國立臺灣大學電機資訊學院生醫電子與資訊學研究所

碩士論文

Graduate Institute of Biomedical Electronics and Bioinformatics
College of Electrical Engineering and Computer Science
National Taiwan University

Master Thesis

基於注意力機制之時間序列原型卷積神經網路與傳統及量子機器學習模型應用於重度憂鬱症腦波之經顱磁刺激抗憂鬱療效預測與分析

EEG Analysis for Prediction of Antidepressant Responses of Transcranial Magnetic Stimulation in Major Depressive Disorder Based on Attentional Convolution Time Series Prototypical Neural Network Model and Classical/Quantum Machine Learning Approaches

陳麒升

Chi-Sheng Chen

指導教授：陳中平 博士

共同指導：李正達 博士

Advisor: Chung-Ping Chen, Ph.D.

Co-Supervisor: Cheng-Ta Li, M.D., Ph.D.

中華民國 110 年 6 月

June, 2021

doi:10.6342/NTU202101201

doi:10.6342/NTU202101201

國立臺灣大學碩士學位論文
口試委員會審定書

基於注意力機制之時間序列原型卷積神經網路與傳統及量子機器學習模型應用於重度憂鬱症腦波之經顱磁刺激抗憂鬱療效預測與分析

EEG Analysis for Prediction of Antidepressant Responses of Transcranial Magnetic Stimulation in Major Depressive Disorder Based on Attentional Convolution Time Series Prototypical Neural Network Model and Classical/Quantum Machine Learning Approaches

本論文係陳麒升君（學號 R08945009）在國立臺灣大學生醫電子與資訊學研究所完成之碩士學位論文，於民國 110 年 06 月 16 日承下列考試委員審查通過及口試及格，特此證明

口試委員：

______________（指導教授）

______________ 黃聖傑

楊智傑

所　長： 張瑞峰

doi:10.6342/NTU202101201

誌謝

起著祝犁大淵獻，盡昭陽赤奮若。這兩年左右的時間使我從完全沒摸過人工智慧演算法的新手到能獨力完成這篇研究論文，一路上受到許多人的幫助，在此我想表達衷心的感謝。感謝我的指導教授陳中平教授與共同指導教授李正達醫師在這段期間用心的指導，對我的研究提供許多寶貴的建議、應用機器學習及深度學習演算法於臨床資料的機會與在最佳化及憂鬱症等等工程與醫學專業知識上的指導，讓我學習與體會到何謂真正的跨領域學術研究。感謝臺北榮總精神部李醫師團隊中學長學姐們的大力協助，提供專業的醫學知識與腦波收案資料以及教我如何現場臨床收案、使用腦波儀與TMS。感謝各位口試委員願意百忙之中抽空來參加我的遠距口試以及遠端處理相關文件。感謝實驗室幫忙過我的學長和同學們，讓我有一個珍貴豐碩的研究所學習歷程。感謝我的鯊鯊抱枕陪伴著我度過無數個念書、debug、與自我懷疑的夜晚。最後感謝我的家人們一直以來對我的幫助與支持，使我能夠專心順利完成研究所的學業。

陳麒升

於BL405，2021年6月29日

doi:10.6342/NTU202101201

中文摘要

重度憂鬱症現今被認為是一種慢性惡化的精神疾病，並且具有與其他症狀甚至是自殺意念併發的風險。有一定比例的重度憂鬱症患者在嘗試過數種抗憂鬱藥物的治療過後並無顯著好轉，而此類型的病患被發現其有一定機率能被經顱磁刺激所治療。目前經顱磁刺激較為常見的類型有重複性經顱磁刺激與間歇性θ脈衝式經顱磁刺激兩種，而是否能為患者於臨床治療前預測各參數對各自病人的抗憂鬱反應並提供個人化精準有效的診療參數建議在未來將會是一個重要的技術。本研究利用129位重度憂鬱症患者的臨床腦電波資料訓練數種傳統與量子機器學習演算法並建構全新的深度學習模型來對兩種模式的經顱磁刺激療效進行訓練與預測，尤其針對幾乎沒有前人研究的間歇性θ脈衝式經顱磁刺激進行療效預測，並在包含模型類型的選取、深度學習模型的損失函數等等地方盡最大努力下避免模型過擬合。在結果中，本研究所提出的新式注意力機制卷積時間序列深度學習模型在常見的深度學習模型中預測兩種經顱磁刺激療效的效果均為最好。傳統機器學習方法中則使用了集成模型中的套袋與助推兩類模型，本研究也提出了一個能夠優化模型敏感度的資料前處理演算法，其綜合效果也較前人的支持向量機效果好。最後本研究也在真實的超導量子電腦上訓練了量子機器學習模型，並由結果證明其效果比同種傳統演算法好上許多。

關鍵字：重度憂鬱症、腦電圖、經顱磁刺激、機器學習、深度學習、量子計算

doi:10.6342/NTU202101201

ABSTRACT

Major depressive disorder (MDD) is a chronic worsening mental illness with high comorbidity and there is a risk of complications with other symptoms or even suicidal ideation. A certain proportion of patients with severe depression have not significantly improved after trying several antidepressant treatments, and this type of patients has been found to have a certain chance of being treated by transcranial magnetic stimulation. At present, the more common types of transcranial magnetic stimulation are the repetitive transcranial magnetic stimulation and the intermittent theta-burst transcranial magnetic stimulation. Therefore, whether it can predict the antidepressant response of each parameter to each patient before clinical treatment and provide personalized, accurate and effective diagnosis and treatment parameter recommendations will be an important technology in the future. This study used the clinical electroencephalography (EEG) data of 129 patients with MDD to train several traditional and quantum machine learning algorithms and construct a new deep learning model to train and predict the efficacy of two modes of transcranial magnetic stimulation. Especially for the intermittent theta-burst transcranial magnetic stimulation, which is hardly studied by previous studies, the curative

doi:10.6342/NTU202101201

effect is predicted, and the model type selection, the loss function of the deep learning model, etc. are included in the best efforts to avoid the model overfitting. In the results, the new attention mechanism convolutional time series prototypical deep learning model (ACTSNet) proposed in this research has the best effect in predicting the efficacy of the two types of transcranial magnetic stimulation among the common deep learning models. Traditional machine learning methods use the bagging and boosting models. This research also proposes a data pre-processing algorithm, booster transformation, that can optimize the sensitivity of the model, and its comprehensive effect is also better than that of previous support vector machines (SVMs). Finally, this research also trained a quantum machine learning model on a real superconductor quantum computer, and the results proved that its effect is much better than the same traditional algorithm.

Keywords: Major depressive disorder, Electroencephalography, Transcranial magnetic stimulation, Machine learning, Deep learning, Quantum computing

CONTENTS

doi:10.6342/NTU202101201

doi:10.6342/NTU202101201

doi:10.6342/NTU202101201

doi:10.6342/NTU202101201

LIST OF FIGURES

doi:10.6342/NTU202101201

doi:10.6342/NTU202101201

doi:10.6342/NTU202101201

LIST OF TABLES

doi:10.6342/NTU202101201

doi:10.6342/NTU202101201

doi:10.6342/NTU202101201

Chapter 1 Introduction

1.1 Thesis Motivation

Previous study in our team, by using a linear approach [29, 30], we discovered that rostral anterior cingulate cortex (rACC)-engaged cognitive task (RECT)-modulated frontal theta (the detail described in section 2.1) more accurately represents rACC behavior than non-modulated frontal theta. Recent studies in our team [31], we had found that RECT-modulated frontal theta could predict the antidepressant response for the patients with rTMS, but not iTBS, treatment by linear and nonlinear methods. However, the dataset was too small $(N_{rTMS} = 32,\ N_{iTBS} = 30)$, and the previous work only use support vector machine (SVM) [100] to predict well only on the rTMS patients but the iTBS. Having said that, a well-known problem of SVM that for a finite number of samples the minimization of SVM’s expected risk function alone does not lead to a good prediction model [113], thus SVM does not perform very well when the data set has more noise (i.e., target classes are overlapping) [113]. Because of the unique characteristics of omics data, such as the limited number of samples and large number of variables, an SVM classifier's overfitting in classification (the detail described in section 1.4.3) is technically increased, potentially leading to misleading diagnostic results [114]. In this thesis

doi:10.6342/NTU202101201

we tried to find the good models for iTBS response prediction. Using the SVM model saved from [31] to make prediction on the new dataset, we found the model has overfitting problem on it, the antidepressant response could not be completely predicted.

In this study, we used several classical machine learning and deep learning approaches with larger dataset $(N_{rTMS} = 62,\ N_{iTBS} = 67)$, modified both on increasing data and more complex, good generalization (anti-overfitting) ability algorithms to make the better performance models than SVM to predict the rTMS and iTBS response by EEG data correctly before treatment.

In addition, quantum machine learning would be also applied on the treatment response prediction. The quantum machine learning approaches can embed the classical data as the Hamiltonian operators into the Hilbert space which may can let the same model learn more efficient than its classical version [105] and predicted the treatment response more accurately.

1.2 Major Depressive Disorder (MDD)

Major depressive disorder (MDD), also known as depression, is becoming more widely understood as a chronic, deteriorating condition with a high risk of comorbidity. MDD symptoms that persist can lead to worsening outcomes, such as

doi:10.6342/NTU202101201

higher relapse rates, suicidality [1], a lower quality of life, and poor psychosocial functioning [2, 3, 4]. Low mood for at least two weeks, lack of confidence, changes in sleep habits, fatigue, increased or decreased appetite due to unexplained weight improvement, feelings of loss of hope or extreme pain, reduction in concentration, hesitancy, thoughts of death, and so on are all clinical signs of depression. MDD has a major impact on the patient's brain development and quality of life. As a consequence, how to handle and avoid MDD has become a global concern [5, 6, 7, 8, 16]. MDD is typically diagnosed based on clinician judgments of depression scales such as the Statistical Manual on Mental Disorders, fifth edition (DSM-V) [9]. Electroencephalography (EEG), positron emission tomography (PET), and magnetic resonance imaging (MRI) have also been used in recent research to try to identify an objective biomarker to diagnose MDD (MRI). EEG is one of them, and because of its low cost and ease of recording, it is a valuable tool for evaluating MDD.

1.3 Advanced Treatments for MDD Patients

Because of their low cost and wide supply, antidepressant medications are now the first line of therapy for MDD patients. However, due to the high heterogeneity of MDD patients [10], about 33 % of them have a negative reaction to several antidepressants [10, 11]. As a result, therapies such as electroconvulsive therapy

doi:10.6342/NTU202101201

(ECT), repetitive transcranial magnetic stimulation (rTMS), intermittent theta-burst stimulation (iTBS), and others exist in addition to medications. rTMS and iTBS use various transcranial magnetic stimulation (TMS) parameters. The antidepressant reaction of rTMS and iTBS therapies was the subject of this study.

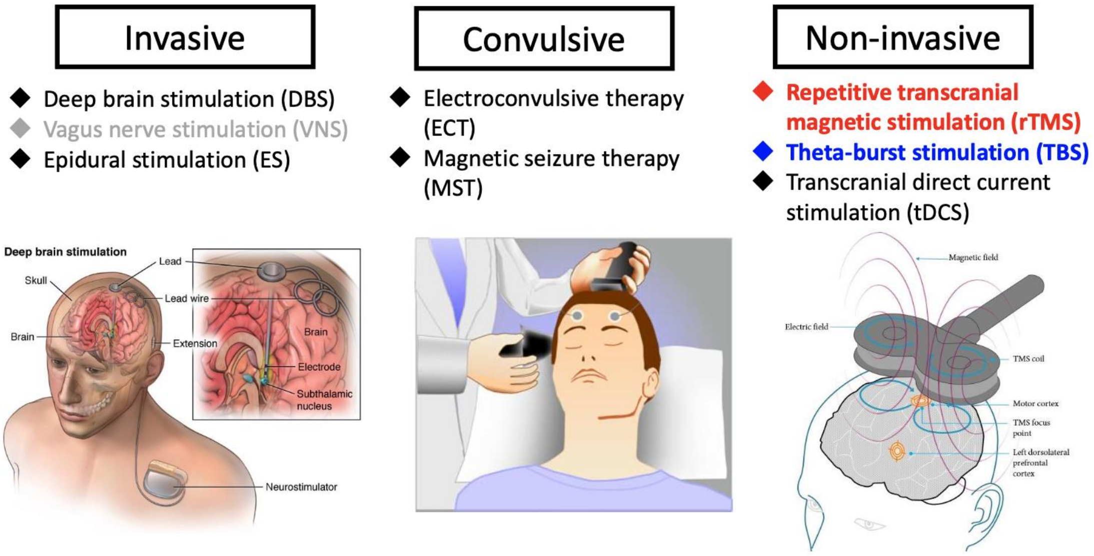

Invasive
◆ Deep brain stimulation (DBS)
◆ Vagus nerve stimulation (VNS)
◆ Epidural stimulation (ES)
Convulsive
◆ Electroconvulsive therapy (ECT)
◆ Magnetic seizure therapy (MST)
Non-invasive
◆ Repetitive transcranial magnetic stimulation (rTMS)
◆ Theta-burst stimulation (TBS)
◆ Transcranial direct current stimulation (tDCS)
Deep brain stimulation
Skull
Brain
Lead
Lead wire
Extension
Brain
Electrode
Subthalamic nucleus
Neurostimulator
Magnetic field
Electric field
TMS coil
Motor cortex
TMS focus point
Left dorsolateral prefrontal cortex

Figure 1.1: The brain stimulation treatments of MDD [11].

rTMS is a non-invasive, painless, and effective medication for depression that has been approved by the US Food and Drug Administration (FDA) and the Taiwan Food and Drug Administration (TFDA). Metal coils are used to directly emit heavy yet intermittent magnetic waves to various regions of the brain, causing a slight current to flow through the neural system. Recent research suggests that high-frequency (10Hz) rTMS applied to the left dorsolateral prefrontal cortex (DLPFC) might have a stronger antidepressant answer [13]. iTBS is a newer type of rTMS, which takes shorter stimulation time and generate a powerful effect, the previous researches in our team have shown that the efficiency of iTBS is no less than rTMS [14, 15] (Figure 1.3), and have proved the efficacy both on the rTMS and iTBS therapies.

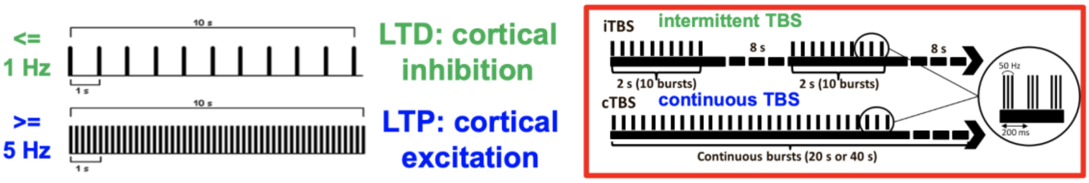


Figure 1.2: The protocol of rTMS (on the left side), and the protocol of iTBS (on the right side) [12].

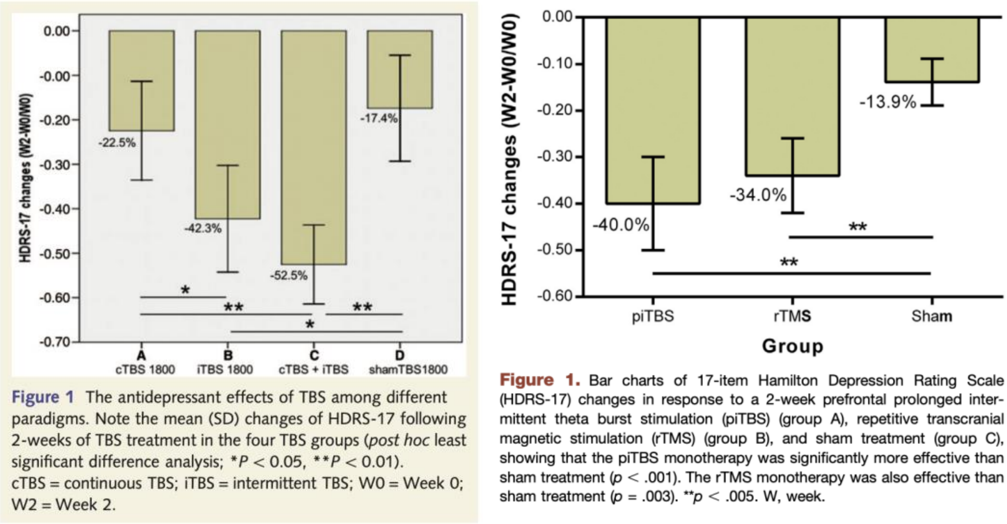


Figure 1 The antidepressant effects of TBS among different paradigms. Note the mean (SD) changes of HDRS-17 following 2-weeks of TBS treatment in the four TBS groups (*post hoc* least significant difference analysis; *$P < 0.05$, **$P < 0.01$). cTBS = continuous TBS; iTBS = intermittent TBS; W0 = Week 0; W2 = Week 2.

Figure 1. Bar charts of 17-item Hamilton Depression Rating Scale (HDRS-17) changes in response to a 2-week prefrontal prolonged intermittent theta burst stimulation (piTBS) (group A), repetitive transcranial magnetic stimulation (rTMS) (group B), and sham treatment (group C), showing that the piTBS monotherapy was significantly more effective than sham treatment ($p < .001$). The rTMS monotherapy was also effective than sham treatment ($p = .003$). **$p < .005$. W, week.

Figure 1.3: The clinical proof of the rTMS and iTBS on MDD [14, 15].

doi:10.6342/NTU202101201

1.4 Electroencephalogram

1.4.1 Prediction of TMS Response by EEG

Executive dysfunctions are linked to DLPFC in MDD patients. In the human brain, the DLPFC connects mental and cognitive functions [17]. During a depressive episode, the DLPFC and anterior cingulate cortex (ACC) can also transmit irregular signals from the amygdala [18]. The ACC is a part of the brain that integrates affective and attentional information, and it's particularly associated with attention dysfunctions in MDD patients [19, 20]. ACC is connected to prefrontal cortex (PFC) and amygdala, also acts as a bridge between attention and emotion, rostral anterior cingulate cortex (rACC) activity has been considered as a reliable biomarker for the antidepressant response [21]. Increased rACC activity may result in improved antidepressant responses in MDD patients [22]. Furthermore, an analysis found that the frontal theta EEG signal during corresponding mental activities indicates activity in the medial PFC and ACC [22, 23]. Theta waves are associated with frontal midline (ACC and PFC) cognitive control processes, as well as constructive and reactive cognitive control processes in the brain [24, 25]. PFC rTMS modulates not only geographical but also deeper regions, according to an analysis. (e.g., ACC, temporal cortex, and so on.) [26]. In conclusion, according to the previous study above, we can

doi:10.6342/NTU202101201

use MDD EEG data to predict the response of the rTMS treatments [27]. However, the central mechanism of iTBS had been found to be different from that of rTMS [115], and involve functional changes of frontal cortex (i.e., fronto-cingulate circuit) [115]. So, the RECT-related program may not be optimal for prediction of iTBS antidepressant responses, but frontal signals may include useful information for the prediction of it.

