)

Table 4.2 Model Evaluation for Support Vector Machine (SVM)

	(-1, 0, 1)	(-1, 1)
Best Parameter	'C': 10, 'gamma': 0.01, 'kernel': 'rbf'	'C': 10, 'gamma': 0.001, 'kernel': 'rbf'
Accuracy	0.39	0.52
Precision	0.46	0.52
Recall	0.39	0.52
F1 Score	0.34	0.52

Note: -1 refers to negative class, 0 refers to neutral class and 1 refers to positive class.

3. Adaptive Boosting (AdaBoost)

Table 4.3 Model Evaluation for Adaptive Boosting (AdaBoost)

	(-1, 0, 1)	(-1, 1)
Best Parameter	Learning rate: 0.5, N_estimators: 50	Learning rate: 0.5, N_estimators: 150
Accuracy	0.44	0.56
Precision	0.45	0.56
Recall	0.44	0.56
F1 Score	0.44	0.55

Note: -1 refers to negative class, 0 refers to neutral class and 1 refers to positive class.

4.1.2 Analysis on Phase 1 Experiment Result

Table 4.4 Model Evaluation for KNN, SVM and AdaBoost with accuracy, precision, recall and F1 score.

	KNN		SVM		AdaBoost	
	(-1, 0, 1)	(-1, 1)	(-1, 0, 1)	(-1, 1)	(-1, 0, 1)	(-1, 1)
Accuracy	0.43	0.62	0.39	0.52	0.44	0.56
Precision	0.43	0.62	0.46	0.52	0.45	0.56
Recall	0.43	0.62	0.39	0.52	0.44	0.56
F1 Score	0.42	0.61	0.34	0.52	0.44	0.55

Note: -1 refers to negative class, 0 refers to neutral class and 1 refers to positive class.

During phase 1 experiment, electrode T7 classification between all classes (-1, 0, 1) which refers to -1 (negative), 0 (neutral), 1 (positive) emotion and binary classes (positive and negative) emotion is conducted. In multi-class classification, the performance of machine learning is relatively low across all metrics. AdaBoost slightly outperforms with accuracy of 44% and consistent results for precision, recall and F1 score. KNN achieved an accuracy of 43% which is similar with the performance of AdaBoost. However, SVM only achieved accuracy of 39%, which is the lowest among all machine learning models. When the problem comes to binary classification, all models show a significant improvement in performance. KNN increases its performance from accuracy of 43% to 62%. It also achieved the highest accuracy among all machine learning models in binary classification. AdaBoost also improves its performance and achieved all metrics to approximately 56%. Similarly, SVM also increases its overall metrics to 52%. This trend reflects that binary classification performs better than multi-class classification in emotion recognition.

One of the assumptions made is the presence of neutral emotion class may make the model ambiguous in classification tasks. The models might face difficulty in managing classes caused by the neutral class. Thus, it results in lower performance across all metrics. Based on Table 4.4, binary classification results in significant improvement compared to multi-class classification. The removal of the neutral class allows models focus on differentiating between two distinct classes which is positive and negative. This indicates that KNN, SVM and AdaBoost able to handle binary classification effectively in emotion classification. Based on the phase 1 experiment result, the performance of models considered low as the highest accuracy is only 62% from KNN binary classification. Consequently, further evaluation of proposed system to enhance the model performance is needed. For channel selection, the current approach selects only one channel out of 62 channels to perform emotion classification. Papers like [14], [16] and [17] are using channel combination leads to accuracy improvement.

4.2 Phase 2 Model

4.2.1 Experiment Results from SFS

Sequential Forward Selection (SFS) is conducted to determine the optimal combination with different available electrodes. Different machine learning is applied for both multi-class classification and binary classification. At the beginning of feature selection, SFS will identify the most relevant combination of electrodes from 62 electrodes. It will select the best-performing electrode and adding other electrodes to maximise classification accuracy. The electrode combinations reflect which combinations and models perform better under each classification scenario. As the number of features selected in this case will select up to 20 features, the final pair will be the combinations of top 20 electrodes out of 62 electrodes.

Figure 4.3 shows a sample output when selecting electrodes through Sequential Forward Selection (SFS). For each iteration, electrodes will be added one by one to get the optimal combination until top 20 features. The output lists the index of selected features with the accuracy for every combination pair.

```
Top 1 feature(s): [5] with Accuracy: 0.4193
Top 2 feature(s): [5, 27] with Accuracy: 0.5123
Top 3 feature(s): [5, 15, 27] with Accuracy: 0.5881
Top 4 feature(s): [5, 15, 27, 44] with Accuracy: 0.6419
Top 5 feature(s): [5, 15, 27, 36, 44] with Accuracy: 0.6828
Top 6 feature(s): [5, 12, 15, 27, 36, 44] with Accuracy: 0.7347
Top 7 feature(s): [0, 5, 12, 15, 27, 36, 44] with Accuracy: 0.7458
Top 8 feature(s): [0, 5, 12, 15, 27, 36, 44, 49] with Accuracy: 0.7625
Top 9 feature(s): [0, 5, 9, 12, 15, 27, 36, 44, 49] with Accuracy: 0.7681
Top 10 feature(s): [0, 5, 9, 12, 15, 27, 36, 43, 44, 49] with Accuracy: 0.7847
Top 11 feature(s): [0, 5, 9, 12, 15, 27, 36, 41, 43, 44, 49] with Accuracy: 0.7903
Top 12 feature(s): [0, 5, 9, 12, 15, 27, 36, 41, 43, 44, 49, 50] with Accuracy: 0.7774
Top 13 feature(s): [0, 5, 9, 12, 15, 27, 32, 36, 41, 43, 44, 49, 50] with Accuracy: 0.7793
Top 14 feature(s): [0, 5, 9, 12, 15, 27, 32, 36, 41, 43, 44, 48, 49, 50] with Accuracy: 0.7904
Top 15 feature(s): [0, 5, 9, 12, 15, 27, 32, 36, 41, 43, 44, 48, 49, 50, 59] with Accuracy: 0.7904
Top 16 feature(s): [0, 5, 9, 12, 15, 27, 32, 36, 39, 41, 43, 44, 48, 49, 50, 59] with Accuracy: 0.7867
Top 17 feature(s): [0, 5, 9, 12, 15, 27, 32, 36, 39, 41, 43, 44, 48, 49, 50, 54, 59] with Accuracy: 0.7922
Top 18 feature(s): [0, 5, 9, 12, 15, 27, 31, 32, 36, 39, 41, 43, 44, 48, 49, 50, 54, 59] with Accuracy: 0.8014
Top 19 feature(s): [0, 5, 9, 12, 15, 24, 27, 31, 32, 36, 39, 41, 43, 44, 48, 49, 50, 54, 59] with Accuracy: 0.8033
Top 20 feature(s): [0, 5, 9, 11, 12, 15, 24, 27, 31, 32, 36, 39, 41, 43, 44, 48, 49, 50, 54, 59] with Accuracy: 0.7958
```

Figure 4.3 Sample output of feature selection which includes the index of selected features and accuracy for each combination pair.