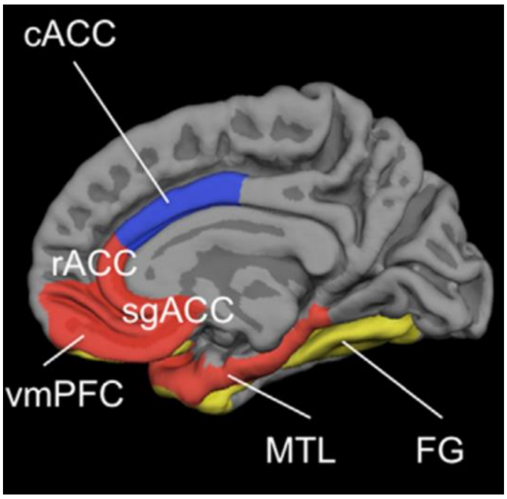


Figure 1.4: Illustration of rACC [28].

1.4.2 Prediction of TMS Response by Artificial Intelligence with EEG Data

Artificial Intelligence include the several types of learning algorithms like machine learning, deep learning, etc. (Figure 1.5). Machine learning technology has become widely used since 21th century and fitted in human daily life in many aspects. A key

doi:10.6342/NTU202101201

to successful process of machine learning is the feature extraction from raw data before training the model. However, there was limitation in the traditional algorithms for us to design a universal feature extractor to recognize various complicated patterns such as natural language, image or time series. Fortunately, the development of deep learning technology helps us solve the problem. Deep learning architecture is composed of numerous perceptrons, which act like the neurons in human nervous system. The perceptrons are arranged in multiple layers to build a complex neural network. Each layer carries abundant parameters, which can be regulated at every iteration during training neural network. The key aspect of deep learning is that the computer can adjust the parameters itself without human's aid. Whenever an input is inserted to the module, a loss function is carried out by calculating the error between the output and the true result. Recently, most practitioners often use some optimization algorithms like stochastic gradient descent (SGD) method [127] to minimize the loss function to improve the model. Another procedure called backpropagation [75] is used to adjust parameters in the multilayer network from output layers backward to input layers. And these properties may can apply on rTMS and iTBS response prediction to get better accuracy by MDD EEG data with model-computed nonlinear features.

doi:10.6342/NTU202101201

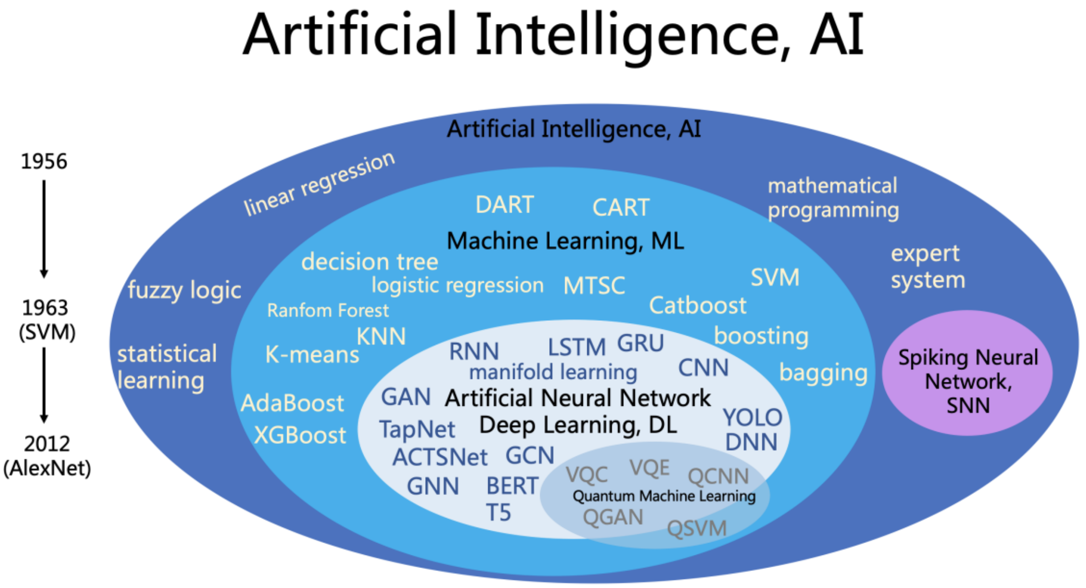


Figure 1.5: The rough map of artificial intelligence (AI).

1.4.3 The Overfitting Problem on SVM

Overfitting problem occurs when a learning algorithm (the classifier) fails the ability to generalize its learning and generates false diagnostic outcomes. It occurs when a model or algorithm that violates Occam's razor principle is used [118]: when too many parameters are introduced compared to the data set, or when a model that is too complex compared to the data set is used. It could get some positive diagnostic results on some training data, but it won't be able to generalize that skill to new test data. In other words, diagnostic findings are limited to a few unique data points rather than a broad range of information. In statistical learning theory, [119] proved that the probability of the test error distancing from an upper bound which is on data taken

doi:10.6342/NTU202101201

from the same independent and identically distributed (i.i.d.) distribution as the training set can be described as the following:

$$\Pr\left(\text{testing error} \leq \text{training error} + \sqrt{\frac{1}{N}\left[D\left(\log\left(\frac{2N}{D}\right)+1\right)-\log\left(\frac{\eta}{4}\right)\right]}\right) = 1-\eta$$

where D is the Vapnik-Chervonenkis (VC) dimension, a metric for the capability of a set of functions that can be learned using a statistical binary classification algorithm (complexity, expressive ability, richness, or flexibility) [120]. VC dimension defined as the cardinality of the largest set of data points that the algorithm can shatter, in other words VC dimension usually depends on the model complexity. N is the size of the training set, and $0 < \eta \leq 1$, is the probability of whether the equation is established. The mathematical rendition of overfitting is that when D is large, the testing error may be much higher than the training error. To describe the upper and lower bonds more clearly, we can do a simple calculate to define the Vapnik-Chervonenkis bond (VC bond) as the following:

$$\text{testing error} \leq \text{training error} + \sqrt{\frac{8}{N}\log(\frac{4(2N)^D}{\eta})}$$

in this lemma, the number of data point N or D is larger, the testing and training errors are closer. The VC bond can give the rough definition on overfitting of machine learning theoretically, but in real-world implements, the D and η are very hard to know, there are some more simple indexes can evaluate the rough trend of the generalization ability, e.g., the κ index [45, 46], and for the deep learning, there are also have some different theories tried to explain the overfitting mechanism and give the axiomatic approaches on neural network [121, 122, 123]. There have been several researches about the overfitting problems on SVM by the learning theory like risk bound, functional analysis, neural tangent kernel, margin bounds, generalization bounds, overparameterized models, and interpolated models by real analysis, etc. [113, 114, 116, 117, 124, 125]. In general, by only diagnosing a test sample as the plurality sort in the training results, the identity or isometric identity kernel matrix under the radial basis function (rbf) nonlinear kernel allows the SVM classifier to easily lose diagnostic capability [113, 114, 116]. In this thesis, we found that the previous SVM model [31] has the overfitting problem (described in the section 2.5), and we tried to find the larger D (more complex), the larger κ models and increase the size of dataset to prevent the models from overfitting.

doi:10.6342/NTU202101201

1.5 Thesis Organization

This section describes the structure of this thesis. The first and second chapters provide a basic introduction to major depressive disorder (MDD), the relationship between Rostral Anterior Cingulate Cortex (rACC)-Engaging Cognitive Task (RECT) and electroencephalogram (EEG), rTMS and iTBS therapy, the potential overfitting problem from the previous research, and some artificial intelligence approaches. The third chapter describe the EEG analysis pipeline, the deep learning model we modified, the classical and quantum machine learning approaches we used for this work. The computational experiment results and comparisons were shown in the fourth chapter. The fifth chapter summarizes the results of this thesis and the sixth chapter proposes the future work.

doi:10.6342/NTU202101201

Chapter 2 Previous Research

2.1 Rostral Anterior Cingulate Cortex (rACC)-Engaging Cognitive Task (RECT)

The frontal theta wave, on the other hand, can only partly indicate rACC behavior. In a previous study, our team discovered that using a linear approach, pre-treatment of modulated neural activity would better represent rACC activity than non-modulated activity [32] (Figure 2.2). The manipulating approach is computerized RECT which increases the rACC activity. The computerized RECT is a test designed according to the flexibility task of the Test for Attentional Performance [29]. The panel will begin to show a separate range of sharp and circular patterns on the left and right sides at random during the RECT test. Patients must locate the corresponding graphic using the rules we provided and click on it right away. The discriminating of sharp and round patterns has been known to improve prefrontal theta power, which is why graphs are shown in sharp and round patterns (Figure 2.1) [29]. A recent research conducted by our group discovered that using a nonlinear approach, RECT-modulated frontal theta could predict by logistic regression antidepressant reaction in patients receiving rTMS or iTBS therapy (Figure 2.3) [30].

doi:10.6342/NTU202101201

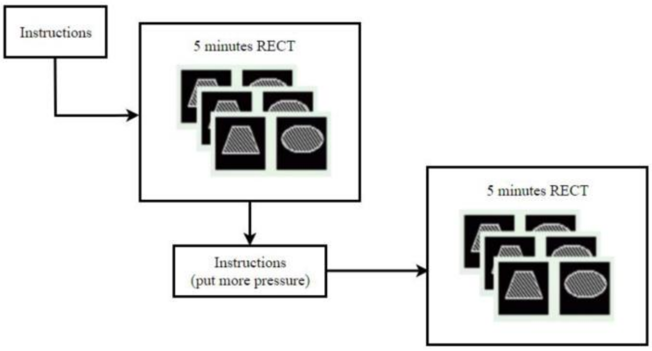


Figure 2. 1: The procedure of RECT [29].

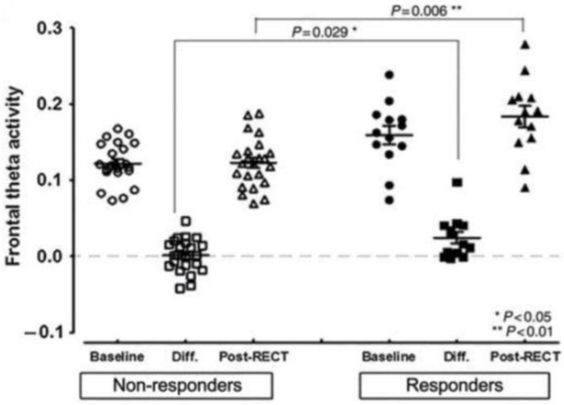


Figure 2. 2: RECT-modulated frontal theta more reflects rACC activity than non-modulated one [32].

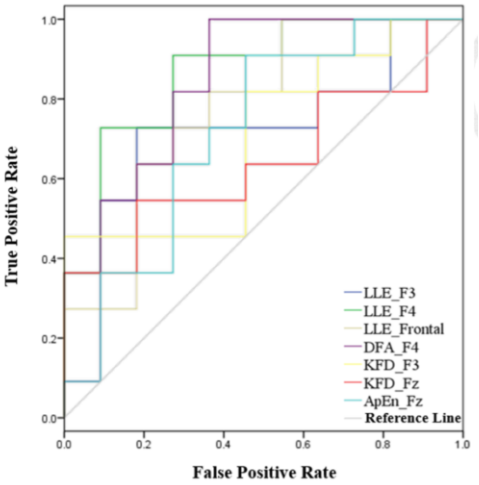


Figure 2. 3: The classification result of applying nonlinear features with logistic regression to RECT-modulated frontal theta [30].

2.2 Machine Learning on MDD EEG Data

There are several related researches on applying machine learning methods for EEG [56, 57, 58, 59], and to predict the response of rTMS treatment on MDD patients [33, 34, 35, 36, 37, 54], but the models they used are some relatively simple methods like k-nearest neighbor (KNN) [33] and support vector machine (SVM) [34, 35, 36, 37, 54] with their manually extracted features. Different from the popular machine learning for tabular feature set, in the field of machine learning for continuous time series data, there are the type of algorithm called multivariate time series classification (MTSC) [128, 129, 130] which do not need to extract the features manually, on the other words, MTSC can process continuous time series data end-to-end. At present, there are few studies on using bagging, boosting and MTSC machine learning methods to predict the response of rTMS. There is no research on using machine learning method to predict the response of iTBS by EEG data. Therefore, in this thesis, we used several machine learning methods like bagging, boosting and MTSC to train more powerful models to predict the response both on the rTMS and iTBS treatments.

2.3 Deep Learning on MDD EEG Data

With the development of deep learning, deep learning-based time series classification approaches have also made great progress in previous studies for rTMS response prediction [37, 38, 39, 40]. However, the deep learning model used by the previous work [37, 38, 39] are mostly simple artificial neural network (ANN) [37, 38], deep neural network (DNN) [31] and back propagation neural network (bpNN) [39] instead of more advanced deep learning models. Nowadays, the most research on MDD EEG deep learning models is used to discriminate between depressive and normal control [41, 42]. There is no research on using machine learning method to predict the response of iTBS by EEG data, either. Hence, in this thesis, we specially tailored for MDD EEG time series data a new deep learning model, ACTSNet, to predict the response both on the rTMS and iTBS treatments.

2.4 Quantum Machine Learning on Time Series Data

The concept of quantum computing originated from Richard Feynman [131], where the perspective of using a supervised quantum method to simulate various forms of Hamiltonians was discussed. Quantum machine learning (QML) [132] is an

doi:10.6342/NTU202101201

interdisciplinary discipline that incorporates quantum physics and machine learning and has exploded in popularity, drawing international interest in recent years. The fundamental issue in quantum machine learning is whether and how a quantum computer can improve existing machine learning methods [133]. In fact, the study of QML is more concerned with developing quantum algorithms that can solve machine learning problems more quickly than traditional methods such as classification tasks [103, 134, 135, 136, 137, 138], linear regression tasks [139, 140, 141, 142], clustering analysis [143, 144], dimensionality reduction tasks [134, 145, 146]. Because of the hardware technology technical limitations, humans do not apply the quantum machine learning algorithm on real-world data on the real quantum processer until recent years [43, 44, 168, 169, 170]. However, because of the high dimension of the EEG signal, many signal processing and machine learning techniques suffer from long processing and computation times. In order to achieve corresponding quantum speedup and embedding the features in the Hilbert space well by the quantum superposition and entanglement properties to learn well than classical machine learning, this thesis proposes a quantum mechanics-based method, quantum support vector machine (QSVM) [103], for classical features classification in the MDD EEG signal and run on the real IBM-Q quantum computer to predict the response both on the rTMS and iTBS treatments.

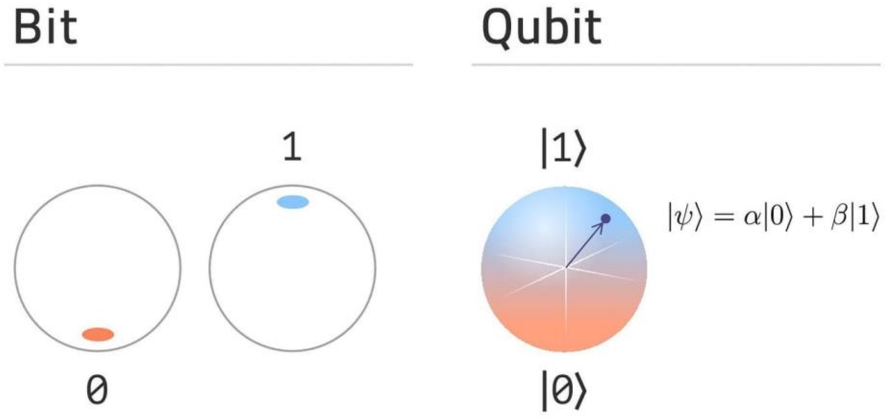


Figure 2.4: Illustration of classical bit and quantum bit (qubit) [126], where $\alpha^2 + \beta^2 = 1$.

2.5 Previous Work is Overfitting on the New Dataset

The previous study in our team [31] used SVM to predict the rTMS response with MDD EEG data, thus we do the validation with the model which saved from previous work [31], the validation result is shown in the Table 2.1.

We can see that the validation performance on the new dataset (OPD dataset, detailed in chapter 3.) is not well enough, and have 40.1% lower than its original validation result on the accuracy, 50% lower than its original validation result on the sensitivity, and 37.1% lower than its original validation result on the specificity. It shows obvious overfitting phenomena, the reason may be presumed to be too less data point to train well or the Vapnik-Chervonenkis (VC) dimension of the SVM model is too simple to represent and learn the complex representation from the MDD EEG data.

Biomarker	Frontal EEG signal	
Treatment	rTMS	
Patients (N)	32 (previous dataset)	31 (OPD dataset)
Features	Linear and Nonlinear features on Delta, Theta, Alpha, Beta, Gamma bands	
Classification	Previous work	Previous work saved model
	SVM	SVM
Accuracy	91.1%	**50.0%**
Sensitivity	83.3%	**36.3%**
Specificity	95.0%	**57.9%**

Table 2. 1: The validation result of SVM model saved form previous work [31].

2.6 Bagging/Boosting is Less Overfitting than SVM

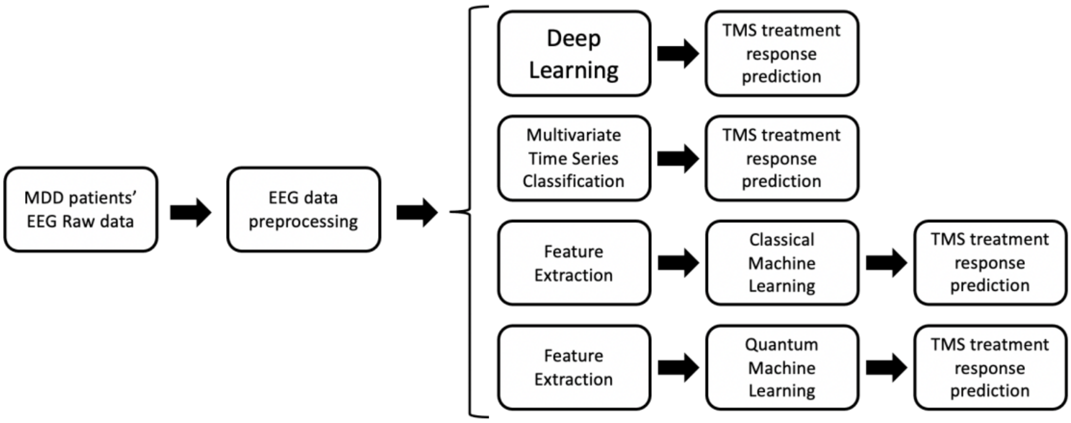
To solve the problem in section 2.5, we need to select the efficient methods better than SVM from many of machine learning approaches. According to [45, 46], the kappa index is bigger, the trend of model’s anti-overfitting (generalization) ability is better. And in order to hope to get more explainability on machine learning model, combining the above reasons above, we choosed the bagging (random forest) and boosting (XGBoost, and CatBoost) machine learning methods which can output the feature importance of the models to let us know which feature has larger weight in the datasets. Bagging/boosting is better than SVM on many datasets [47, 48, 49, 50, 51]. Furthermore, there are many studies found that neural network based deep learning algorithms have better performance than SVM on various open datasets, and in TapNet [77], it provides a skill called prototypical learning [147, 148] which can let neural networks train well by small time series dataset and prevent the overfitting, thus in the deep learning part in this thesis I invented a new model inspired by TapNet to let the model train well on the MDD EEG dataset.