Table 4.5 below shows the classification accuracy for different feature combinations. Combinations up to 20 electrodes is conducted for each multi-class and binary class using different models. It helps to provide an optimal feature selection and enhance the model performance.

Table 4.5 Classification accuracy with top 20 combinations using machine learning models like KNN, SVM and AdaBoost. Note: -1 refers to negative class, 0 refers to neutral class and 1 refers to positive class.

Classes	(-1, 0, 1)			(-1,1)		
Features	**KNN**	**SVM**	**AdaBoost**	**KNN**	**SVM**	**AdaBoost**
1	0.42	0.45	0.46	0.67	0.67	0.68
2	0.51	0.53	0.51	0.75	0.74	0.74
3	0.59	0.61	0.54	0.80	0.79	0.77
4	0.64	0.62	0.58	0.82	0.82	0.80
5	0.68	0.63	0.59	0.83	0.83	0.84
6	0.73	0.66	0.61	0.84	0.84	0.84
7	0.75	0.68	0.63	0.85	0.84	0.85
8	0.76	0.69	0.64	0.85	0.84	0.86
9	0.77	0.70	0.65	0.86	0.85	0.86
10	0.78	0.71	0.65	0.87	0.86	0.87
11	0.79	0.73	0.65	0.88	0.86	0.87
12	0.78	0.75	0.65	0.89	0.86	0.87
13	0.78	0.78	0.65	0.89	0.87	0.87
14	0.79	0.79	0.65	0.90	0.88	0.86
15	0.79	0.79	0.64	0.90	0.89	0.86
16	0.79	0.80	0.64	0.90	0.89	0.86
17	0.79	0.80	0.65	0.90	0.91	0.86
18	0.80	0.81	0.65	0.89	0.91	0.85
19	0.80	0.81	0.65	0.90	0.92	0.86
20	0.80	0.80	0.65	0.89	0.92	0.86
Average	0.73	0.71	0.62	0.85	0.85	0.84

Figure 4.4 and **Figure 4.5** show the accuracy performance of feature selection from 1 feature to 20 features through Sequential Forward Selection (SFS) for multi-class classification and binary classification. (-1, 0,1) refers to multi-class classification while (-1, 1) refers to binary classification. In the graph, the lines represent different

machine learning models used in selecting features. The red line corresponds to KNN classifier, the blue line represents SVM classifier and the green line represents AdaBoost classifier. These lines illustrate the performance trends of model over number of selected features and accuracy.

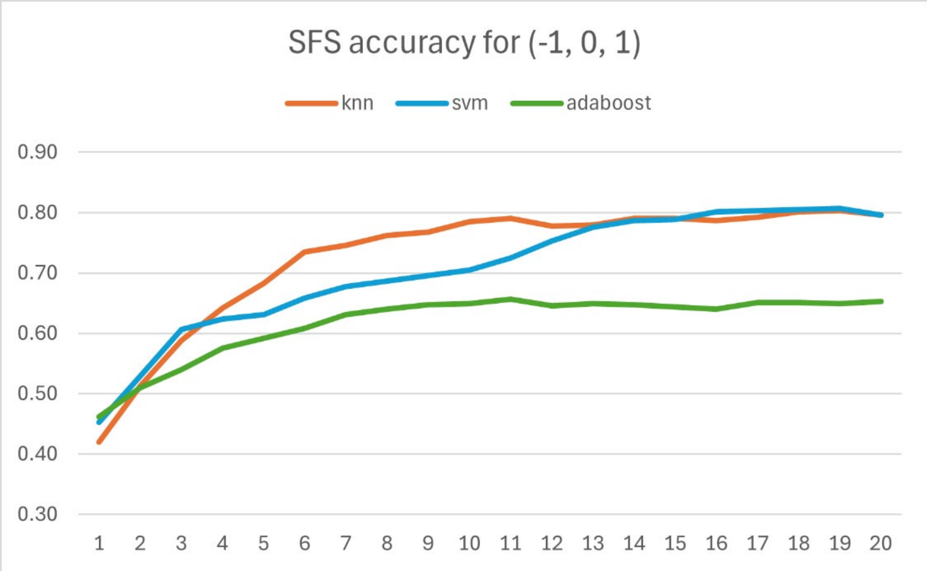


Figure 4.4 Accuracy performance of feature selection from 1 feature to 20 features through Sequential Forward Selection (SFS) for multi-class classification.[Note: -1 refers to negative class, 0 refers to neutral class and 1 refers to positive class.]

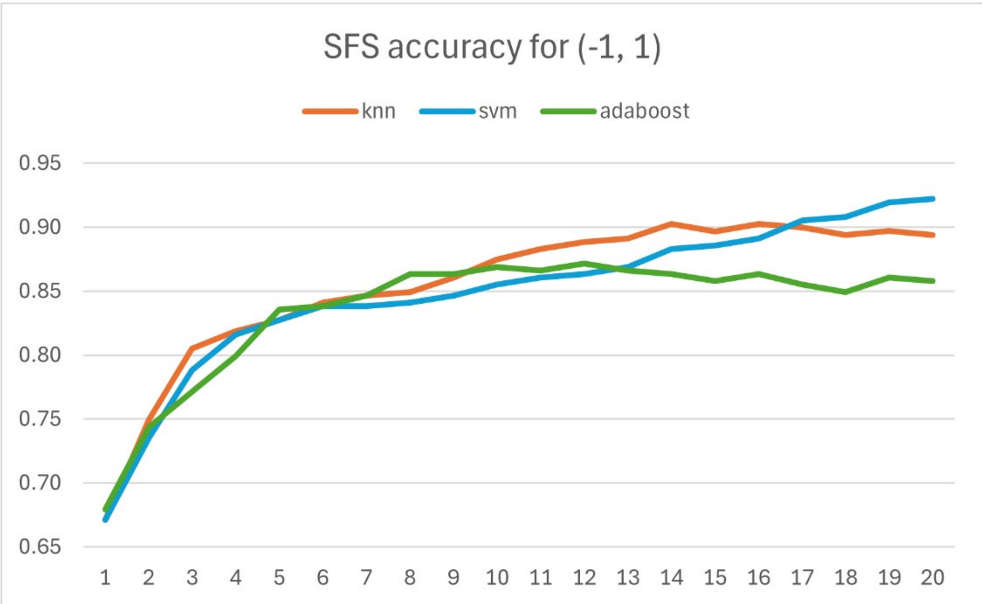


Figure 4.5 Accuracy performance of feature selection from 1 feature to 20 features through Sequential Forward Selection (SFS) for binary classification.[Note: -1 refers to negative class and 1 refers to positive class.]