2.7 Thesis Aims and Hypothesis

Previous work's model [31] is overfitting on the new OPD EEG dataset, and the previous work [30,31] did not find the good performance model for iTBS prediction task. We chose the bagging and boosting machine learning models because the characteristics of the model based on residual training make boosting better than SVM [50, 51], and the bagging model characteristics based on fusing different decision trees make it better than SVM [47, 52]. In this thesis we hypothesized that for the machine learning part we applied the larger D (more complex), the larger κ models include boosting, bagging, machine learning (ML) methods and increase the data size. may do more accurate for rTMS and iTBS responder classification of the EEG bands than SVM did. For the deep learning part, we hypothesized that deep learning algorithm can predict the rTMS and iTBS responder and non-responder accurately [41, 42, 43], and deep learning algorithms have better robustness than classical machine learning algorithms have [51]. For the number of the training data, there are some deep learning based MDD EEG prediction works showed that deep learning can work well on MDD EEG data (N=13 [53] ~ 55 [54]), but may have potential overfitting problems [41, 42, 43] and in some researches, prototypical learning can solve this problem to a certain degree [147, 148], thus we hypothesized that prototype learning based deep learning (DL) methods may do more accurate for rTMS and iTBS responder classification of the EEG bands than SVM on the small dataset. Lastly, for the quantum machine learning, we hypothesized that QSVM can learn well than classical SVM on the same features of MDD EEG data by the more efficient information embedding properties of quantum feature map.

Chapter 3 Methodology

In this work, we designed the new methods to predict the responder and nonresponder of rTMS and iTBS from electroencephalography (EEG) by the deep learning, machine learning, and quantum machine learning algorithms. In the machine learning part, we both used the original machine learning methods and boostered machine learning methods, booster is the new transformation on the test data preprocessing that can shift the distribution of model performance we figured out.


MDD patients' EEG Raw data
EEG data preprocessing
Deep Learning
TMS treatment response prediction
Multivariate Time Series Classification
TMS treatment response prediction
Feature Extraction
Classical Machine Learning
TMS treatment response prediction
Feature Extraction
Quantum Machine Learning
TMS treatment response prediction

$Figure\ 3.1$: The overview of the methodology of this thesis.

doi:10.6342/NTU202101201

3.1 EEG Data Acquisition

3.1.1 Psychiatric Evaluations

Psychiatric evaluations are used for evaluating mood and somatic severity. Patients were taken 17-item Hamilton Depression Rating Scale (HDRS-17), Young Mania Rating Scale (YMRS), Clinical Global Index-Severity (CGI-S), Maudsley staging method (MSM), Depression and Somatic Symptoms Scale (DSSS), and Life stress scale before any treatment (W0, baseline). Among them, HDRS-17, YMRS, CGI-S and MSM were evaluated by the clinician; DSSS and Life stress scale were filled by patients. However, all patients were only evaluated by HDRS-17, YMRS, CGI-S and DSSS after the 2-week treatment (W2). The HDRS-17 is the most widely used depression scale. Higher ratings indicate a more serious case of depression. The severity of the disorder can be determined by comparing the scores before and after therapy. In this research, the differences in HDRS-17 scores (W0 vs. W2) are the subject of this study. Responders are those whose HDRS-17 ratings have decreased by more than 50% at the completion of the two-week therapies (W2).

3.1.2 EEG Data Acquisition

MDD patients' EEGs were recorded in a dimly lit, electrically shielded silent space. A standard 32-channel digital EEG cap (Quik-Cap) with Ag/AgCl electrodes was placed according to the international 10/20 system (Figure 3.2) [60]. The impedances of all EEG electrodes were held below 5 kΩ. The numbers "10" and "20" indicate that the distances between opposite electrodes are 10% and 20% of the overall front–back or right–left width of the skull, respectively. Neuroscan amplifiers (Nuamps) with Neuroscan 4.3 software or the Brain Products machine were used for EEG recording (each EEG signal for 5 minutes) while patients were seated in a comfortable arm-chair with eyes closed.

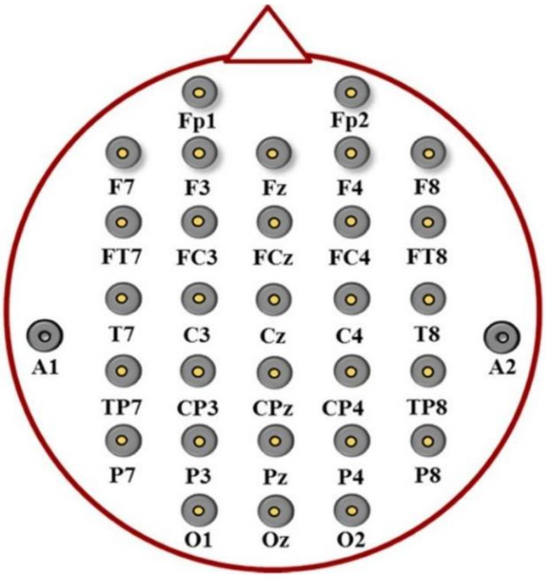


Figure 3.2: International 10/20 system of 32 channels placement [60].

doi:10.6342/NTU202101201

3.1.3 MDD EEG Datasets

In this thesis, we used two different datasets. One is called previous dataset or clinical trial dataset, which was collected for the previous work [31] to compare the different groups (rTMS, iTBS, and sham control) to get some findings. The other is called clinical outpatient (OPD) dataset, which was collected from the real-world clinical trials. In the both datasets, the patients with depression diagnosed by DSM-IV as major depressive disorder (MDD) have no major medical and surgical problems and other major mental illnesses (such as: bipolar mood disorders, schizophrenia, organic psychosis, substance use-related diseases... etc.) [15], and no response to at least one antidepressant treatment (that is, no more than 50% improvement in melancholy mood after treatment with 10-20 mg of escitalopram in terms of drug dose conversion for at least 8 weeks). The main differences between these two datasets are that the clinical trial dataset (previous dataset) need the depression symptoms of patients are at least 18 points on the 17-item Hamilton Depression Rating Scale (HDRS-17) scale, and the Clinical Global Impression Severity score is at least 4 points. The patients in the clinical trial dataset will be randomly assigned 1:1:1 to iTBS (80% active motor threshold, 1800 rounds), 10-Hz rTMS (100% resting motor threshold, 3000 rounds), or placebo sham group (half for iTBS, half for 10-Hz rTMS, but with sham coil, the

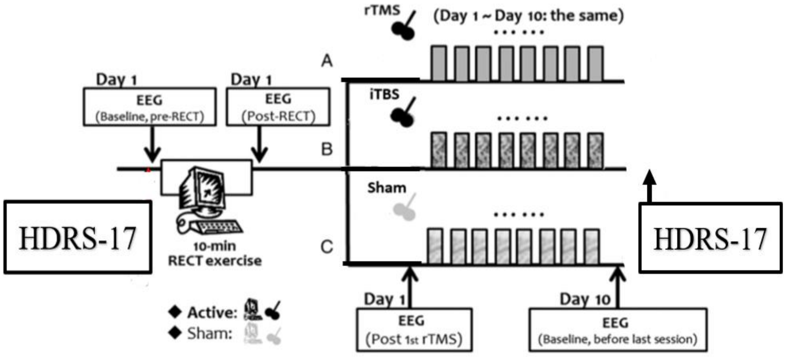
doi:10.6342/NTU202101201

patient will not receive real stimulation). Each patient will receive 10 brain nerve stimulation treatments once a day for 5 consecutive days and 2 consecutive weeks. In other words, the patients in the clinical OPD dataset there is no limit to the depressive symptoms of patients before treatment, but they must not be stable or consciously no longer depressed or HDRS-17 less than 7 points in order to be more widely used in clinics and does not have sham control group. All patients were received treatment of rTMS or iTBS, applied to left DLPFC, for two consecutive weeks (5 days/week). Parameters of rTMS is set to 10-Hz, 100% motor threshold, 4 seconds on and 26 second off, 40 times/session (3000 pulses), and 5 sessions/week; parameters of iTBS is set to 3-pulse 50Hz bursts given every 200 milliseconds at 5 Hz with at an intensity of 80% active motor threshold, and a 2 seconds train of bursts was repeated every 10 seconds for a total of 570 seconds (1800 pulses). For each patient, a 5-minute segment of EEG data will be recorded separately before and after a 10-minute RECT, referred to as the pre-RECT and post-RECT EEG signals, respectively. Patients received a 2-week (10-day) regimen with active rTMS, iTBS, or sham (sham control group is only in the previous dataset). All the EEG data of MDD patients were supported by Dr. C.T. Li, Functional Neuroimaging and Brain Stimulation Lab, Taipei Veterans General Hospital. The following sections will describe the details of these two datasets.

doi:10.6342/NTU202101201

3.1.3.1 Randomized Controlled Trial Data (RCTD) (Previous Dataset)

The EEG data were recorded from 90 patients (31 male and 59 female). Age from 22 to 72 years old. The dataset is used the same as the data in [31] in order to compare the model performance. Before rTMS, iTBS or sham treatments, all MDD patients were evaluated by 17-item Hamilton Depression Rating Scale (HDRS-17), Clinical Global Index (CGI) and Depression and Somatic Symptoms Scale (DSSS) at W0 (baseline). The other details of this dataset about the clinical subjects are in the previous work [31] (e.g., data exclusion criteria). The dataset is shown in Table 3.1.



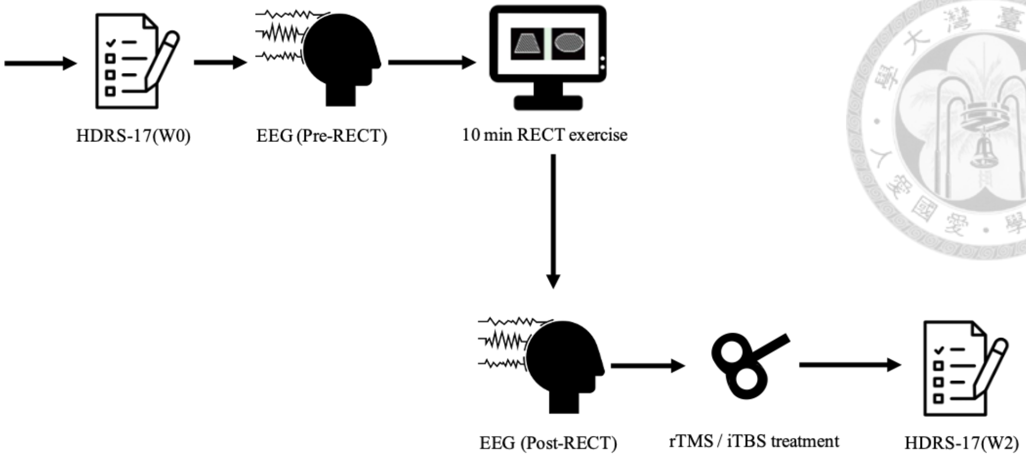
$Figure$ 3. 3: The clinical design in the previous study [31].

	rTMS (N=32)	iTBS (N=30)	Sham (N=28)	P-Value
Age (years)	46.5 (13.6)	47.1 (14.1)	47.2 (13.0)	0.994
Sex, male/female (N/N)	12/20	11/19	8/20	0.731
HDRS-17 (W0)	23.0 (3.9)	22.9 (3.5)	23.1 (3.6)	0.958
HDRS-17 (W2)	16.0 (6.6)	14.1 (6.8)	19.8 (6.0)	0.005
Responders, N (%)	11 (34.8%)	13 (43.3%)	1 (3.6%)	0.002

$Table$ 3.1: The detail of the RCTD (previous) dataset.

3.1.3.2 Clinical OPD dataset

The EEG data were recorded from 67 patients (27 male and 40 female). Age from 18 to 77 years old. We did exclusion of the MDD patients data with obsessive-compulsive disorder (OCD), dementia, fibromyalgia, and behavioral disorder (BD) complications to make the dataset more pure on MDD. Before rTMS, iTBS or sham treatments, all MDD patients were evaluated by 17-item Hamilton Depression Rating Scale (HDRS-17), Clinical Global Index (CGI) and Depression and Somatic Symptoms Scale (DSSS) at W0 (baseline). The dataset details are shown in Table 3.2.



$Figure$ 3.4: The clinical design in this study.

	rTMS (N=30)	iTBS (N=37)	P-Value
Age (years)	51.1 (17.5)	47.9 (18.1)	0.461
Sex, male/female (N/N)	10/20	17/20	0.299
HDRS-17 (W0)	20.0 (5.5)	20.5 (7.4)	0.746
HDRS-17 (W2)	12.6 (7.4)	11.5 (8.7)	0.579
Responders, N (%)	12 (40.0%)	20 (54.1%)	0.258

$Table$ 3.2: The detail of the OPD dataset.

	rTMS			**iTBS**		
	Previous (N=32)	**OPD (N=30)**	**P-Value**	**Previous (N=30)**	**OPD (N=37)**	**P-Value**
Age (years)	46.5 (13.6)	51.1 (17.5)	0.132	47.1 (14.1)	47.9 (18.1)	0.422
Sex, male/female	12/20	10/20	0.368	11/19	17/20	0.225
HDRS-17 (W0)	23.0 (3.9)	20.0 (5.5)	0.106	22.9 (3.5)	20.5 (7.4)	0.339
HDRS-17 (W2)	16.0 (6.6)	12.6 (7.4)	0.053	14.1 (6.8)	11.5 (8.7)	0.232
Responders, N (%)	11 (34.8%)	12 (40.0%)	0.327	13 (43.3%)	20 (54.1%)	0.195

$Table$ 3.3: The comparison between the datasets.

doi:10.6342/NTU202101201

The main different nature of these two datasets is that the previous dataset is the research used dataset, which heterogeneity is less than the real-word OPD dataset. To make more robust model with more wide distribution of MDD EEG data representation, we combined two datasets to train the machine learning models (detail described in chapter 4) if the premise conditions of machine environment permit.

3.2 EEG Data Preprocessing

The EEG raw data from the hospital have three datatypes due to different EEG recording machines made from the different companies, .cnt file (from Neuroscan), .edf file (European Data Format [61]) and .eeg/.vhdr/.vmrk files (from Brain Products). Moreover, EEG raw data contain several artifacts such as eye movement, eye blink, heart beats, muscle movement, and so on. Therefore, some preprocessing steps of EEG signals, including resampling, filtering, Independent Component Analysis (ICA) [62] methods were applied. The data preprocessing steps will be introduced in this section.

3.2.1 EEG Signal Resampling

The original raw data from all three types of data files are all 1,000 Hz and about 300 seconds per file, include too high sample rate to analysis (1000Hz) because the

doi:10.6342/NTU202101201

memory issue on the computer. The maximum frequency in this work, according to Nyquist frequency theorem [149], is about 60 Hz. As a consequence, to prevent aliasing effect, the sampling frequency must be greater than 120 Hz. To minimize computation time and memory, we resample the EEG data from 1,000 Hz to 250 Hz.

3.2.2 Band Pass Filter

The MDD EEG data is filtered by the band-pass finite-duration impulse response (FIR) filter with Hamming window. The low cut-off frequency is setting to 1 hertz because of the definition of delta band, and the high cut-off frequency is setting to 60 hertz for the frequency of the noise from direct current (DC) power line and the definition of gamma band.

3.2.3 Independent Component Analysis

First, after we got the data from hospitals, we filtered the raw data from 1Hz to 60Hz and then do Independent Component Analysis (ICA) [62] to remove the noise such as electrooculogram (EOG) and electromyography (EMG).

The independent component analysis (ICA) algorithm is shown below. Independent component analysis (ICA) [62, 63] is a powerful algorithm to remove the noise in

doi:10.6342/NTU202101201

signal. To rigorously define ICA, we can use a statistical “latent variables” model. Assume that we observe n linear mixtures $x_1, x_2, \ldots, x_n$ of n independent components like

$$x_i = a_{i_1} s_1 + a_{i_2} s_2 + \cdots + a_{i_n} s_n, \forall\, i.$$

In the ICA model, we assume that each mixture x_i and independent component s_k is a random variable, and the ICA model is for non gaussianity. Without loss of generality, we can assume that both the mixture variables and the independent components have zero mean. Then, we rewrite the ICA equation above into the vector-matrix form

$$\mathbf{x} = \mathbf{As}$$

Using this vector-matrix notation, it is equivalent to

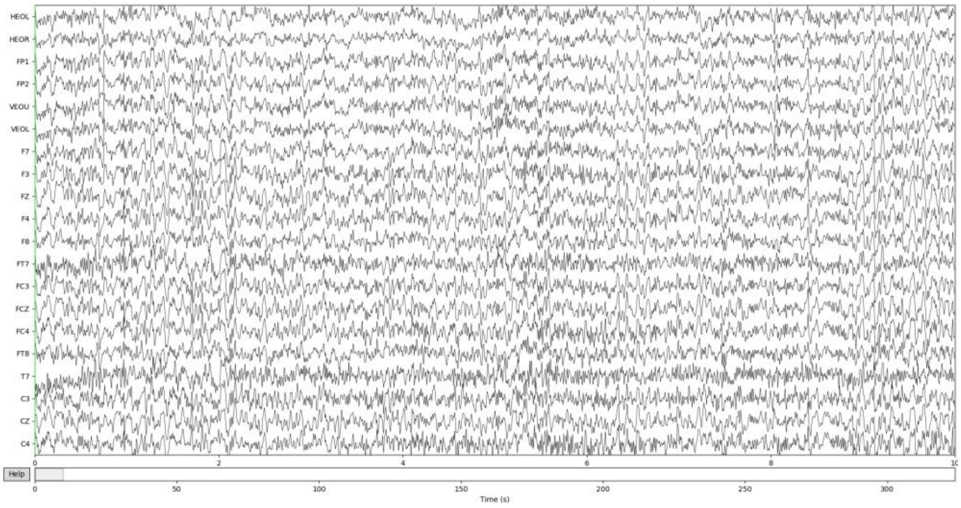
$$\boldsymbol{x} = \sum_{i=1}^{n} a_i\, s_i$$

The ICA model is a generative model, which means that it describes how the observed data are generated by a process of mixing the components s_i. The independent components are latent variables, meaning that they cannot be directly observed. The mixing matrix is assumed to be unknown. All we observe is the random vector $\boldsymbol{x}$, and we must estimate both **A** and **s** using it. This must be done under as general assumptions as possible.

doi:10.6342/NTU202101201

3.3 ACTSNet: Attentional Convolution Time Series Neural Network

In this thesis, the electroencephalography (EEG) data recorded from the major depressive disorder (MDD) patients are considered as time series data. Time series data are the sets of sequential numerical values by time. The EEG data are continuous time series, and multivariate time series classification (MTSC) and deep learning algorithms are good at this [64, 65, 66, 67].



$Figure$ 3. 5: The MDD EEG (after ICA).

Due to the well performance on image classification tasks, convolution-based neural network like convolutional neural network (CNN) [68], fully convolutional network (FCN) [69], multi-channel deep convolutional neural network (MCDCNN) [70], and residual network (ResNet) [71], etc. have been tried to use on time series classification tasks [66]. There are also time series data customized convolutional

doi:10.6342/NTU202101201

network models like Time Le-Net (TLENT) [72] which is inspired by the great performance of Le-Net’s architecture for the document recognition task, and InceptionTime [73] which is affected by the inception module in Inception Network [74]. In the field of deep learning, recurrent neural network (RNN, or Naïve RNN) [75], its improved version, long short-term memory (LSTM) [76], and the modifications of combining with CNN, LSTM-FCN [79] and MLSTM-FCN [80], are often used to process time series data.

All the experiments are run on three NVIDIA GeForce RTX™ 3090 graphic cards (graphic processing units, GPUs) with python 3.6+ (tensorflow, keras and pytorch frameworks) and CUDA 10.2+, the rough experiment flow is shown as below:

- □ 1. Read raw EEG data (.cnt/.edf/.eeg) and transform them into .npy for python processing.
- □ 2. Filter out frequencies other than 1Hz~60Hz.
- □ 3. Do independent component analysis (ICA) to remove the artifacts in EEG.
- □ 4. Separate alpha (α), beta (β), gamma (γ), delta (δ), theta (θ) sub-bands in FP1, FP2, F7, F3, Fz, F4, F8 electrodes by finite impulse response (FIR) filter.
- □ 5. Divide the data into train set and test set. Train the deep learning and multivariate time series classification classifiers models with train dataset.
- □ 6. Validation the model with test dataset (70:30), and modify the model till the

best performance.

□ 7. Test the model with real world data (e.g., clinical data), and modify the model till the best performance.

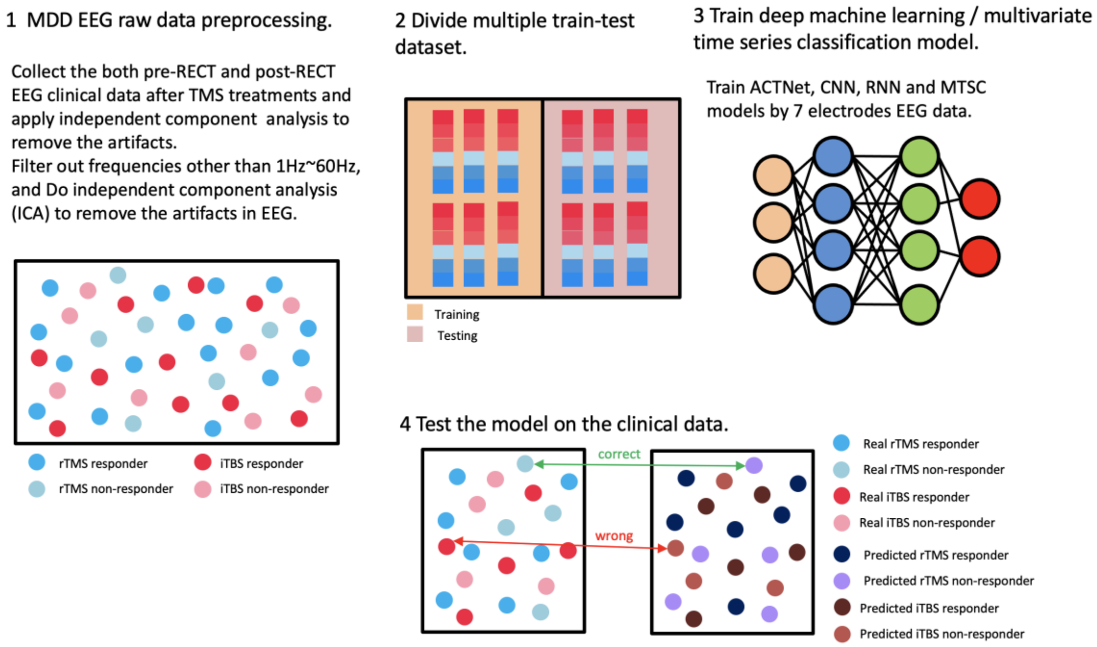

1 MDD EEG raw data preprocessing.
Collect the both pre-RECT and post-RECT EEG clinical data after TMS treatments and apply independent component analysis to remove the artifacts.
Filter out frequencies other than 1Hz~60Hz, and Do independent component analysis (ICA) to remove the artifacts in EEG.
rTMS responder
rTMS non-responder
iTBS responder
iTBS non-responder
2 Divide multiple train-test dataset.
Training
Testing
3 Train deep machine learning / multivariate time series classification model.
Train ACTNet, CNN, RNN and MTSC models by 7 electrodes EEG data.
4 Test the model on the clinical data.
correct
wrong
Real rTMS responder
Real rTMS non-responder
Real iTBS responder
Real iTBS non-responder
Predicted rTMS responder
Predicted rTMS non-responder
Predicted iTBS responder
Predicted iTBS non-responder

Figure 3.6: The methodology of deep learning and MTSC.

3.3.1 ACTSNet Architecture

At the time of writing this thesis, multivariate time series classification with attentional prototypical network (TapNet) [77] is one of the state-of-the-art (SOTA) deep learning model in time series classification task. It is an ensemble model of LSTM, CNN and attentional prototypical learning. By its prototypical learning architecture, it can train with small time series datasets and prevent overfitting, but it did not perform well on the noisy data [89, 90] like MDD EEG data for its LSTM architecture (experiment results are in the section 4.1). Hence, I modified TapNet to a new model I figured out, attentional convolution time series with attentional prototypical network (ACTSNet). ACTSNet changed the LSTM module to the attentional convolution neural network, this module is inspired by the neural network called Encoder [78]. The attentional convolution (AC) of tensor X is described as follows:

$$AC = Softmax(X)$$

After AC layer is a multiply layer (Mul) to concatenate the tensor values, is shown as the following:

$$Mul = AC \odot X$$

where $\odot$ is layer-wise tensor multiplication.

In this attentional convolution neural network model structure I used three 1D-concolution (Conv1D) layers, instance normolization (IN) [81], and parametric rectified linear unit (PReLU) [82, 83], attentional convolution (AC) and multiply layer (Mul) to let the ACTSNet on MDD EEG data classification task can perfrom better than original TapNet architecture. The detail architecture is shown as below:

$$\begin{aligned} ACTSNet_{AC} = \{&3 \times [Conv1D + (IN + PReLU)] + AC + Mul \\ &+ fully\ connected\ (FC) + (Sigmoid + IN) + global\ pooling\}. \end{aligned}$$

doi:10.6342/NTU202101201

3.3.2 Prototypical Learning

Traditional deep learning networks which have an inductive bias due to the small number of training samples in time series results [147]. By training the distance-based loss function, [77] suggest a novel attentional prototype learning process. [77] learn a class prototype embedding by the algorithm in [148], for each class, and classify the input time series according to their distance from each class's prototype. According to the [148], let $M_k = [m_1, \ldots, m_{|S_k|}] \in \mathbb{R}^{|S_k| \times d}$ be a matrix of time series embeddings belonging to the class k, where S_k is the set of indices for data samples with class label k. Then, as seen below, a weighted sum of individual sample embeddings vectors $\mathfrak{h}_k$ of class prototypes can be used to display the prototype embedding of class k, $\mathcal{W}_{k,i}$ is the trainable weight of i^{th} data in class k according to the embeddings of time series, and $M_{k,i}$ represents the embedding of the data sample.

$$\mathfrak{h}_k = \sum_i \mathcal{W}_{k,i} \cdot M_{k,i}$$

We treat time series samples from the same class as a bag of discrete instances in particular. For each instance, an attention-based multi-instance pooling metod [151] is applied to determine its instance weight for each class prototype. The attention weights for the k^{th} class is:

$$\mathcal{W}_k = Softmax(w_k^T \tanh(V_k M_k^T))$$

where $\mathscr{w}_k \in \mathbb{R}^{u \times 1}$ and $V_k \in \mathbb{R}^{u \times d}$ are trainable separate parameters for the model, the hypothesis that the different classes may have distinct attentions on their feature spaces, u is the size of hidden latent space dimension for $\mathscr{w}_k$ and V_k. The distribution over classes for a given time series $\boldsymbol{x} \in \mathbb{R}^d$ can be expressed as a softmax over distances to prototypes in the embedding space as the equation below:

$$\mathscr{p}_\Theta(y = k|\boldsymbol{x}) = \frac{\exp(-D(f_\Theta(\boldsymbol{x}), \boldsymbol{\mathfrak{h}}_k))}{\sum_i \exp(-D(f_\Theta(\boldsymbol{x}), \boldsymbol{\mathfrak{h}}_k))}$$

where the distance function $D : \mathbb{R}^d \times \mathbb{R}^d \mapsto [0, +\infty)$ measure the distances between two embedding vectors $\boldsymbol{\mathfrak{h}}_k$s. The distance function can be chosen from regular Bregman divergences [152]. In this thesis, we applied squared Euclidean distance:

$$D(\mathscr{j}, \mathscr{j}') = ||\mathscr{j} - \mathscr{j}'||^2$$

to measure the distances between two embedding vectors $\boldsymbol{\mathfrak{h}}_k$s. The correlation between the class template and the time series is used to calculate the probabilities over classes, thus we multiply -1 in front of the distance function like what the related work did in [77]. Then the training of our model can proceed by minimizing the negative log probability (the equation below) of the true class via the Adam algorithm [153].

$$\mathcal{L}(\Theta) = -\log(\mathscr{p}_\Theta(y = k|\boldsymbol{x}))$$

[150] shows that the efficay of the convolutional prototype learning (CPL) neural network can address the open world recognition issue well, resulting in increased

robustness. CPL improve the intra-class compactness of the feature representation, which can be viewed as a generative model based on the Gaussian assumption of different classes, thence, in this thesis we use these architectures to create a new model, ACTSNet, to imporve the performance on rTMS and iTBS response prediction tasks.

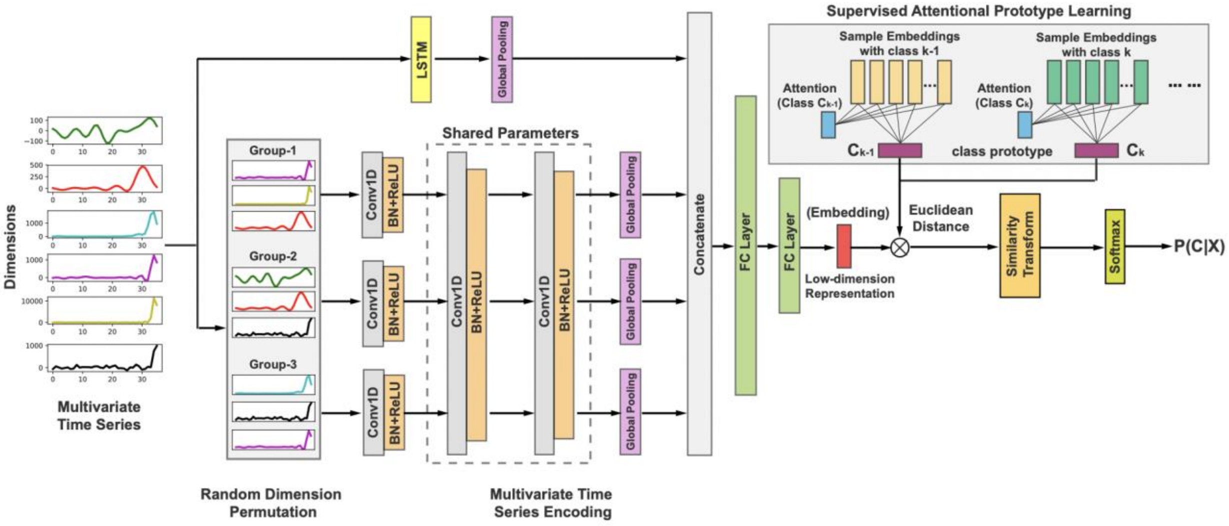


Figure 3.7: TapNet [77].

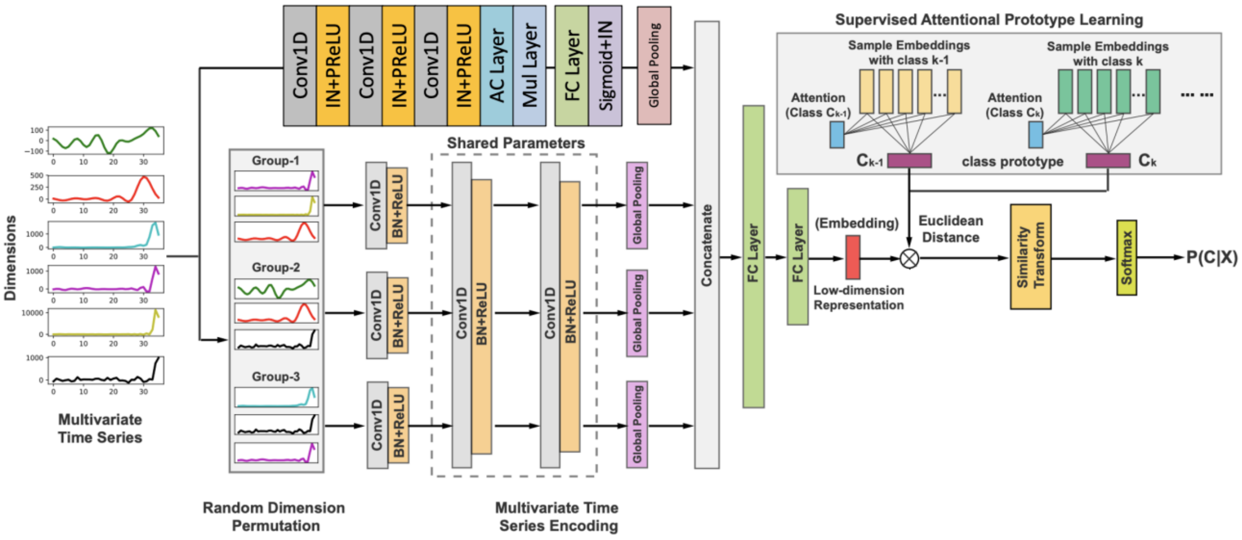


Figure 3.8: ACTSNet (My work).

doi:10.6342/NTU202101201

3.4 Multivariate Time Series Classification (MTSC) Methods

Multivariate time series classification (MTSC) stands for the type of machine learning approaches that do not need to extract features manually in this thesis, the model can deal with continuous time series data and make classification prediction end-to-end. MTSC algorithms is less popular than deep learning methods, but we still compared some MTSC algorithms with deep learning methods in this thesis.

3.4.1 Bag-of-SFA-Symbols Ensemble (BOSS Ensemble)

In this work, to compare the prediction results with the deep learning methods, we selected Bag-of-SFA-Symbols Ensemble (BOSS Ensemble) algorithm [84, 85] for its noise-resistant characteristics [86]. BOSS Ensemble is a dictionary-based MTSC machine learning approach, which combines symbolic Fourier approximation (SFA) [87] and bag-of-words (BoF) [88] methods to classified the time series.

doi:10.6342/NTU202101201

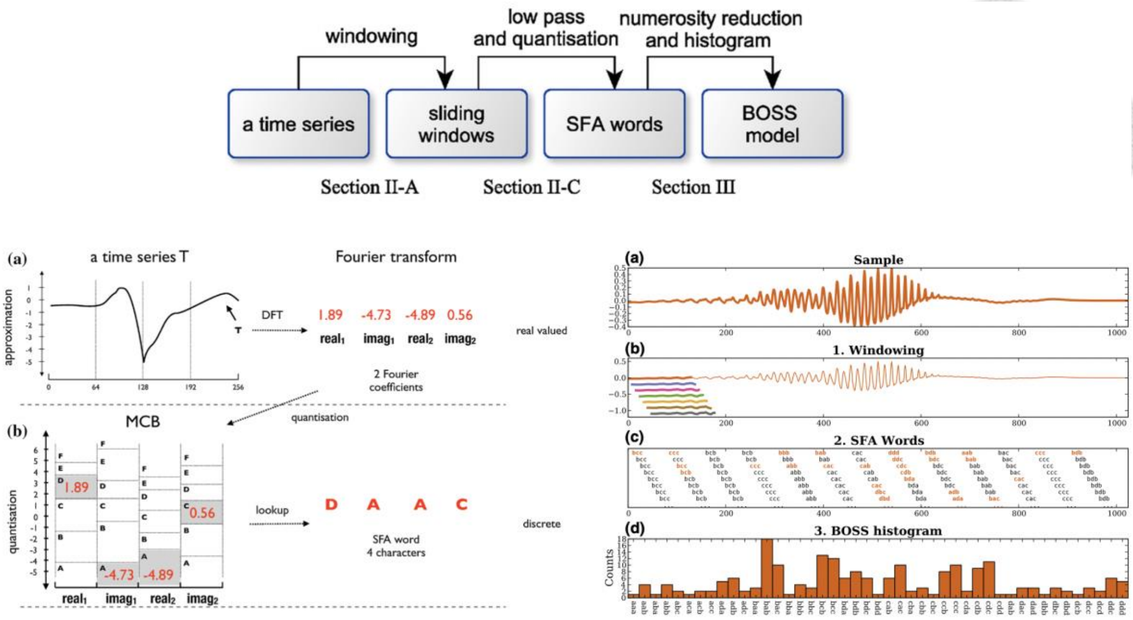


Figure 3.9: BOSS Ensemble [85].

3.5 Classical Machine Learning Methods

All the experiments are run on the computer with python 3.6+ (include scikit-learn framework), the statistical packages we used are scipy and pycm, the rough experiment flow is shown as the follows:

□ 1. Read raw EEG data (.cnt/.edf/.eeg) and transform them into .npy for python processing.

□ 2. Filter out frequencies other than 1Hz~60Hz.

□ 3. Do independent component analysis (ICA) to remove the artifacts in EEG.

□ 4. Separate alpha, beta, gamma, delta, theta sub-bands in FP1, FP2, F7, F3, Fz, F4, F8 electrodes by finite impulse response (FIR) filter.

doi:10.6342/NTU202101201

□ 5. Linear and nonlinear features extraction and feature selection.

□ 6. Divide the data into train set and test set randomly (70:30) for 10 times. Train the machine learning classifiers models with train dataset and average each accuracy for validation. (The code of feature extraction, selection and cross validation we used the same as it in [31] for the model performance comparison.)

□ 7. Validation the model with test dataset, and modify the model till the best performance.