4.2.2 Analysis on SFS Experiment Result

Based on **Figure 4.4** and **Figure 4.5**, a significant trend shows that combination of electrodes leads to an increase in performance. When the number of electrodes increases, the performance of models increases too. For multi-class classification, the accuracy across models is around 42% to 46% when only one electrode selected. As more features added, all models show improvement in terms of accuracy. The KNN model increases from 41% with one feature to 80% with 20 features. This improvement also goes to SVM model whose performance increases from 45% to 85%. It reflects that KNN and SVM can capture more underlying patterns with multiple electrodes instead of one electrode. However, AdaBoost model only shows a slight improvement from 46% to 65%. It reaches plateau at 10 features and remains flat at this stage.

After computing the average across all electrode combinations, KNN achieved the highest average of accuracy with 73% compared to SVM of 71% and AdaBoost of 62%. Thus, the feature selected using KNN will be used as the training combination pairs for multi-class classification. **Table 4.6** has included the list of features selected by KNN for top 1, 2, 3, 4, 5, 10 and 20. These selected electrodes will go through phase 2 experiment to improve the overall performance. **Figure 4.6** provides a visual representation of selected electrodes for multi-class classification.

Table 4.6 shows a list of features selected for top 1, 2, 3, 4, 5, 10 and 20 via Sequential Forward Selection (SFS) using KNN algorithm for multi-class classification. Initially, feature F7 is selected as the first feature and added until 20 electrodes.

Table 4.6 List of electrodes selected for top 1, 2, 3, 4, 5, 10 and 20 selected using KNN.

Top	Electrode Selected
1	F7
2	F7,CZ
3	F7, FC5, CZ
4	F7, FC5, CZ, P1
5	F7, FC5, CZ, CPZ, P1
10	FP1, F7, FZ, F6, FC5, CZ, CPZ, P3, P1, P8
20	FP1, F7, FZ, F4, F6, FC5, C5, CZ, T8, TP7, CPZ, CP6, P7, P3, P1, P6, P8, PO7, PO4, OZ

Figure 4.6 visualises the selected electrode placement by on the scalp. The 20 selected electrodes are highlighted with red circles which chosen as the optimal electrode combinations during Sequential Forward Selection (SFS) for multi-class classification task. Based on the visualisation, the key electrodes are distributed across different regions of brain. For instances, areas such as frontal (FP1, F7), central (CZ,CPZ) and parietal (PZ) regions.

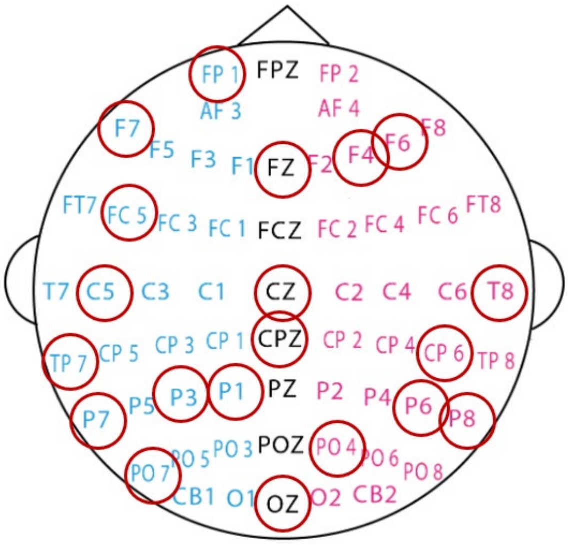


Figure 4.6 Visualization of the top 20 electrodes selected via Sequential Forward Selection (SFS) using KNN algorithm for multi-class classification.

Following the discussion on multi-class classification, the next section will address binary classification. The initial performance for all models starts with higher accuracy compared to multi-class classification. KNN and SVM begin with 67% while AdaBoost begins at 68%. As the number of selected features increases, all models show strong improvement in accuracy. Based on **Table 4.5**, KNN improves from 67% to approximately around 90% at the end of feature selection. Besides, SVM shows the highest improvement which is 67% and achieved the highest accuracy of 92% at 19 features. In contrast, AdaBoost performs well at the earlier feature selection. After being selected for 10 features, it remains at accuracy around 87%.

After computing the average across all electrode combinations, KNN and SVM achieved the same average accuracy of 85%. However, the performance of SVM is better than KNN after selecting 17 electrodes until the end. It shows that SVM is more capable in leveraging additional features. Thus, the feature selected using SVM model will be used as the training combination pairs for binary classification. **Table 4.7** has included the list of features selected by SVM for top 1, 2, 3, 4, 5, 10 and 20. These selected electrodes will go through phase 2 experiment to improve the overall performance. **Figure 4.7** provides a visual representation of selected electrodes for binary classification.

Table 4.7 shows a list of features selected for top 1, 2, 3, 4, 5, 10 and 20 via Sequential Forward Selection (SFS) using SVM algorithm for binary class classification. Initially, feature F3 is selected as the first feature and added until 20 electrodes.

Table 4.7 List of electrodes selected for top 1, 2, 3, 4, 5, 10 and 20 selected using SVM.

Top	Electrode Selected
1	F3
2	F3,CZ
3	F3, CZ, PO8
4	F3, CZ, P1, PO8
5	F3, FC4, CZ, P1, PO8
10	F3, FT7, FC2, FC4, FT8, CZ, P1, PO7, POZ, PO8
20	FPZ, AF3, F3, FT7, FC5, FC2, FC4, FC6, FT8, T7, CZ, CP1, CPZ, P7, P5, P1, PO7, POZ,PO8,CB2

Figure 4.7 visualises the selected electrode placement on the scalp. The 20 selected electrodes are highlighted with red circles which chosen as the optimal electrode combinations during Sequential Forward Selection (SFS) for binary class classification task. The selected electrode combinations are different compared to multi-class classification. However, some electrodes remain consistent across both classification such as FC5, CZ, CPZ, P1, P7 and PO7.