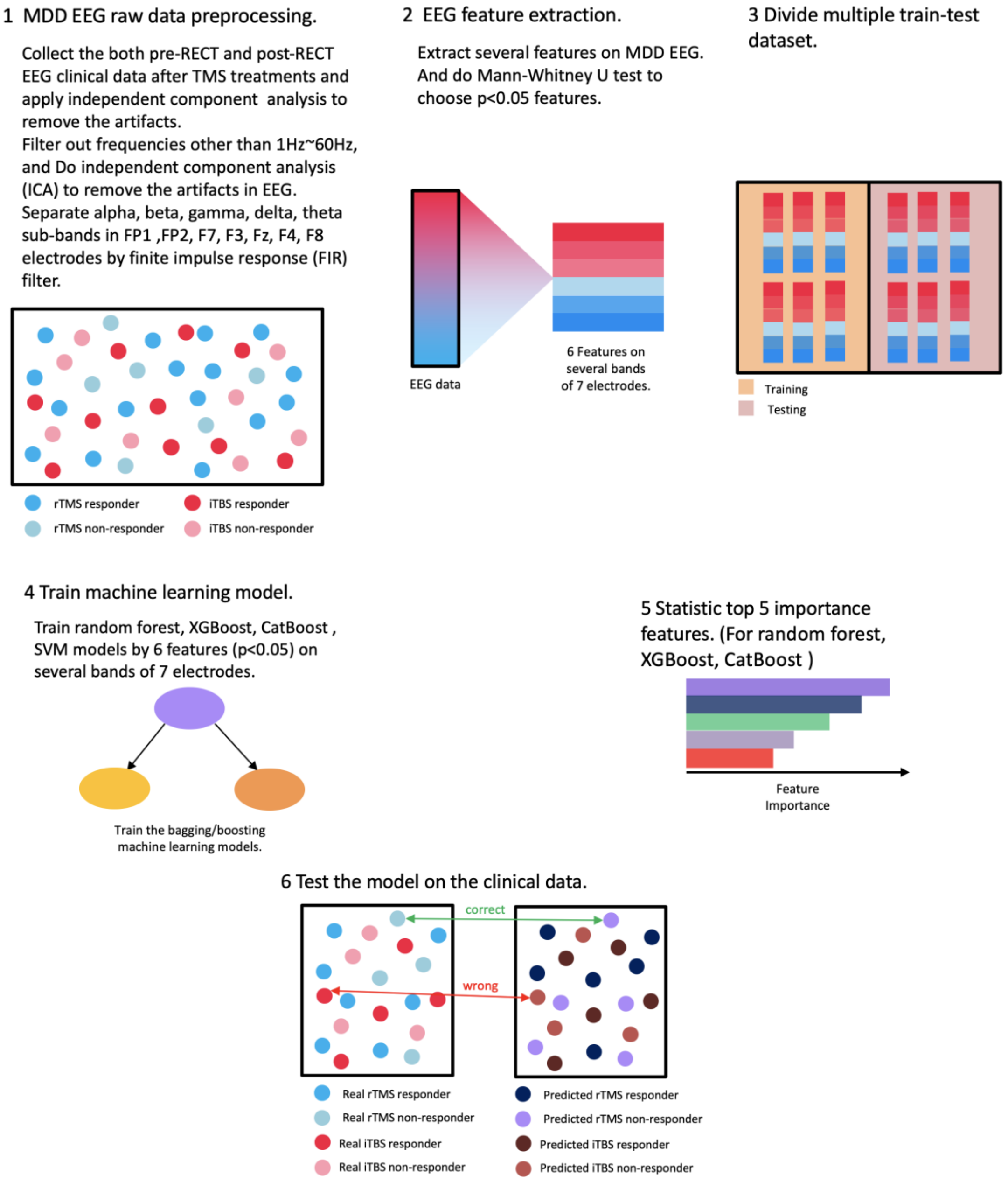
□ 8. Test the model with real world data (e.g., clinical data), and modify the model till the best performance.

3.5.1 Data Preprocessing

We use FIR band-pass filters to decomposed EEG into 5 bands (delta to gamma) after ICA. Let the input electroencephalography data which after ICA is D, define the bandpass filter $\mathcal{F}_k,\ k \subset \mathbb{N}$,

$$\begin{cases} \delta = \mathcal{F}_{[1,4)\mathcal{H}z}(D), \\ \theta = \mathcal{F}_{[4,8)\mathcal{H}z}(D), \\ \alpha = \mathcal{F}_{[8,15)\mathcal{H}z}(D), \\ \beta = \mathcal{F}_{[15,30)\mathcal{H}z}(D), \\ \gamma = \mathcal{F}_{[30,60)\mathcal{H}z}(D) \end{cases}$$

For the machine learning method, we should extract the features manually. We extract six features from the seven frontal electrodes.



1 MDD EEG raw data preprocessing.

Collect the both pre-RECT and post-RECT EEG clinical data after TMS treatments and apply independent component analysis to remove the artifacts.
Filter out frequencies other than 1Hz~60Hz, and Do independent component analysis (ICA) to remove the artifacts in EEG.
Separate alpha, beta, gamma, delta, theta sub-bands in FP1 ,FP2, F7, F3, Fz, F4, F8 electrodes by finite impulse response (FIR) filter.

2 EEG feature extraction.

Extract several features on MDD EEG. And do Mann-Whitney U test to choose p<0.05 features.

3 Divide multiple train-test dataset.

4 Train machine learning model.

Train random forest, XGBoost, CatBoost , SVM models by 6 features (p<0.05) on several bands of 7 electrodes.

5 Statistic top 5 importance features. (For random forest, XGBoost, CatBoost)

6 Test the model on the clinical data.

Figure 3.10: The methodology of classical machine learning.

3.5.2 Feature Extraction

The novelty in our previous work [31] and in this thesis is we extracted both the linear and nonlinear features. We integrated all of the feature extraction methods used for predicting antidepressant reaction for rTMS and iTBS both in our previous work [31] and in this thesis. The features we extracted are Largest Lyapunov Exponent (LLE) [91], Detrended Fluctuation Analysis (DFA) [92], Approximate Entropy (ApEn) [93], Katz Fractal Dimension (KFD) [94], Higuchi Fractal Dimension (HFD) [95], and Welch method [96].

Categories		**Methods**	**Features**
Nonlinear	Trend	Largest (Maximum) Lyapunov Exponent (LLE, MLE) [91]	Instability or unpredictability
		Detrended Fluctuation Analysis (DFA) [92]	Represent the long-range temporal correlation
	Complexity	Approximate Entropy (ApEn) [93]	Regularity and complexity
	Fractal Dimension	Katz Fractal Dimension (KFD) [94]	Obtains fractal dimension based on morphology, which measures the roughness of a time series
		Higuchi Fractal Dimension (HFD) [95]	Quantifies the self-similarity of an objection in the time domain
Linear	Band Power	Welch periodogram [96]	Compute band power by integration

$Table\ 3.4$: Feature extraction for EEG signals in the classical and quantum machine learning part this work.

doi:10.6342/NTU202101201

3.5.2.1 Largest Lyapunov Exponent

Lyapunov Exponents (LE), which provide a quantitative characterization of dynamical behavior, are related to the fast divergence or convergence of nearby paths in phase space. In most applications, the Largest Lyapunov Exponent (LLE) is widely used because LLE calculates the predictability of a system. The operation steps of LLE are described below. Assume $S = [x(1), x(2), \dots, x(N)]$ is the sequence of the data. Reconstructed phase space X is defined as

$$X(i) = [x(i), .., x(i + m - 1)]$$

where $i = 1,2, \dots, (N - m + 1)$ and m is the embedding dimension. The maximum Lyapunov exponent λ_1 is defined as

$$d_j(i) = d_j(0) e^{\lambda_1 i \Delta t}$$

where $d_j(i)$ is the average Euclidian distance between two nearby paths at t_i, and $d_j(0)$ is the Euclidian distance between $j\ th$ pair of nearest neighbors. We can have

$$ln\left(d_j(i)\right) = ln\left(d_j(0)\right) + \lambda_1 i \Delta t$$

Then we can calculate the LLE by fitting the slope of the best least squares curve to the mean log curve, which is defined as

$$y(i) = \frac{1}{\Delta t}\left(ln\left(d_j(t)\right)\right)$$

doi:10.6342/NTU202101201

3.5.2.2 Detrended Fluctuation Analysis

Detrended Fluctuation Analysis (DFA), a nonlinear method for measuring the degree of long-range correlations in time-series data, quantifies the temporal property as index by scaling exponent. The operation steps of DFA are described below. Assume $S = [x(1), x(2), \dots, x(N)]$ is the sequence of the data. N new sequences are constructed for $k = 1,2,3, \dots, N$,

$$y(k) = \sum_{i=1}^{k} [x(i) - \langle S \rangle]$$

where $\langle S \rangle$ is average value of S, which is defined as

$$\langle S \rangle = \frac{1}{N} \sum_{i=1}^{N} x(i)$$

$y(k)$ is divided into equal box sizes n, each box size has a least square line, denoted by $y_n(k)$. Then we can compute the root mean square fluctuation $F(n)$ by

$$F(n) = \sqrt{\frac{1}{N} \sum_{k=1}^{N} [y(k) - y_n(k)]^2}$$

Then the scaling exponent α of DFA can be computed as:

$$\alpha = \frac{log\big(F(n)\big)}{log(n)}$$

doi:10.6342/NTU202101201

3.5.2.3 Katz Fractal Dimension (KFD) and Higuchi Fractal Dimension (HFD)

The fractal dimension (FD) counts the effective number of degrees of freedom in the dynamical system and thus quantifies its complexity. Katz FD (KFD) and Higuchi FD (HFD) are two different methods to compute the self-similarity in FD.

3.5.2.3.1 Katz Fractal Dimension

The KFD obtains fractal dimension based on morphology, which measures the roughness of a time series. The operation steps of KFD are described below. Assume $S = [x(1), x(2), \dots, x(N)]$ is the sequence of the data. For $j = 1, \dots, N$. Find the maximum Euclidian distance between $x(1)$ and $x(j)$ and call it d. Compute L as the total length of the time series

$$L = \sum_{i=2}^{N} d\big(x(i), x(i-1)\big)$$

Compute a as the average distance between continuous points of S

$$a = \frac{L}{N-1}$$

Then we can compute KFD as

doi:10.6342/NTU202101201

$$KFD = \frac{ln\frac{L}{a}}{ln\frac{d}{a}}$$

3.5.2.3.2 Higuchi Fractal Dimension

The HFD of a time series is a measure of its complexity and self-similarity in the time domain, which can easily give us stable indices and time scale corresponding to the characteristic frequency of a data. The operation steps of HFD are described below. Assume $S = [x(1), x(2), \dots, x(N)]$ is the sequence of the data. New sequences of length k are constructed for m = 1, 2, ..., k.

$$X_m^k = \{x(m), x(m+k), \dots, x\left(m + \left\lfloor \frac{N-m}{k} \right\rfloor k\right)\}$$

where k determines the delay in continuous points. The normalized average length L_m can be calculated as

$$L_m(k) = \frac{1}{k} \frac{\sum_{i=1}^{\left\lfloor \frac{(N-m)}{k} \right\rfloor} |x(m+ik) - x[m+(i-1)]k| \times (N-1)}{\left\lfloor \frac{(N-m)}{k} \right\rfloor}$$

Compute the total average length for scale k

$$L(k) = \frac{1}{k} \sum_{m=1}^{k} L_m(k)$$

Then we can calculate the HFD as the slope of the best least squares line for the curve $logL(k)$ versus $log\left({}^{1}/_{k}\right)$.

doi:10.6342/NTU202101201

3.5.2.4 Approximate Entropy

Approximate entropy (ApEn) is a nonlinear method that quantifies the amount of regularity and the unpredictability of a time-series data. The larger the ApEn, the greater the irregularity and unpredictability of a time-series data. The operation steps of ApEn are described below. Assume $S = [x(1), x(2), \dots, x(N)]$ is the sequence of the data, $x^*(i)$ is the subsequence of the original data S.

$$x^*(i) = [x(i), x(i+1), \dots, x(i-m+1)]$$

where m is the length of sampling window. The threshold r = $SD * k$, where SD means the standard deviation of S and k represents a constant between 0.1 and 0.9. For each $1 \leq i, j \leq N - m + 1, i \neq j, C_i^m$ can be expressed as

$$C_r^m(r) = \frac{\sum_{j=1}^{N-M+1} \theta(r - |x^*(i) - x^*(j)|)}{N - M + 1}$$

Where θ is the Heaviside function. The quantity $\Phi^m(r)$ can be calculated as

$$\Phi^m(r) = \frac{\sum_{j=1}^{N-M+1} lnC_i^m(r)}{N - M + 1}$$

Then ApEn is defined as

$$ApEn = \Phi^m(r) - \Phi^{m+1}(r)$$

doi:10.6342/NTU202101201

3.5.2.5 Welch Periodogram

Welch periodogram is an approach for computing power spectral density (PSD). It is widely used for estimating the power of a signal. This method is based on the concept of using periodogram spectrum estimates, which are the result of converting a signal from time domain to frequency domain. The better performance of noise reduction from Welch periodogram is often desired in some works. The frequency spectrum is given by the operation step below.

$$P_W^p(f) = \frac{1}{P}\sum_{p=0}^{P-1}\frac{1}{UDT}\left|T\sum_{n=0}^{D-1} w[n]x^p[n]e^{-j2\pi fnT}\right|^2$$

Where U is the discrete-time window energy of $w[n]$

$$U = T\sum_{n=0}^{D-1} w^2[n]$$

3.5.3 Feature Selection

We use the Shapiro-Wilk test and Levene's test before performing the one-way ANOVA test to ensure that the normal distribution and homogeneity of variance are not broken, respectively. If the outcomes of the two experiments were statistically important $(p < 0.05)$, the Kruskal–Wallis test was used to compare them. Due to the limited sample size and non-normal distribution of our results, the Mann–Whitney U-

doi:10.6342/NTU202101201

test was used to compare responder and non-responder of each category. FDR-corrected $p < 0.05$, correcting for multiple comparisons. The feature set we selected and input into the machine learning and quantum machine learning model are statistically important $(p < 0.05)$.

3.5.4 Logistic Regression

Logistic regression is a widely used [180] linear statistical model which is usually used to model the probability of a certain binary class based on logistic function [180] and there are many more complex extensions such as for several class classification, etc. In this thesis, according to the previous work in our team [30], we used the most popular form, the logistic regression with l_2 penalty, the cost function shown as follows:

$$min_{w,c}\left[\frac{1}{2}w^T w + C\sum_{i=1}^{n} log(exp(-y_i(X_i^T w + c)) + 1)\right]$$

where X is the data, y means the label, c is a constant, $C = 1$ is the inverse of regularization strength, and w is the weight vector of the cost function. We use a type of quasi-Newton methods, an optimization algorithm that approximates the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm [181], to optimize this cost function with balanced class weight. In this thesis, we apply logistic regression as a

doi:10.6342/NTU202101201

linear analysis method with optimal threshold to classify the TMS responder or non-responder and compare the efficacy with the SVM machine learning method.

3.5.5 Support Vector Machine (SVM)

Support Vector Machine (SVM) [100] is a supervised machine learning algorithm for linear and nonlinear binary classification. The key concept of the SVM is that SVM need to separate the data points in different class by the hyperplanes with the largest optimal margin distance between each two classes. When the data points are not linearly-separable in the original space, kernel trick can be applied to project the data nonlinearly into higher dimensional space to hope that the hyperplane can separate them. To introduce simply this problem mathematically, we use the notation in [182]. Kernel SVM can be described as the follows:

$$K_{SVM}(x) = sign\left(\sum_{SV} \alpha_n y_n K(x_n, x') + b\right)$$

where *SV* is the support vectors, α_n is the optimal variable from the quadratic programming which fits the Karush-Kuhn-Tucker (KKT) optimality conditions, x_n is the data point, y_n is the label, $K(x_n, x')$ means the kernel matrix and *b* is the bias constant. In this thesis we use the popular nonlinear kernel matrix called radial basis function (rbf) kernel, also called infinite dimensional transform or Gaussian SVM.

doi:10.6342/NTU202101201

The rbf SVM is widely used in EEG classification [186]. The rbf kernel function shows below:

$$K(x_n, x') = exp(-\gamma \|x_n - x'\|^2), with\, \gamma > 0$$

and this kernel function can be extended by Taylor expansion:

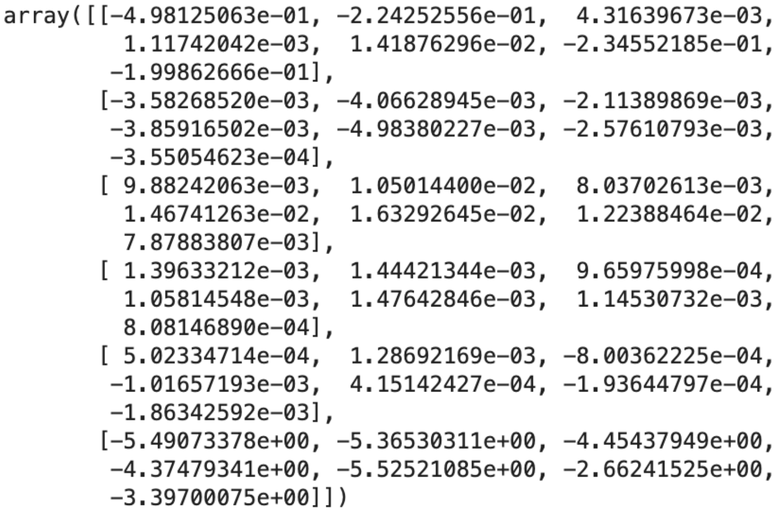
$$K(x_n, x') = \sum_{i=0}^{\infty} e^{-\gamma x_n^T x_n} \cdot e^{-\gamma x'^T x'} \cdot \frac{(2\gamma x_n^T x')^i}{i!}$$

$$= \sum_{i=0}^{\infty} e^{-\gamma x_n^T x_n} \cdot e^{-\gamma x'^T x'} \cdot \sqrt{\frac{2^i \gamma}{i!}} \sqrt{\frac{2^i \gamma}{i!}} (x_n^T x')^i$$

After Taylor expansion, it can be seen that it is actually a combination of various sub-items, from 0 sub-items to unlimited sub-items are included, the solution is actually a linear combination of Gaussian functions on *SV*.

3.5.6 Bagging and Boosting Classification Models

The extracted features are discreate (tabular), and the bagging/boosting machine learning algorithms are good at this [50, 51, 52].



```
array([[-4.98125063e-01, -2.24252556e-01,  4.31639673e-03,
         1.11742042e-03,  1.41876296e-02, -2.34552185e-01,
        -1.99862666e-01],
       [-3.58268520e-03, -4.06628945e-03, -2.11389869e-03,
        -3.85916502e-03, -4.98380227e-03, -2.57610793e-03,
        -3.55054623e-04],
       [ 9.88242063e-03,  1.05014400e-02,  8.03702613e-03,
         1.46741263e-02,  1.63292645e-02,  1.22388464e-02,
         7.87883807e-03],
       [ 1.39633212e-03,  1.44421344e-03,  9.65975998e-04,
         1.05814548e-03,  1.47642846e-03,  1.14530732e-03,
         8.08146890e-04],
       [ 5.02334714e-04,  1.28692169e-03, -8.00362225e-04,
        -1.01657193e-03,  4.15142427e-04, -1.93644797e-04,
        -1.86342592e-03],
       [-5.49073378e+00, -5.36530311e+00, -4.45437949e+00,
        -4.37479341e+00, -5.52521085e+00, -2.66241525e+00,
        -3.39700075e+00]])
```

$Figure$ 3.11: A part of iTBS 6 features on 7 electrodes of alpha band.