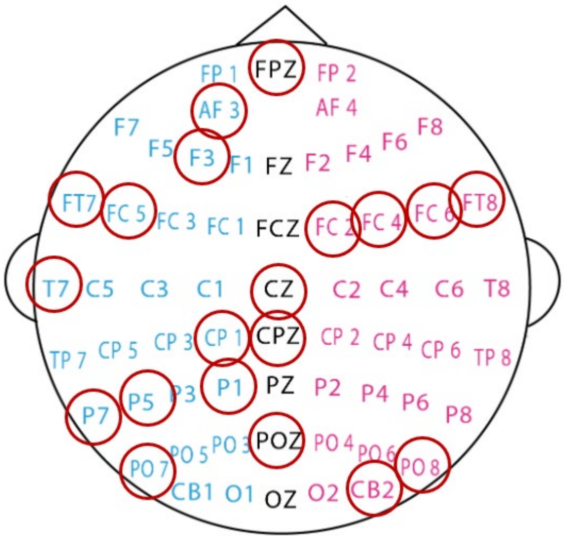


Figure 4.7 Visualization of the top 20 electrodes selected via Sequential Forward Selection (SFS) using SVM algorithm for binary class classification.

4.2.3 Phase 2 Experiment Result

In phase 2 model training, the models use features selected through KNN Sequential Forward Selection for multi-class classification and features selected through SVM Sequential Forward Selection for binary classification. The selected feature is the input features throughout the machine learning training and testing phase.

Table 4.8 summarizes the accuracy performance of three classifiers: KNN, SVM and AdaBoost for multi-class (-1, 0, 1) and binary (-1, 1) classification. The classifiers are evaluated with initial selected feature T7, all electrodes and different subsets through Sequential Forward Selection (SFS). For SFS, it includes the top 20, 10, 5, 4, 3, 2 and 1 feature. The highlighted column indicating the highest accuracy for each model across different input features.

Table 4.8 Accuracy performance of classifiers using different features as input.

Note: -1 refers to negative class, 0 refers to neutral class and 1 refers to positive class.

Features	KNN		SVM		AdaBoost	
	(-1, 0, 1)	(-1, 1)	(-1, 0, 1)	(-1, 1)	(-1, 0, 1)	(-1, 1)
T7	K=3, 0.43	K=18, 0.62	0.39	0.52	0.44	0.56
All	K=1,0.76	K=1, 0.87	0.81	0.92	0.65	0.87
Top 20	K=1, 0.82	K=3, 0.88	0.86	0.86	0.70	0.88
Top 10	K=3, 0.79	K=11, 0.83	0.79	0.80	0.69	0.74
Top 5	K=1, 0.69	K=17, 0.81	0.68	0.76	0.60	0.76
Top 4	K=3, 0.66	K=11, 0.80	0.58	0.74	0.56	0.70
Top 3	K=8, 0.60	K=11, 0.76	0.53	0.68	0.43	0.62
Top 2	K=3, 0.52	K=14, 0.75	0.47	0.62	0.33	0.62
Top 1	K=19, 0.44	K=19, 0.68	0.35	0.60	0.35	0.56

4.2.4 Analysis on Experiment Result

Figure 4.8 shows the comparison graph between electrode T7, all electrodes, top 20 features and top 1 features for binary classification. The evaluation metrics is rated with accuracy.

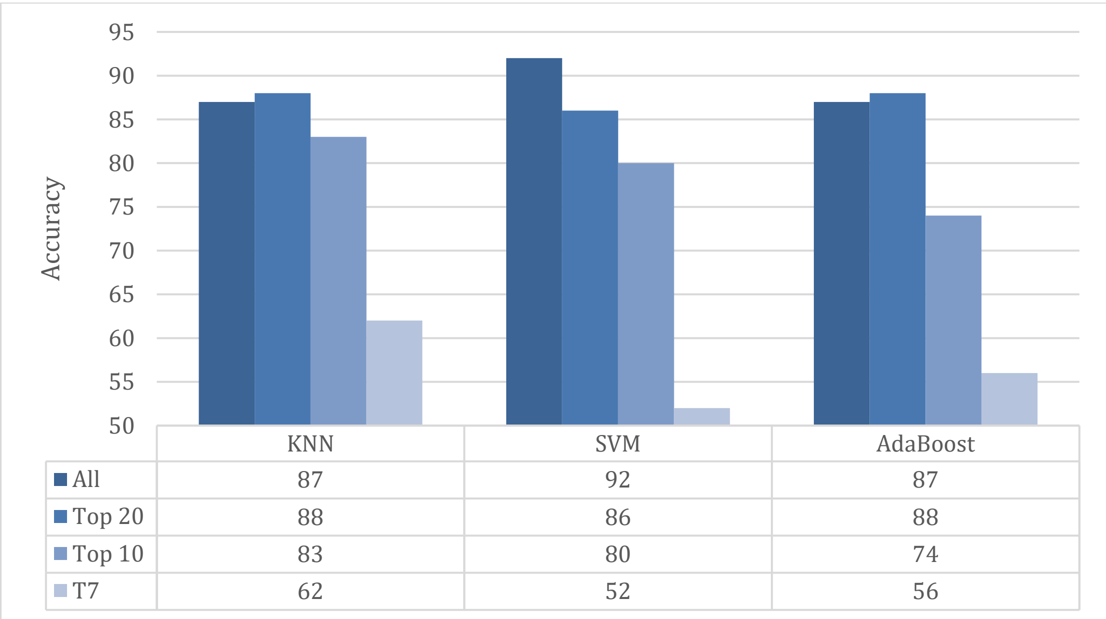
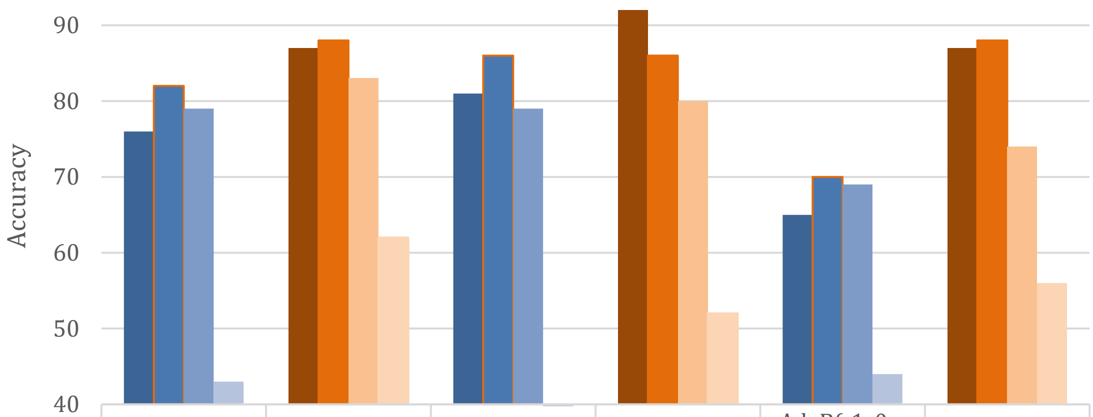


Figure 4.8 Comparison of performance using T7, all electrodes and SFS features for binary classification.

Based on **Figure 4.8**, the performance of input features with all electrodes, top 20 features and top 10 features are significantly increases compared to T7 input features only. SVM classifier achieved the highest accuracy of 92% in classifying all electrodes. The accuracy reflects that SVM is effective in binary classification will all data. It capable in capturing pattern of EEG signals from all electrodes. When using top 20 features selected through SVM model with SFS, it still able to achieve accuracy of 86%. It shows that SVM classifier is robust to classify class using 20 electrodes.

However, KNN classifier and AdaBoost classifier perform better with top 20 features as input. The achieved the same accuracy of 88% which is higher than using all electrodes. For top 10 features, KNN classifier well performs with accuracy of 83% compared to SVM 80% and AdaBoost 74%. The insight gain from the result is all models benefit from SFS by reducing the complexity of input data. For KNN and SVM classifier, the accuracy is slightly higher for top 20 features rather than using all electrodes for model training. Using all electrodes as input may consist of noises and irrelevant information in the EEG data. As a result, Sequential Forward Selection (SFS) significantly impacts each model with optimal features.

Figure 4.9 illustrates the model performance across different machine learning models. The blue block refers to multi-class classification (-1, 0, 1) which classify emotion states into negative, neutral or happy. The red block refers to binary classification (-1, 1) which classify emotion states into either positive or negative. The input feature is all electrodes, top 20 features, top 10 features and electrode T7.

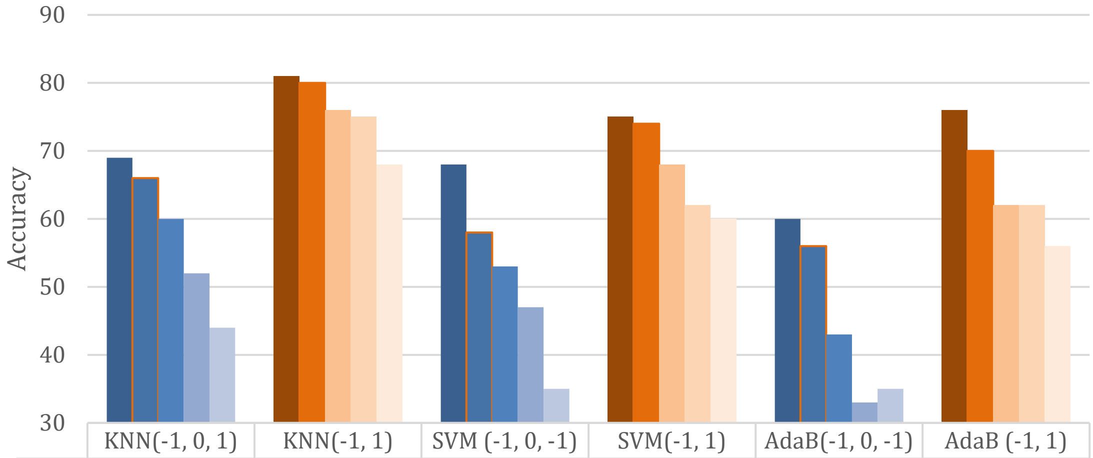


	KNN(-1, 0, 1)	KNN(-1, 1)	SVM (-1, 0, -1)	SVM(-1, 1)	AdaB(-1, 0, -1)	AdaB (-1, 1)
All	76	87	81	92	65	87
Top 20	82	88	86	86	70	88
Top 10	79	83	79	80	69	74
T7	43	62	39	52	44	56

Figure 4.9 Visualization performance between T7, all electrodes and SFS features for multi-class classification and binary classification across models.[Note: -1 refers to negative class, 0 refers to neutral class and 1 refers to positive class.]

Looking into multi-class classification, using the top 20 features from Sequential Forward Selection (SFS) achieved better than using all electrodes as input. Performance of top 20 features increases approximately 5% for all models compared to all electrode input. Among all classifiers, SVM models achieved the highest accuracy using top 20 features with accuracy of 86 %. For the top 10 features, KNN classifier and SVM classifier achieved the same accuracy of 79%, which approximates to 80%. Although multi-class classification uses 3 classes, KNN and SVM classifiers are still able to perform well with features selected through Sequential Forward Selection. In contrast, AdaBoost classifier is not performing well in multi-class classification. The reason may be the presence of neutral class adding complexity to the classification process.

Figure 4.10 illustrates the model performance across different machine learning models. The blue block refers to multi-class classification (-1, 0, 1) which classify emotion states into negative, neutral or happy. The red block refers to binary classification (-1, 1) which classify emotion states into either positive or negative. The input feature is top 1 feature until top 5 features.



	KNN(-1, 0, 1)	KNN(-1, 1)	SVM (-1, 0, -1)	SVM(-1, 1)	AdaB(-1, 0, -1)	AdaB (-1, 1)
Top 5	69	81	68	75	60	76
Top 4	66	80	58	74	56	70
Top 3	60	76	53	68	43	62
Top 2	52	75	47	62	33	62
Top 1	44	68	35	60	35	56

Figure 4.10 Visualization performance of SFS top 1 until top 5 features for multi-class classification and binary classification across models. [Note: -1 refers to negative class, 0 refers to neutral class and 1 refers to positive class]

Based on **Figure 4.10**, the performance is increasing from the initial 1 electrode to 5 electrodes. For multi-class classification, the performance from top 5 features to top 1 features is not ideal as all the accuracy is under 70%. It reflects that using 5 electrodes is not enough to classify between three classes as classifying between three classes is more complex. Therefore, it needs more electrodes to study the class pattern from the EEG signals. For binary classification, the performance is still acceptable for top 4 features across all models as its performance remains 70% and above. However, KNN classifier is still able to perform with top 2 and top 3 electrodes with accuracy of 75% and 76%. It shows that KNN classifier is able to perform even with fewer electrodes. Overall, a minimal number of electrodes is not enough for a classifier to differentiate between classes. It needs more electrodes to understand the underlying pattern to classify between classes.

4.2.5 Summary on Experiment Result

In summary, the classifier able to perform classification task with high performance. If considering using all electrodes, SVM is the most suitable classifier as it able to capture complex pattern from all electrodes. It achieved the highest accuracy in both multi-class classification and binary classification with accuracy of 81% and 92 %. For top 20 features, the best classifier also falls to SVM classifier as it able to get the average accuracy of 86% between multi-class classification and binary classification. Its ability shows that SVM is robustness in handling data from different classes by using top 20 features. For top 10 features, the best classifier is KNN classifier. It achieved accuracy of 79% for multi-class classification and 83% for binary classification. This shows that KNN classifier is effectively capture the underlying patterns even with 10 electrodes.