According to [45, 46], the kappa index κ is bigger, the trend of model's anti-overfitting (generalization) ability is better, in [45, 46], the kappa index values of the bagging model like random forest [97, 98] and the boosting model like XGBoost [99] are larger than SVM [100].

doi:10.6342/NTU202101201

3.5.6.1 XGBoost

XGBoost, named from extreme gradient boosting, a highly efficient adaptive machine learning algorithm for tree boosting, has been commonly used in many fields to produce state-of-the-art outcomes on a variety of data challenges. XGBoost is an optimized distributed gradient boosting designed to be highly efficient, flexible and portable. The scalability of XGBoost is attributed to many important systems and algorithmic optimizations, including a novel tree learning algorithm, a theoretically justified weighted quantile sketch technique, as well as parallel and distributed computing, as developed by Tianqi Chen et al. It implements machine learning algorithms under the Gradient Boosting framework. XGBoost provides a parallel tree boosting (also known as GBDT, GBM) that solve many data science problems in a fast and accurate way. Tree boosting is a powerful ensemble learning algorithm that combines many weak classifiers into a single strong classifier for improved classification results. Let $D = (x_i, y_i)(|D| = n, x_i \in \mathbb{R}^{\mathrm{m}}, \mathrm{y_i} \in \mathbb{R}^{\mathrm{n}})$ represents a dataset with n samples and m features.

A tree boosting model with K trees output $\widehat{y_i}$ is defined as follows:

$$\widehat{y_i} = \sum_{k=1}^{K} f_k(x_i), f_k \in F$$

where F is the function space of regression or classification trees (CART), T denotes the number of leaves on the tree and ω denotes the weights of leaf :

$$F = \{f(x) = \{\omega_q(x) |\, q\colon \mathbb{R}^m \longrightarrow T, \omega \in \mathbb{R}^T\}\}$$

The decision tree of XGBoost is defined as $f_k(x) = w_{q(x)}$, x is a sample, where $q(x)$ represents the leaf node where the sample is located, and w_q is the leaf node weight w, so $w_{q(x)}$ is the value w of each sample (that is, the predicted value).

The objective function of tree model is

$$O(\phi) = \sum_i l(\widehat{y_i}, y_i) + \sum_k \Omega\,(f_k)$$

In the equation, $l(\widehat{y_i}, y_i)$ is the convex, differentiable loss function which measures the distance between the prediction $\widehat{y_i}$ and the object y_i, the second term Ω represents the penalty term of the tree model complexity. Minimizing the following objective function will be used to learn the set of functions f_k in the tree model.

In Euclidean space, a tree boosting model with the objective function above cannot be optimized using standard optimization approaches. Gradient Tree Boosting is an advanced variant of tree boosting that uses an additive training method to train the tree model, which means the prediction of the t-th iteration $\widehat{y^{(t)}} = \widehat{y^{(t-1)}} + f_t(x)$.

Thus, the t-th iteration is updated as

$$O^{(t)}(\phi) = \sum_{i=1}^{n} l\left(y_i, \widehat{y_i^{(t-1)}} + f_t(x_i)\right) + \Omega\,(f_t)$$

doi:10.6342/NTU202101201

XGBoost approximates the equation above by utilizing the second order Taylor expansion and the final objective function at step t can be rewritten as:

$$O^{(t)} \cong \widetilde{O^{(t)}} = \sum_{i=1}^{n}\left[l\left(y_i, \widehat{y_\imath^{(t-1)}}\right) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^{2}(x_i)\right] + \Omega\,(f_t)$$

where g_i and h_i are first and second order gradient statistics on the loss function, and in XGBoost, model complexity is $\Omega\,(f_t) = \gamma T + \frac{1}{2}\lambda\left|\left|w\right|\right|^2$. T represents the number of leaf nodes of a given tree, γ and λ are the penalty terms. $\left|\left|w\right|\right|^2$ represents the square of the output score on each leaf node (equivalent to L2 regularity). From the definition of the objective function, it can be seen that the penalty term of XGBoost takes into account the number of leaf nodes of each tree and the sum of squares of the output score of each leaf node for the model complexity. Denote $I_j = \{i|q(x_i) = j\}$ as the instance set of leaf j, after removing the constant terms and expanding Ω, the equation simplified as below:

$$\widetilde{O^{(t)}} = \sum_{j=1}^{T}\left[\left(\sum_{i\in I_j} g_i\right)\omega_j + \frac{1}{2}\left(\sum_{i\in I_j} h_i + \lambda\right)\omega_j^2\right] + \gamma T$$

The following equation can be used to calculate the solution weight ω_j^* of leaf j for a given tree structure $q(x)$:

$$\omega_j^* = -\frac{\sum_{i\in I_j} g_i}{\sum_{i\in I_j} h_i + \lambda}$$

Substitute the ω_j^* into $\widetilde{O^{(t)}}$, we can get a scoring function to evaluate the tree structure $q(x)$ and find the optimal tree structures for classification:

$$\widetilde{O(q)} = -\frac{1}{2}\sum_{j=1}^{T}\frac{\left(\sum_{i\in I_j} g_i\right)^2}{\sum_{i\in I_j} h_i + \lambda} + \gamma T$$

In fact, however, it is difficult to search all feasible tree structures q. The XGBoost paper defines a greedy algorithm that begins with a single leaf and grows the tree structure by adding branches iteratively. The following function will determine whether or not a split should be added to the current tree structure:

$$O_{split} = \frac{1}{2}\left[\frac{\left(\sum_{i\in I_L} g_i\right)^2}{\sum_{i\in I_L} h_i + \lambda} + \frac{\left(\sum_{i\in I_R} g_i\right)^2}{\sum_{i\in I_R} h_i + \lambda} - \frac{\left(\sum_{i\in \{\mathrm{I}|I=I_L\cup I_R\}} g_i\right)^2}{\sum_{i\in \{\mathrm{I}|I=I_L\cup I_R\}} h_i + \lambda}\right] - \gamma$$

where I_L and I_R are the instance sets of left and right nodes after the split. The author emphasizes that compared with regularized greedy forest, this penalty item is more convenient for optimization of parallel computing. XGBoost is a fast implementation of GB algorithm, which has the advantages of fast speed and high accuracy.

3.5.6.2 CatBoost

CatBoost [101], named from categorical boosting, is a kind of gradient boosting decision tree (GBDT) machine learning algorithm and it was proposed by engineers of Yandex company in 2017. It can deal with problems involving multiple functions, noisy data, and complex dependencies. CatBoost has the following benefits over other GBDT algorithms:

Firstly, instead of preprocessing time, categorical characteristics are dealt with during training time. CatBoost allows the use of whole dataset for training. This algorithm is capable of handling categorical functions. Categorical elements can be replaced with average label values using the traditional GBDT algorithm. The average label value will be used as the metric for node splitting in a decision tree. Greedy Target-based Statistics (GTBS) is the name of this procedure, which is defined as follows:

$$GTBS = \frac{\sum_{j=1}^{p}\left[x_{j,k} = x_{i,k}\right]Y_i}{\sum_{j=1}^{n}\left[x_{j,k} = x_{i,k}\right]}$$

Features, on the whole, provide more detail than labels. If we forcibly reflect features using average label worth, it will lead to a conditional shift [102]. CatBoost adds a prior value to GTBS to solve this problem. Assume we've been given a series of data $D = \{X_i, Y_i\}, i = 1, \dots, n$ to work with, if a permutation is $\sigma = [\sigma_1, \dots, \sigma_n]_n^T$,

$$x_{\sigma_{p,k}} = \frac{\sum_{j=1}^{p-1}\left[x_{\sigma_{j,k}} = x_{\sigma_{p,k}}\right]Y_{\sigma_j} + \beta P}{\sum_{j=1}^{p-1}\left[x_{\sigma_{j,k}} = x_{\sigma_{p,k}}\right] + \beta}$$

where P is a prior value and β is the prior weight. For regression tasks, the basic method for measuring prior is to take the dataset's average label value.

Secondly, many of the categorical characteristics could be combined to form a single one. CatBoost considers the combinations in a greedy manner while creating a new split for the tree. For the first split in the tree, no combinations are considered; however, for the second and subsequent splits, CatBoost blends all predetermined combinations with all categorical features in the dataset. All splits selected in the tree are considered as a category with two values and used in combination.

Thirdly, the most important contribution of CatBoost is the unbiased boosting with categorical features. When using the target statistics method to transform categorical attributes into numerical values, the distribution will vary from the initial distribution, and this divergence will allow the solution to deviate, which is an unavoidable issue for conventional GBDT methods. Via theoretical research, CatBoost paper developed a new approach for overcoming gradient bias called ordered boosting. The following is the pseudo-code for ordered boosting:

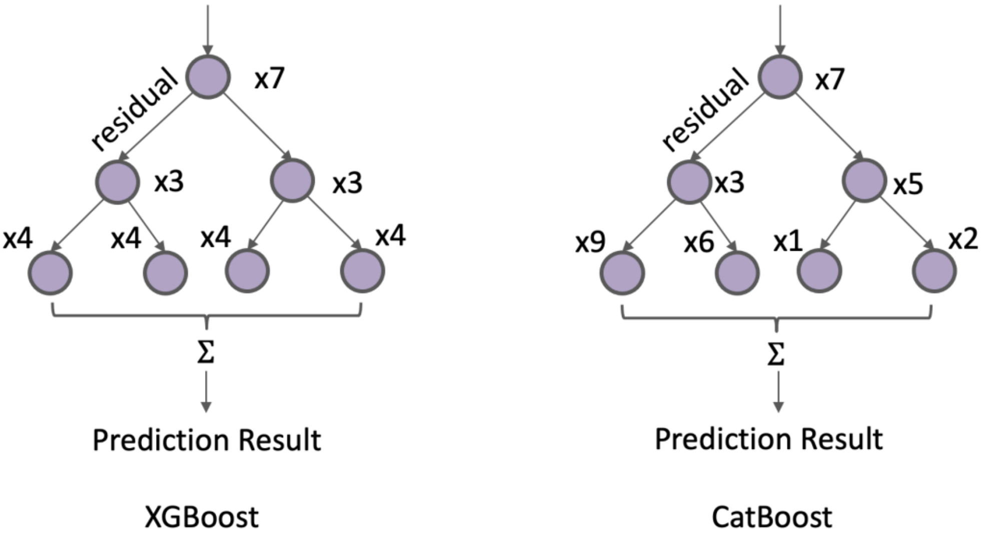
Algorithm: Ordered Boosting [101]

```
Input: ${(X_k, Y_k)}_{k=1}^{n}$ ordered according to σ, the number of trees $I$ ;
    σ ← random permutation of $[1, n]$
    $M_i \leftarrow 0$ for $i = 1, \dots, n$
    for $t \leftarrow 1\ to\ I$ do
      for $i \leftarrow 1\ to\ n$ do
          $r_i \leftarrow y_i - M_{\sigma(i)-1}(X_i)$;
    for $i \leftarrow 1\ to\ n$ do
    $\Delta M \leftarrow LearnModel[(X_i, r_j) : \sigma(j) \leq i]$
    $M_i \leftarrow M_i + \Delta M$
return $M_n$.
```

Note that M_i is trained without using the example X_i and each M_i shares the same tree structure.

$Table$ 3.5: Order Boosting algorithm.

Random permutations of the training data are produced in CatBoost. By sampling a random permutation and collecting gradients on its basis, several permutations can be used to improve the algorithm's robustness. These permutations are the same as those used to calculate categorical function statistics, and CatBoost does gradient algorithm by minimal variance sampling (MVS) [185] to let every leaf node has different weights.



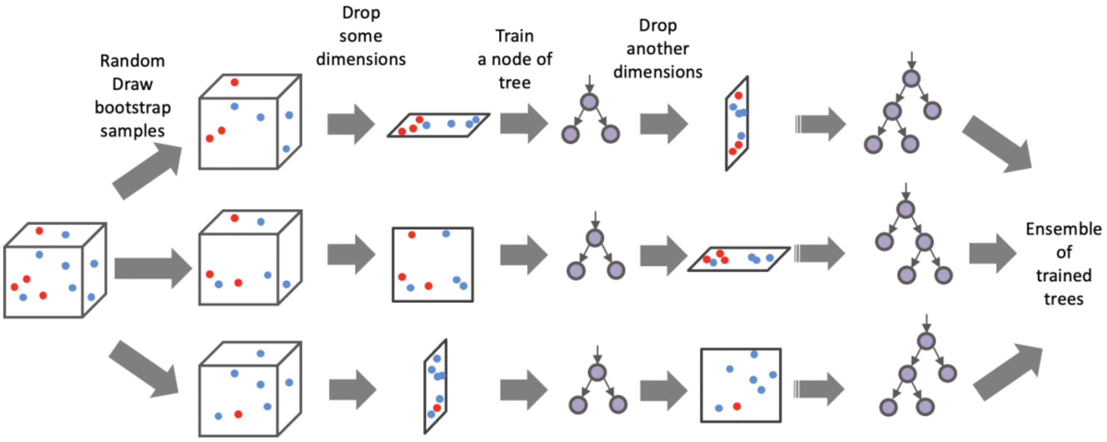
$Figure$ 3. 12: The comparison of XGBoost and CatBoost algorithms.

3.5.6.3 Random Forest

Random Forests (RF) combines multiple decision tree classifiers and employs Breiman's "bagging" concept to combine multiple decision trees into a single, powerful model. It employs the self-help approach (also known as bootstrap

doi:10.6342/NTU202101201

resampling technology) to create new training sample sets from the initial N training samples by selecting random k (k<N) sets of samples repeatedly. Some samples may be taken more than once during the overall selection process. Approximately 36.8% of the training data will not be sampled in each round of random bagging sampling, referred to as out-of-bag (OOB) data. These uncollected data aren't included in model fitting during training, but they can be used to determine a model's generalization capacity.



$Figure$ 3.13: Random Forest algorithm.

The training sample is used to produce k buffeting decision or regression trees (CART) for the construction of random forests, and the test sample is then classified using majority vote decision or average return values. Random forests in general can achieve strong generalization ability and low variance tolerance without additional pruning due to the fact that randomness can efficiently minimize model variance. Of

doi:10.6342/NTU202101201

course, as the model's fit improves during testing, the bias increases, but this is just relative.

The CART algorithm, though, is a binary tree, which means that each non-leaf node can only lead to two branches. A non-leaf node that is a multi-level (more than two) discrete variable is likely to be seen more than once. Around the same time, if a non-leaf node is a continuous variable, it would be treated as a discrete variable by the decision tree. The CART used in RF is based on the Gini coefficient's feature selection. The Gini coefficient is chosen based on the requirement that each child node achieves maximum purity, with all observations on that child node falling under the same classification. In this case, the Gini coefficient achieves its lowest value while maintaining the greatest purity, thus Gini coefficient is also called Gini impurity in some papers.

CART's Gini impurity on the n-th tree which has J nodes is calculated as follows:

$$Gini(n) = \sum_{k=1}^{J} p_k(1 - p_k)$$

When traversing each feature's segmentation points, if feature A is used, D is divided into two parts, namely D_1 (the sample set is A) and D_2 (the sample set is not A). The Gini coefficient of D is:

$$Gini(D, A) = \frac{D_1}{D} Gini(D_1) + \frac{D_2}{D} Gini(D_2)$$

doi:10.6342/NTU202101201

where $Gini(D, A)$ represents the uncertainty of D. The aim of a CART in RF is to find the feature segmentation point with the smallest Gini coefficients by traversing all possible feature segmentation points in the tree and splitting the data set into two subsets until the stop condition is met.

```
Algorithm: Random Forest [52, 97]
1. for $b = 1$ to $B$:
   (a) Draw a bootstrap sample $X^*$ of size $N$ from the training data
   (b) Grow a decision tree $T_b$ to the data $X^*$ by doing the following recursively
       until the minimum node size $n_{min}$ is reached:
       i. Select $m$ of the $p$ variables
       ii. Pick the best variable/split-point from the $m$ variables and partition
2. Output the ensemble $\{T_b\}_b^B$

Let $\hat{C}_b(x^*)$ be predicted class of tree $T_b$. Then $\hat{C}_{rf}^B(x^*) =$ majority vote$\hat{C}_b(x^*)_1^B$.
```

$Table\ 3.6$: Random Forest algorithm.

In RF, the CART algorithm is not like other conventional algorithms. To begin with, each function chosen for use in an RF tree is created at random from all m features, reducing the probability and propensity of overfitting. Relevant eigenvalues or function variations would not decide the model. The regulation of the model's fitting potential would not change forever as randomness is increased. Second, unlike ordinary decision trees, RF increases decision tree establishment. To do the left and right subtree division of a decision tree, an ordinary decision tree requires selecting an optimal function from among all m sample features on the node. Each RF tree, on the other hand, is made up of specific features. An optimal function is chosen from among these few features to separate the left and right subtrees of the decision tree, increasing the impact of randomness and the model's generalization potential. If each tree has m sub properties, the smaller the m sub, the weaker the model's fitting degree to the training set will be, and the bias will increase, but the model's generalization potential will be greater, and the variance will decrease. The same is true for a broader m sub. In reality, the value of m sub is usually treated as a parameter that is continuously tweaked until it reaches a satisfactory level.

3.5.7 The Booster Transformation

I figured out a navel test data preprocessing method, Booster Transformation. Let X is the training dataset and Y is the testing dataset in the dataset D_{EEG}, $F_A(\cdot)$ is the feature set which generate by A algorithm.

Thus,

$$\mathcal{F} = \{F_{LLE}(X), F_{DFA}(X), F_{KFD}(X), F_{HFD}(X), F_{ApEn}, F_{Welch}\}$$

And the booster transformation is described as below:

$$Booster\ Transformation = \bigcup_{i=0}^{dim(\mathcal{F})-1} (Y_i \odot \mathcal{F}_i)$$

doi:10.6342/NTU202101201

which $\odot$ is Hadamard product.

And apply on model M, the best result can select as:

$$sup_i\{M_r[\bigcup_{i=0}^{dim(\mathcal{F})-1}(Y_i \odot \mathcal{F}_i)]|r \in (Accuracy, Sensitivity, Specificity)\}$$

This transformation is a filter that can amplify the characteristics of the extracted feature set, and then shift the distribution of the sensitivity and specificity on the same dataset applied model. In the field of clinical medicine, need to select the high sensitivity model, for the sensitivity (true positive rate) is higher, the rate which the real patients do treatment is higher.

doi:10.6342/NTU202101201

3.6 Quantum Machine Learning Methods

All the classical signal preprocessing and feature extraction processes are run on the computer with python 3.6+ and all the quantum computing are done with Qiskit 0.22.0 on IBMQ quantum computer. The rough experiment flow is as the following:

□ 1. Read raw EEG data (.cnt/.edf/.eeg) and transform them into .npy for python processing.