4.3 Limitation

One of the limitations of this study is the narrow age range of the participants. Based on information in [18], the participants are populating with 7 male and 8 female. The average age of the participants is 23.27 years old with standard deviation of 2.37. The estimated age range of the participants is between 21 to 26 years old. This indicates that the samples are primarily made up of young adults. It does not fully capture the diversity emotion across different age groups. For instances, EEG patterns from of adolescents and elderly could be different. According to [22], EEG patterns may affect due to aging-related factors such as brain development and cognitive decline. Therefore, the finding from this study may not be applicable for all age groups. Future studies across broader age groups are desirable to enhance EEG responses in emotion classification.

Besides, the limitation of this study is due to small sample size. In SEED dataset, the available sample size is 15 only. It may not fully discover the patterns for all classes. Although the sample size is small, it still able to perform well and referenceable for future studies. In this study, only one feature extraction is used, which is Differential Entropy (DE). More feature extractions method can be discovered in future to enhance the model performance.

Chapter 5
Conclusion & Future Work

5.1 Conclusion

Classification of human emotion is hard to identify. Traditional methods relying on facial expressions and questionnaires often fail in capturing human emotions. By leveraging Electroencephalogram (EEG) which capture brain activity, this research project is able to classify human emotions with high accuracy using machine learning algorithms. The model involves studies in feature extraction process for Gamma band features, energy computation such as Differential Entropy (DE), Sequential Forward Selection (SFS), machine learning models and performance analysis.

In this project, we aim to classify human emotions status accurately based on Electroencephalogram (EEG) signals using machine learning algorithms and evaluate performance of different machine learnings in emotion classification task. For classification, we used K-Nearest Neighbours (KNN) which takes votes on closest neighbours, Support Vector Machine (SVM) which optimal the hyperplane to classify data and Adaptive Boosting (AdaBoost) which combines of weak learners. Among all classifiers, SVM classifier achieved the highest performance when using all available electrodes. It achieved accuracy of 81% for multi-class classification and 92 % for binary classification. Besides all features, SVM also achieved the highest average accuracy in handling top 20 features selected by Sequential Forward Selection (SFS). When the features are reduced to 10 electrodes, KNN shows its ability to detect key patterns with fewer electrodes. It achieved 79% accuracy for multi-class and 83% for binary classification. For AdaBoost classifier, it did not perform well compared to KNN and SVM. It is struggled to handle complex EEG signals in emotion classification.

5.2 Future work

In future work, we plan to look into different ways to improve how we extract features from EEG signals to make emotion classification models work better. By trying out new and advanced techniques for capturing important information from EEG data, we hope to enhance the models' ability to identify patterns and boost their accuracy. Besides, we will be exploring more feature methods like Root Mean Square (RMS) and Power Spectral Density (PSD) to capture transient features over time. Moreover, we will explore additional machine learning models to enhance classification performance. This may involve using latest algorithms or combining multiple models to address the limitations of current approaches. The goal of this future work is to achieve a more reliable and high performance in classifying emotion classes.

REFERENCES

[1] S. Wioleta, “Using physiological signals for emotion recognition,” *2013 6th International Conference on Human System Interactions (HSI)*, 2013. doi:10.1109/hsi.2013.6577880

[2] M. Soufineyestani, D. Dowling, and A. Khan, “Electroencephalography (EEG) Technology Applications and Available Devices,” *Applied Sciences*, vol. 10, no. 21, p. 7453, Oct. 2020, doi: https://doi.org/10.3390/app10217453.

[3] C. S. Nayak and A. C. Anilkumar, “EEG Normal Waveforms,” *PubMed*, 2020. https://www.ncbi.nlm.nih.gov/books/NBK539805/#:~:text=However%2C%20the%20most%20frequently%20used

[4] Wei-Long Zheng and Bao-Liang Lu, “Investigating Critical Frequency Bands and Channels for EEG-Based Emotion Recognition with Deep Neural Networks,” IEEE Transactions on Autonomous Mental Development, vol. 7, no. 3, pp. 162–175, Sep. 2015, doi: https://doi.org/10.1109/tamd.2015.2431497.

[5] C. Rajdeep, Ensemble learning approach to motor-imagery EEG signal ... - researchgate, https://www.researchgate.net/profile/Rajdeep-Chatterjee-2/publication/329468472_Ensemble_Learning_Approach_to_Motor_Imagery_EEG_Signal_Classification/links/5d259fd1299bf1547ca77c33/Ensemble-Learning-Approach-to-Motor-Imagery-EEG-Signal-Classification.pdf

[6] JavaTPoint, “Support Vector Machine (SVM) Algorithm - Javatpoint,” *www.javatpoint.com*. https://www.javatpoint.com/machine-learning-support-vector-machine-algorithm

[7] JavaTpoint, “K-Nearest Neighbor(KNN) Algorithm for Machine Learning - Javatpoint,” *www.javatpoint.com*, 2021. https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning

[8] P. Siirtola, S. Tamminen, G. Chandra, A. Ihalapathirana, and J. Röning, “Predicting Emotion with Biosignals: A Comparison of Classification and Regression Models for

Estimating Valence and Arousal Level Using Wearable Sensors," *Sensors*, vol. 23, no. 3, p. 1598, Feb. 2023, doi: https://doi.org/10.3390/s23031598.

[9]Wei-Long Zheng and Bao-Liang Lu, "Investigating Critical Frequency Bands and Channels for EEG-Based Emotion Recognition with Deep Neural Networks," *IEEE Transactions on Autonomous Mental Development*, vol. 7, no. 3, pp. 162–175, Sep. 2015, doi: https://doi.org/10.1109/tamd.2015.2431497.

[10] R.-N. Duan, J.-Y. Zhu, and B.-L. Lu, "Differential entropy feature for EEG-based emotion classification," *IEEE Xplore*, Nov. 01, 2013. https://ieeexplore.ieee.org/document/6695876 (accessed Apr. 07, 2023).

[11] M. Luiza *et al.*, "Towards emotion recognition for virtual environments: an evaluation of eeg features on benchmark dataset," *Personal and Ubiquitous Computing*, vol. 21, no. 6, pp. 1003–1013, Dec. 2017, doi: https://doi.org/10.1007/s00779-017-1072-7.

[12] T. Sweet and D. E. Thompson, "Applying Big Transfer-based classifiers to the DEAP dataset," *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, Jul. 2022, doi: https://doi.org/10.1109/embc48229.2022.9871388.