□ 2. Filter out frequencies other than 1Hz~60Hz.

□ 3. Do independent component analysis (ICA) to remove the artifacts in EEG.

□ 4. Separate alpha, beta, gamma, delta, theta sub-bands in FP1, FP2, F7, F3, Fz, F4, F8 electrodes by finite impulse response (FIR) filter.

□ 5. Linear and nonlinear features extraction and feature selection.

□ 6. Divide the data into train set and test set. Train the quantum machine learning classifiers models with train dataset.

□ 7. Optimize the number of used qubits.

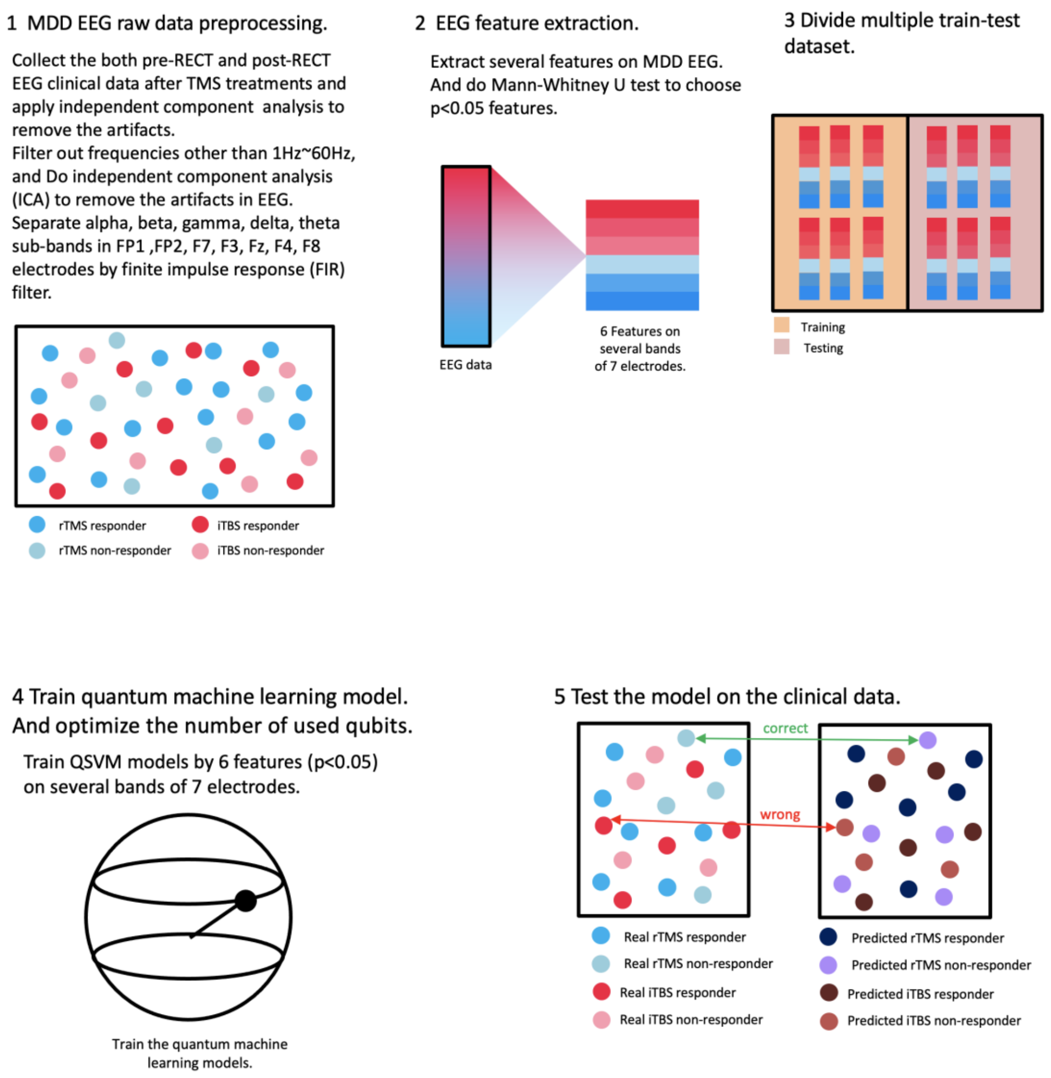
□ 8. Validation the model with test dataset (70:30), and modify the model till the best performance.

□ 9. Test the model with real world data (e.g., clinical data), and modify the model till the best performance.

doi:10.6342/NTU202101201

3.6.1 Data Preprocessing, Feature Extraction and Selection

I used FIR band-pass filters to divide EEG into 5 bands after ICA. Other processing steps are the same as section 3.5.1, 3.5.2, and 3.5.3.



1 MDD EEG raw data preprocessing.

Collect the both pre-RECT and post-RECT EEG clinical data after TMS treatments and apply independent component analysis to remove the artifacts.
Filter out frequencies other than 1Hz~60Hz, and Do independent component analysis (ICA) to remove the artifacts in EEG.
Separate alpha, beta, gamma, delta, theta sub-bands in FP1 ,FP2, F7, F3, Fz, F4, F8 electrodes by finite impulse response (FIR) filter.

2 EEG feature extraction.

Extract several features on MDD EEG. And do Mann-Whitney U test to choose p<0.05 features.

3 Divide multiple train-test dataset.

4 Train quantum machine learning model. And optimize the number of used qubits.

Train QSVM models by 6 features (p<0.05) on several bands of 7 electrodes.

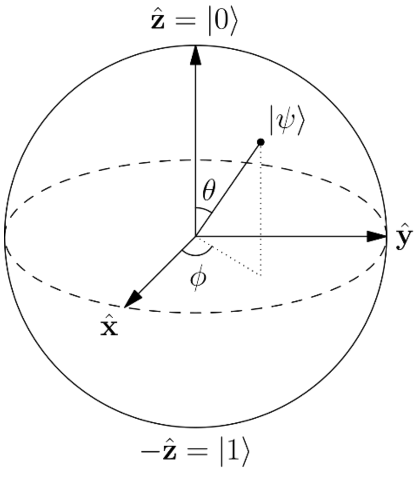
5 Test the model on the clinical data.

Figure 3.14: The methodology of quantum machine learning.

doi:10.6342/NTU202101201

3.6.2 Introduction to Quantum Computing

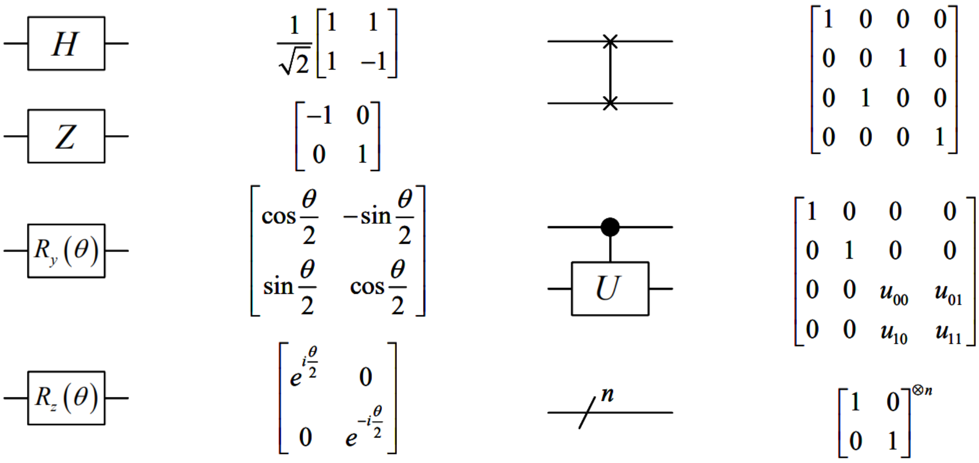
The most widely accepted model of quantum computation describes the process as a network of quantum logic gates. A classical circuit can be thought of as an abstract linear-algebraic generalization of this model. A quantum computer capable of successfully operating these circuits is known to be physically realizable and this circuit model obeys quantum mechanics. There are 2^n possible states in a memory of n bits of data. As a result, a vector containing all memory states has 2^n entries (one for each state). This vector can be thought of as a chance vector, as it indicates that the memory will be located in a certain state. The Bloch sphere is a representation of a qubit, the fundamental building block of quantum computers [178]. In the view of classical computation, one entry would have a value of 1 (i.e., a 100% probability of being in this state) and all other entries would have a value of zero. Probability vectors are extended to density operators in quantum mechanics.



$Figure$ 3.15: Bloch sphere.

doi:10.6342/NTU202101201

This is the mathematical basis for quantum logic gates that is theoretically rigorous, but the intermediate quantum state vector formalism is typically adopted first because it is conceptually simpler [178], thus a quantum computing algorithm can be divided into several reversible matrix multiplication combinations of quantum logic gates in this view. The quantum gates must be unitary operators mathematically, and are often described as unitary matrices relative to some orthogonal basis.


H
$\frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$
$\begin{bmatrix}1&0&0&0\\0&0&1&0\\0&1&0&0\\0&0&0&1\end{bmatrix}$
Z
$\begin{bmatrix}-1 & 0\\ 0 & 1\end{bmatrix}$
$R_y(\theta)$
$\begin{bmatrix}\cos\frac{\theta}{2} & -\sin\frac{\theta}{2}\\ \sin\frac{\theta}{2} & \cos\frac{\theta}{2}\end{bmatrix}$
U
$\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&u_{00}&u_{01}\\0&0&u_{10}&u_{11}\end{bmatrix}$
$R_z(\theta)$
$\begin{bmatrix}e^{i\frac{\theta}{2}} & 0\\ 0 & e^{-i\frac{\theta}{2}}\end{bmatrix}$
n
$\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}^{\otimes n}$

$Figure\ 3.16$: Common quantum logic gates [44].

Quantum bits (qubits) have the superposition properties in quantum physics, the superposition can be represented as a vector:

$$|\psi_n\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i_b\rangle$$

where $\alpha_i \in \mathbb{C}, \sum_{i=0}^{2^n-1} |\alpha_i|^2 = 1$ is the amplitude of the qubits, $|i_b\rangle$ is the computational basis in the Hilbert space, and $|\psi_n\rangle$ is the quantum state. To know the computational result from the quantum processer after the computation, we need to perform the quantum measurement on the quantum state $|\psi_n\rangle$, and $|\psi_n\rangle$ will collapse to the basis state $|i_b\rangle$ with the probability $|\alpha_i|^2$. The qubits can have entanglement states by the matrix multiplication combinations of quantum gates, this is the key powerful characteristic that can reduce the computational time complexity exponentially and speed-up the computation.

3.6.3 Quantum Support Vector Machine (QSVM)

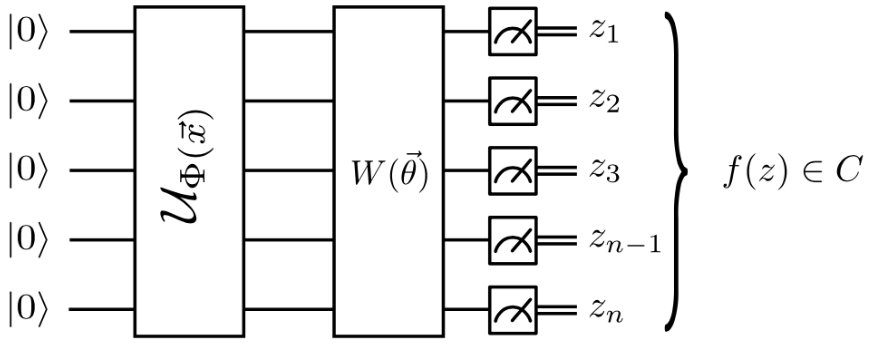
Quantum support vector machine (QSVM) is a kind of quantum machine learning algorithm which figured out by [103]. In the field of quantum machine learning, QSVM [106] is a kind of quantum variational classification algorithm. QSVM is a quantum computing version support vector machine algorithm in general.

The difference between the classical computing and quantum computing is that quantum bit (qubit) has three main characteristics which different from classical bit: quantum superposition, quantum entanglement and quantum tunneling.

These characteristics can provide several special computing peculiarities based on quantum physics like reduce the computational complexity [104], enhance the state space for feature embedding [105, 108], etc. The QSVM construct a separating hyperplane in the state space of n qubits, and the classical data of the features extracted from MDD EEG are mapped to the state space with $dim = 4^n$ using a unitary circuit family starting from the reference state $|0\rangle\langle 0|^n$.

In a similar way to support vector machines (SVM), QSVM takes advantage of quantum processor's big dimensional Hilbert space to find an ideal cutting hyperplane. The algorithm is divided into two parts: a training stage and a classification stage.

A collection of named data points is given for the training level, on which the algorithm is run. We take a separate set of data points for the classification stage and run the optimized classifying circuit on them without any label input. Then we calculate a performance ratio for the data collection by comparing the mark of and data point to the output of the classifier. The quantum circuit that implements the algorithm has three key sections for both the training and classification stages: feature map encoding, variational optimization, and measurement.



$Figure\ 3.17$: Quantum variational classification [106].

The circuit takes a references state, $|0\rangle^n$, applies the unitary $\mathcal{U}_{\Phi(\vec{x})}$ followed by the variational unitary $W(\vec{\theta})$ and applies a measurement in the Z-basis. The resulting bit string $z \in \{0,1\}^n$ is then mapped to a label in C. This circuit is sampled and re-run several times to approximate the expected performance. The expectation value is

$$p_y = \langle \Phi(\vec{x}) | W^{\dagger}(\vec{\theta}) M_y W(\vec{\theta}) | \Phi(\vec{x}) \rangle$$

for the labels $y \in C = +1, -1$.

```
Algorithm: Quantum Variational Classification: The Training Phase [106]
Input: Labeled training samples $T = \{\vec{x} \in \Omega \subset \mathbb{R}^n\} \times \{y \in C\}$, Optimization
routine.
Parameters: Number of measurement shots $R$, initial parameter $\overrightarrow{\theta_0}$.
Calibrate the quantum hardware to generate short depth trial circuits.
Set initial values of the variational parameters $\vec{\theta} = \overrightarrow{\theta_0}$ for the short-depth circuit
$W(\vec{\theta})$
while Optimization (e.g., SPSA [156]) of $R_{emp}(\vec{\theta})$ has not converged do
  for$i = 1to|T|$ do
      Set the counter $r_y = 0 \ \forall \ y \in C$.
      for $shot = 1 \ to \ R$ do
         Use $\mathcal{U}_{\Phi(\vec{x}_i)}$ to prepare initial feature map state $|\Phi(\vec{x}_i)\rangle\langle\Phi(\vec{x}_i)|$
         Apply discriminator circuit $W(\vec{\theta})$ to the initial feature map state.
         Apply $|C| -$ outcome measurement $\{M_y\}_{y \in C}$
         Record measurement outcome label $y$ by setting $r_y \to r_y + 1$
      end for
      Construct empirical distribution $\widehat{p_y}(\vec{x}_i) = r_y R^{-1}$.
      Evaluate $Pr(\widetilde{m}(\vec{x}_i) \neq y_i | m(\vec{x}) = y_i)$ with $\widehat{p_y}(\vec{x}_i)$ and $y_i$
      Add contribution $Pr(\widetilde{m}(\vec{x}_i) \neq y_i | m(\vec{x}) = y_i)$ to cost function $R_{emp}(\vec{\theta})$.
   end for
   Use optimization routine to propose new $\vec{\theta}$ with information from $R_{emp}(\vec{\theta})$
end while
return the final parameter $\overrightarrow{\theta^*}$ and value of the cost function $R_{emp}(\theta^*)$.
```

Table 3.7: Quantum Variational Classification algorithm: the training phase.

After the training phase is complete, the classification can be applied.

Algorithm: Quantum Variational Classification: The Classification Phase [106]

```
Input: An unlabeled sample from the test set $\vec{s} \in S$, optimal parameters $\overrightarrow{\theta^*}$ for the discriminator circuit.
Parameters: Number of measurement shots $R$.
Calibrate the quantum hardware to generate short depth trial circuits.
Set the counter $r_y = 0 \; \forall \; y \in C$
for $shot = 1 \; to \; R$ do
    Use $\mathcal{U}_{\Phi(\vec{s})}$ to prepare initial feature map state $|\Phi(\vec{s})\rangle\langle\Phi(\vec{s})|$
    Apply optimal discriminator circuit $W(\overrightarrow{\theta^*})$ to the initial feature map state.
    Apply $|C| -$ outcome measurement $\{M_y\}_{y \in C}$
    Record measurement outcome label $y$ by setting $r_y \to r_y + 1$
end for
Construct empirical distribution $\widehat{p_y}(\vec{s}) = r_y R^{-1}$.
Set label $= argmax_y\{\widehat{p_y}(\vec{s})\}$
return label
```

$Table\ 3.8$: Quantum Variational Classification algorithm: the classification phase.

We construct the classifier portion of the variational algorithm by appending layers of single-qubit unitaries and entangling gates to the layout of the function map circuit.

The general circuit is comprised of the following sequence of single qubit and multi-qubit gates:

$$W(\vec{\theta}) = U_{loc}^{(l)}(\theta_l)U_{ent} \dots U_{loc}^{(2)}(\theta_2)U_{ent}U_{loc}^{(1)}(\theta_1).$$

where the circuit l repeated entanglers, layers of local single qubit rotations should be interspersed between them:

$$U_{loc}^{(t)}(\theta_t) = \bigotimes_{m=1}^{n} U(\theta_{m,t})$$

and

$$U(\theta_{m,t}) = e^{i\frac{1}{2}\theta_{m,t}^{z} Z_m} e^{i\frac{1}{2}\theta_{m,t}^{y} Y_m}$$

parametrized by $\theta_t \in \mathbb{R}^{2n}$, and the $\theta_{i,t} \in \mathbb{R}^2$. U_{ent} has many choices theoretically, and in this thesis we chose as follows,

$$U_{ent} = \prod_{\{(i,\, j)\in E\, ||E|\le n(n-1)/2\}} CX$$

we used the entangler that is comprised of products of control phase gates $CX = CX(i,j)$ (controlled Pauli X gate, also called controlled-x gate) between qubits i and j by the default of Qiskit [109] settings.

To reduce the number of parameters to be treated by the classical optimizer, the classifier's single-qubit unitaries are limited to Y and Z rotations.