[14] Rajamanickam Pon Nivedha, M. Brinda, Devika Vasanth, M Anvitha, and K. V. Suma, "EEG based emotion recognition using SVM and PSO," *International Conference on Intelligent Computing*, Jul. 2017, doi: https://doi.org/10.1109/icicict1.2017.8342809.

[15] S. T. George, M. S. P. Subathra, N. J. Sairamya, L. Susmitha, and M. Joel Premkumar, "Classification of epileptic EEG signals using PSO based artificial neural network and tunable-Q wavelet transform," *Biocybernetics and Biomedical Engineering*, vol. 40, no. 2, pp. 709–728, Apr. 2020, doi: https://doi.org/10.1016/j.bbe.2020.02.001.

[16] O. Bazgir, Z. Mohammadi, and S. A. H. Habibi, "Emotion Recognition with Machine Learning Using EEG Signals," *2018 25th National and 3rd International*

Iranian Conference on Biomedical Engineering (ICBME), Nov. 2018, doi: https://doi.org/10.1109/icbme.2018.8703559.

[17] M. Li, H. Xu, X. Liu, and S. Lu, "Emotion recognition from multichannel EEG signals using K-nearest neighbor classification," *Technology and Health Care*, vol. 26, pp. 509–519, Jul. 2018, doi: https://doi.org/10.3233/thc-174836.

[18] "SEED Dataset," *bcmi.sjtu.edu.cn*. https://bcmi.sjtu.edu.cn/home/seed/seed.html

[19] "Grid Search in Python - Javatpoint," www.javatpoint.com. https://www.javatpoint.com/grid-search-in-python (accessed Apr. 19, 2024).

[20] "Ada Boost algorithm in Machine Learning - Javatpoint," www.javatpoint.com. https://www.javatpoint.com/ada-boost-algorithm-in-machine-learning (accessed Apr. 19, 2024).

[21] GeeksforGeeks, "Sequential Feature Selection," GeeksforGeeks, https://www.geeksforgeeks.org/sequential-feature-selection/ (accessed Sep. 13, 2024).

[22] J.-H. Kang, J.-H. Bae, and Y.-J. Jeon, "Age-related characteristics of resting-state electroencephalographic signals and the corresponding analytic approaches: A Review," MDPI, https://www.mdpi.com/2306-5354/11/5/418 (accessed Sep. 13, 2024).

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S2	**Study week no.: 2**
Student Name & ID: Lam Yee Wei (21ACB00138)	
Supervisor: Dr Nur Syahirah Binti Roslan	
Project Title: EEG-Based Emotion Recognition using Machine Learning Algorithms	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Reading through FYP1 report and brainstorming for future work.

2. WORK TO BE DONE

Trimming data to specific duration for all samples.

3. PROBLEMS ENCOUNTERED

Time consuming to extract all electrodes within 4 minutes data.

4. SELF EVALUATION OF THE PROGRESS

-

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S2	**Study week no.: 4**
Student Name & ID: Lam Yee Wei (21ACB00138)	
Supervisor: Dr Nur Syahirah Binti Roslan	
Project Title: EEG-Based Emotion Recognition using Machine Learning Algorithms	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Complete data preprocessing to extract specific duration of data.

2. WORK TO BE DONE

Perform Sequential Forward Selection (SFS) for all electrodes.

3. PROBLEMS ENCOUNTERED

Get the optimal electrode combinations to enhance model performance.

4. SELF EVALUATION OF THE PROGRESS

Need time to study how Sequential Forward Selection works.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S2	**Study week no.: 6**
Student Name & ID: Lam Yee Wei (21ACB00138)	
Supervisor: Dr Nur Syahirah Binti Roslan	
Project Title: EEG-Based Emotion Recognition using Machine Learning Algorithms	

1. WORK DONE [Please write the details of the work done in the last fortnight.] Done partial code for Sequential Forward Selection (SFS)
2. WORK TO BE DONE Complete and make correction on coding part.
3. PROBLEMS ENCOUNTERED Misunderstanding of concept and caused incorrect implementation for Sequential Forward Selection (SFS).
4. SELF EVALUATION OF THE PROGRESS Slow progress and have to manage time properly.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S2	Study week no.: 8
Student Name & ID: Lam Yee Wei (21ACB00138)	
Supervisor: Dr Nur Syahirah Binti Roslan	
Project Title: EEG-Based Emotion Recognition using Machine Learning Algorithms	

1. WORK DONE
[Please write the details of the work done in the last fortnight.]

Done for Sequential Forward Selection (SFS) and get results.

2. WORK TO BE DONE

Perform multi-class classification using SFS selected features.

3. PROBLEMS ENCOUNTERED

Time-consuming to perform every input subset.

4. SELF EVALUATION OF THE PROGRESS

Have to evaluate and analyse model for multi-class classification.

Supervisor's signature   Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S2	**Study week no.: 10**
Student Name & ID: Lam Yee Wei (21ACB00138)	
Supervisor: Dr Nur Syahirah Binti Roslan	
Project Title: EEG-Based Emotion Recognition using Machine Learning Algorithms	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Completed for multi-class classification using SFS selected features.

2. WORK TO BE DONE

Perform binary class classification using SFS selected features.

3. PROBLEMS ENCOUNTERED

Time-consuming to perform every input subset.

4. SELF EVALUATION OF THE PROGRESS

Have to evaluate and analyse model for binary classification.

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S2	**Study week no.: 12**
Student Name & ID: Lam Yee Wei (21ACB00138)	
Supervisor: Dr Nur Syahirah Binti Roslan	
Project Title: EEG-Based Emotion Recognition using Machine Learning Algorithms	

1. WORK DONE

[Please write the details of the work done in the last fortnight.]

Partial analysis based on both multi-class classification and binary classification.

2. WORK TO BE DONE

Writing experiment result and complete the report.

3. PROBLEMS ENCOUNTERED

Analyse trends based on model performance across different inputs.

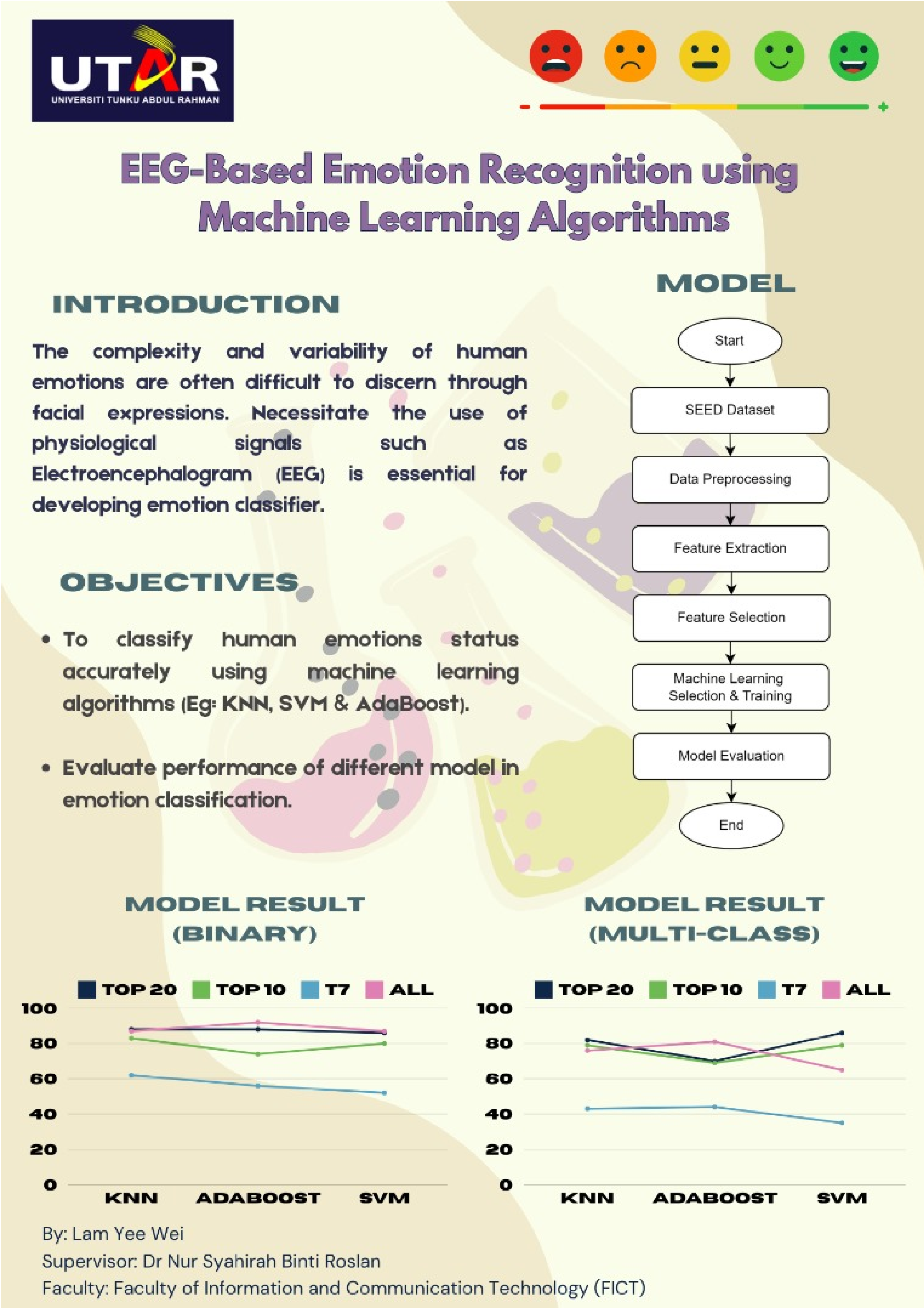
4. SELF EVALUATION OF THE PROGRESS

Moderate progress as other coursework do have coursework as well.

Supervisor's signature

Student's signature

POSTER


UTAR
UNIVERSITI TUNKU ABDUL RAHMAN
EEG-Based Emotion Recognition using Machine Learning Algorithms
INTRODUCTION
The complexity and variability of human emotions are often difficult to discern through facial expressions. Necessitate the use of physiological signals such as Electroencephalogram (EEG) is essential for developing emotion classifier.
OBJECTIVES
• To classify human emotions status accurately using machine learning algorithms (Eg: KNN, SVM & AdaBoost).
• Evaluate performance of different model in emotion classification.
MODEL
Start
SEED Dataset
Data Preprocessing
Feature Extraction
Feature Selection
Machine Learning Selection & Training
Model Evaluation
End
MODEL RESULT (BINARY)
TOP 20
TOP 10
T7
ALL
100
80
60
40
20
0
KNN
ADABOOST
SVM
MODEL RESULT (MULTI-CLASS)
TOP 20
TOP 10
T7
ALL
100
80
60
40
20
0
KNN
ADABOOST
SVM
By: Lam Yee Wei
Supervisor: Dr Nur Syahirah Binti Roslan
Faculty: Faculty of Information and Communication Technology (FICT)

PLAGIARISM CHECK RESULT

EEG-Based Emotion Recognition using Machine Learning Algorithms

ORIGINALITY REPORT

5%	4%	3%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.ijraset.com Internet Source	<1%
2	openaccess.altinbas.edu.tr Internet Source	<1%
3	Submitted to Universiti Tunku Abdul Rahman Student Paper	<1%
4	Submitted to University of Patras Student Paper	<1%
5	www.ncbi.nlm.nih.gov Internet Source	<1%
6	abis-files.aybu.edu.tr Internet Source	<1%
7	Submitted to Cardiff University Student Paper	<1%
8	ojs3.unpatti.ac.id Internet Source	<1%
9	Submitted to University of Aberdeen Student Paper	<1%

Universiti Tunku Abdul Rahman			
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	Lam Yee Wei
ID Number(s)	21ACB00138
Programme / Course	Bachelor of Computer Science
Title of Final Year Project	EEG-Based Emotion Recognition using Machine Learning Algorithms

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index: 5 % **Similarity by source** Internet Sources: 4 % Publications: 3 % Student Papers: 2 %	
Number of individual sources listed of more than 3% similarity: -	
Parameters of originality required and limits approved by UTAR are as Follows: **(i) Overall similarity index is 20% and below, and** **(ii) Matching of individual sources listed must be less than 3% each, and** **(iii) Matching texts in continuous block must not exceed 8 words** ***Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.***	

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

Signature of Supervisor	Signature of Co-Supervisor
Name: Dr Nur Syahirah Binti Roslan	Name: ________________
Date: 13/09/2024	Date: ________________

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY (KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	21ACB00138
Student Name	Lam Yee Wei
Supervisor Name	Dr Nur Syahirah Binti Roslan

TICK (√)	DOCUMENT ITEMS Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
√	Title Page
√	Signed Report Status Declaration Form
√	Signed FYP Thesis Submission Form
√	Signed form of the Declaration of Originality
√	Acknowledgement
√	Abstract
√	Table of Contents
√	List of Figures (if applicable)
√	List of Tables (if applicable)
	List of Symbols (if applicable)
√	List of Abbreviations (if applicable)
√	Chapters / Content
√	Bibliography (or References)
√	All references in bibliography are cited in the thesis, especially in the chapter of literature review
	Appendices (if applicable)
√	Weekly Log
√	Poster
√	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)
√	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report.

____

(Signature of Student)
Date: 13/09/2024