The optimal parameters are used to decide the correct label for new input data. In comparison to the classical equivalent, an exponential acceleration is achieved using Hamiltonian simulation and matrix inversion in the original QSVM model. When the input training feature set $\mathcal{T} = \{(\vec{\boldsymbol{x_i}}, y_i): \vec{\boldsymbol{x_i}} \in \mathbb{R}^L, y_i = \pm 1\}_{i=1}^{M}$, the following system of linear equations is used to formulate it:

$$F^{(M+1)\times(M+1)} \begin{pmatrix}\beta \\ \vec{\alpha}\end{pmatrix} \equiv \begin{pmatrix}0 & \vec{1}^T \\ \vec{1} & K + \gamma^{-1}\cdot\vec{1}^T\end{pmatrix} \begin{pmatrix}\beta \\ \vec{\alpha}\end{pmatrix} = \begin{pmatrix}0 \\ \vec{y}\end{pmatrix}$$

where *L* is the number of the electrode channel, *M* is the number of the extracted features, $\boldsymbol{x_i}$ is the input classical feature, y_i is the label with responder$= +1$ and non-responder$= -1$, $K_{ij} = k(\vec{x_i}, \vec{x_j})$ is the symmetric kernel matrix, $\vec{1} = [1,1,1,\ldots,1]^{\mathrm{T}}$ and $\gamma, \beta \in \mathbb{R}$ is constant. The $\vec{\alpha} = [\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_M]^T$ is a vector that as a notation of distances from the ideal margin. The additional row and column with the $\vec{1}$ arise because of a nonzero offset β. Obviously, QSVM need to learn the model parameters by solving the

$$\begin{pmatrix} \beta \\ \vec{\alpha}^T \end{pmatrix} = F^{-1} \begin{pmatrix} 0 \\ \vec{y}^T \end{pmatrix}$$

One can predict the class of new data point $\vec{x}$ after giving $\vec{\alpha}$ and β by

$$y(\vec{x}) = sign\left[\sum_{i=1}^{M} \alpha_i k(\vec{x_i}, \vec{x_j}) + b\right] = sign\left[\sum_{i=1}^{M} \alpha_i \langle \varphi(\vec{x_i}) | \varphi(\vec{x_j}) \rangle + b\right]$$

where $\varphi(\cdot)$ is the quantum state mapping, in QSVM, we use Harrow Hassidim Lloyd algorithm (HHL) [107] to solve the inverse value of the matrix above, on the other word, to solve the hyperplane. In this work, we implemented QSVM with Qiskit and run the experiment on the real IBMQ quantum computer.

Gate(s)	Operator	Matrix
U1	U1 gate (phase gate)	$U1(\lambda) = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\lambda} \end{pmatrix}$
U2	U2 gate	$U2(\phi, \lambda) = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -e^{i\lambda} \\ e^{i\phi} & e^{i(\phi+\lambda)} \end{pmatrix}$
⊕	controlled-x gate (CX)	$CX = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$
Z	Measurement	Measurement in the z basis.

Table 3.9: Operations glossary of the Qiskit.

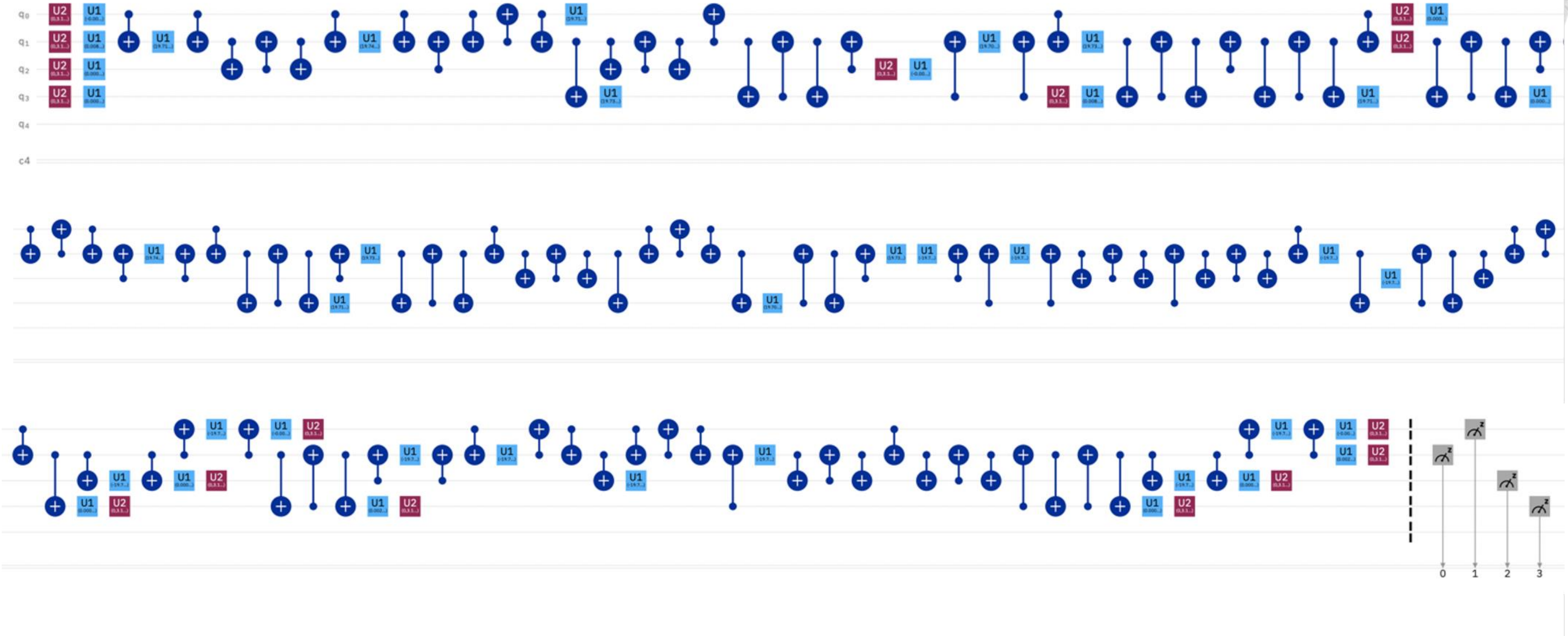


Figure 3.18: The 4-qubit QSVM quantum circuit used in this thesis.

Chapter 4 Experiment Results

4.1 Deep Learning Results

4.1.1 Deep Machine Learning Results

The TMS treatment response prediction results by deep learning algorithms will be shown in this section. To proof ACTSNet's performance, I did the performance comparison with various famous deep learning models for time series classification like classical convolution neural network (CNN), fully convolution network (FCN), multi-channel deep convolutional neural network (MCDCNN), residual network (ResNet), Time Le-Net (TLENT), time warping invariant echo state network (TWIESN), long short-term memory- fully convolution network (LSTM-FCN),

Biomarker	Frontal EEG signal												
Treatment	rTMS												
Patients (N)	30 (OPD dataset)												
EEG Features	Neural Network extracted nonlinear features on Delta, Theta, Alpha, Beta, Gamma bands												
Classification	CNN							RNN	RNN-CNN				ACNN
	CNN	FCN	MCDCNN	ResNet	TLENT	Inception Time	Encoder	TWIESN	LSTM-FCN	ALSTM-FCN	MALSTM-FCN	TapNet	**ACTSNet**
Accuracy	60.0%	46.9%	60.0%	43.8%	53.1%	56.2%	64.5%	60.0%	45.3%	54.7%	45.3%	**76.3%**	**83.5%**
Sensitivity	83.3%	0.0%	100%	36.5%	100%	80.2%	57.1%	100%	0.0%	100%	0.0%	**100%**	**100%**
Specificity	25.0%	100%	0.0%	54.8%	0.0%	20.2%	75.6%	0.6%	100%	0.0%	100%	**47.6%**	**63.5%**

Table 4.1: The deep learning classification result of rTMS response, the accuracy, sensitivity, and specificity are validation results.

doi:10.6342/NTU202101201

Biomarker	Frontal EEG signal												
Treatment	iTBS												
Patients (N)	37 (OPD dataset)												
EEG Features	Neural Network extracted nonlinear features on Delta, Theta, Alpha, Beta, Gamma bands												
Classification	CNN							RNN	RNN-CNN				ACNN
	CNN	FCN	MCDCNN	ResNet	TLENT	Inception Time	Encoder	TWIESN	LSTM-FCN	ALSTM-FCN	MALSTM-FCN	TapNet	**ACTS Net**
Accuracy	66.0%	45.9%	54.4%	55.9%	54.3%	64.4%	57.5%	54.4%	45.6%	54.4%	45.6%	**68.4%**	**93.6%**
Sensitivity	100%	100%	0.0%	100%	0.0%	39.1%	48.3%	0.0%	100%	0.0%	100%	**97.4%**	**97.4%**
Specificity	37.1%	0.0%	100%	18.6%	100%	86.1%	65.4%	98.9%	0.0%	100%	0.0%	**44.1%**	**90.3%**

$Table\ 4.2$: The deep learning classification result of iTBS response, the accuracy, sensitivity, and specificity are validation results.

long short term memory-fully convolution network (LSTM-FCN), attention long short term memory-fully convolution network (ALSTM-FCN) [80], multivariate attention long short term memory-fully convolution network (MALSTM-FCN) [80], and multivariate time series classification with attentional prototypical network (TapNet) [77] both applied on the rTMS and iTBS data only in OPD dataset due to our graphics processing unit (GPU) random access memory (RAM) and our server memory issues, the input tensor size is $(N_{patients} \times N_{electrodes} \times 15000)$, where $N_{patients}$ is the number of the patients in the dataset and $N_{electrodes}$ is the number of the selected electrodes, in this work, $N_{electrodes} = 7$ included Fp1, Fp2, F3, F4, F7, F8, and Fz frontal electrodes. The 15000 data points per patient is from the maximum storage space of the server I worked with, the 15000 data points are selected continuously in the EEG data after the preprocessing. The results are shown in the Table 4.1 and Table 4.2.

We can get the findings that TMS MDD EEG data both on the rTMS and the iTBS do not preforms well on the RNN based models, and ACTSNet really preforms better than all of the deep learning models on these datasets. The models are trained by default hyperparameters, the CNNs are modified from [110], the RNNs are modified from [111], the TapNet model downloaded from [112]. The ACTSNet model writed with pytorch and the learning rate is 10^{-5}, the L2 loss on weight decay starts from 10^{-3}, the stop threshold for the training error is 10^{-9}, the filters used for convolutional network is $(256,156,128)$, the kernels used for convolutional network is (8,5,3), the parameters for random projection is -1, the sub-dimension for each random projection is 3, the metric parameter for prototype distance between classes is 10^{-2}, the dropout rate is 0, the dimension of Encoder embedding is 512. In addition, because of the reduction of the model parameters, ACTSNet spent less time on training than TapNet did. The results show in Table 4.3. All results in this chapter in the tables are validation results.

doi:10.6342/NTU202101201

Treatment	rTMS		iTBS	
Patients	30 (OPD dataset)		37 (OPD dataset)	
Model	TapNet	**ACTSNet**	TapNet	**ACTSNet**
Training Time	39.94s	**26.52s**	95.69s	**21.75s**
Accuracy	76.3%	**83.5%**	68.4%	**93.6%**

Table 4.3: Training time comparison of TapNet and ACTSNet on MDD EEG data.

To proof that the disadvantage part of TapNet on MDD EEG data is its LSTM part, I designed the model architecture ablation experiments as below: Separate TapNet into “LSTM alone” and “CNN alone” two parts and trained them on MDD EEG data respectively. The result in Table 4.4 show that the TapNet really did not well on the LSTM part.

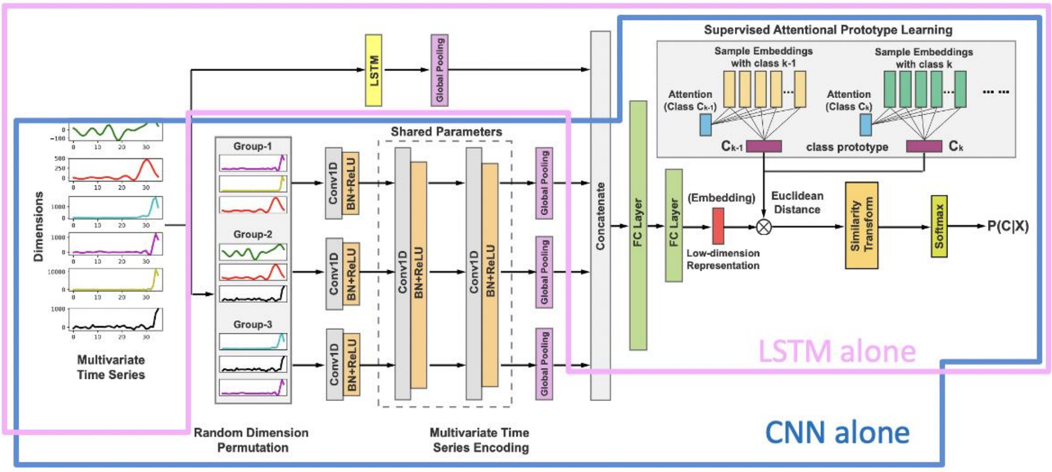


Figure 4.1: The sub-architecture of TapNet.

Treatment	rTMS		iTBS	
Patients	30 (OPD dataset)		37 (OPD dataset)	
Model	TapNet- LSTM alone	TapNet- CNN alone	TapNet- LSTM alone	TapNet- CNN alone
Accuracy	**54.6%**	74.1%	**45.6%**	67.3%
Sensitivity	100.0%	100.0%	100.0%	97.4%
Specificity	**0.0%**	42.9%	**0.0%**	41.9%

Table 4.4: The model architecture ablation experiment result.

4.2 Multivariate Time Series Classification (MTSC) Results

4.2.1 MTSC Results

In this section, we compared several MTSC machine learning algorithms with BOSS Ensemble. And we can see that BOSS Ensemble has the best performance both on the rTMS and iTBS OPD datasets from Table 4.5 and Table 4.6. All results in the tables are validation results.

Biomarker	Frontal EEG signal							
Treatment	rTMS							
Patients (N)	30 (OPD dataset)							
Features	Model extracted linear and Nonlinear features on Delta, Theta, Alpha, Beta, Gamma bands							
Classification	Distance-based		Shapelet-based		Interval-based	Dictionary-based		
	DTW	Catch22	Shapelet Transform	Mr-SEQL	RISE	SFA	WEASEL +MUSE	BOSS Ensemble
Accuracy	47.5%	62.6%	72.7%	76.9%	81.3%	72.7%	80.6%	**95.7%**
Sensitivity	75.0%	60.0%	68.3%	72.0%	75.0%	71.1%	76.3%	**93.8%**
Specificity	46.7%	78.4%	85.7%	89.7%	97.4%	75.5%	89.1%	**98.3%**

Table 4.5: The experiment results of MTSC models on the rTMS OPD dataset.

doi:10.6342/NTU202101201

Biomarker	Frontal EEG signal							
Treatment	iTBS							
Patients (N)	37 (OPD dataset)							
Features	Model extracted linear and Nonlinear features on Delta, Theta, Alpha, Beta, Gamma bands							
Classification	Distance-based		Shapelet-based		Interval-based	Dictionary-based		
	DTW	Catch22	Shapelet Transform	Mr-SEQL	RISE	SFA	WEASEL +MUSE	BOSS Ensemble
Accuracy	61.4%	58.5%	53.8%	59.6%	84.2%	66.7%	84.2%	**97.7%**
Sensitivity	92.3%	56.1%	0.0%	71.4%	82.3%	64.8%	84.9%	**96.3%**
Specificity	58.9%	59.6%	54.1%	58.0%	85.9%	68.0%	83.7%	**98.9%**

$Table\ 4.6$: The experiment results of MTSC models on the iTBS OPD dataset.

4.2.2 BOSS Ensemble vs. ACTSNet

Because of the success of the BOOSS Ensemble on the MDD EEG datasets, we compared BOSS Ensemble with ACTSNet, the results which are shown as Table 4.7.

Treatment	rTMS		iTBS	
Patients	30 (OPD dataset)		37 (OPD dataset)	
Model	BOSS Ensemble	**ACTSNet**	BOSS Ensemble	**ACTSNet**
Training Time	242.42hr	**26.52s**	237.61hr	**21.75s**
Accuracy	**95.7%**	83.5%	**97.7%**	93.6%

$Table\ 4.7$: The training time comparison of BOSS Ensemble and ACTSNet.

From the Table 4.7 above, we can observe that although the validation accuracy of the ACTSNet is not better than BOSS Ensemble, the training time of ACTSNet is almost 32,907 times faster than BOSS Ensemble on the rTMS OPD dataset and 39,328 times faster on the iTBS OPD dataset.

doi:10.6342/NTU202101201

4.3 The Statistically Significant features for Machine Learning on OPD Dataset

In this section we will show the statistically significant features selected after one-way AVOVA of OPD dataset both on rTMS and iTBS patients' EEG data, and these features were used to train the machine learning and the quantum machine learning models.

4.3.1 OPD Dataset

rTMS OPD Dataset		
Sub-band	**Electrode-Feature**	**P-Value**
Alpha	F3-KFD	0.0270*
Beta	F7-HFD	0.0135*
Beta	FP2-ApEn	0.0325*
Beta	F7-ApEn	0.0296*
Beta	Fz-ApEn	0.0355*
Beta	F3-LLE	0.0098**
Beta	Fz-LLE	0.0112*
Beta	F4-LLE	0.0325*
Beta	F7-DFA	0.0246*
Delta	Fz-Welch	0.0166*
Delta	F4-Welch	0.0150*
Delta	FP1-ApEn	0.0355*
Delta	F4-DFA	0.0166*
Theta	FP2-Welch	0.0203*

doi:10.6342/NTU202101201

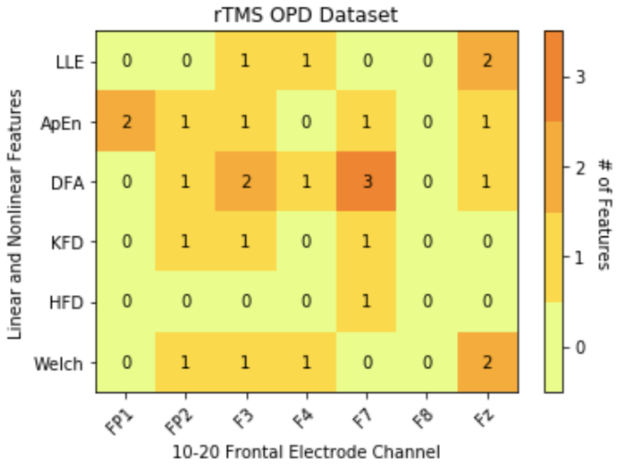
Theta	F3-Welch	0.0166*
Theta	Fz-Welch	0.0462*
Theta	F3-ApEn	0.0424*
Theta	Fz-LLE	0.0166*
Theta	FP2-DFA	0.0424*
Theta	F7-DFA	0.0296*
Theta	F3-DFA	0.0424*
Theta	Fz-DFA	0.0184*
Gamma	FP2-KFD	0.0424*
Gamma	F7-KFD	0.0246*
Gamma	FP1-ApEn	0.0122*
Gamma	F7-DFA	0.0135*
Gamma	F3-DFA	0.0462*

Table 4.8: The statistically significant features of the rTMS OPD data.

Table 4.8 shows the statistically significant features of the rTMS OPD data after the Mann-Whitney U test, a kind of one-way ANOVA test (all the * means p<0.05 and ** means p<0.01 in the tables). From the Figure 4.2, we can see that on this MDD rTMS dataset, F3, F7, and Fz channels have more significant features than other channels, and the DFA nonlinear feature has more significant features than other features. From the Figure 4.3, Figure 4.4 and Table 4.9, the results have shown that on this rTMS OPD dataset, the theta and gamma band may can be the digital biomarker to predict the response of the rTMS.

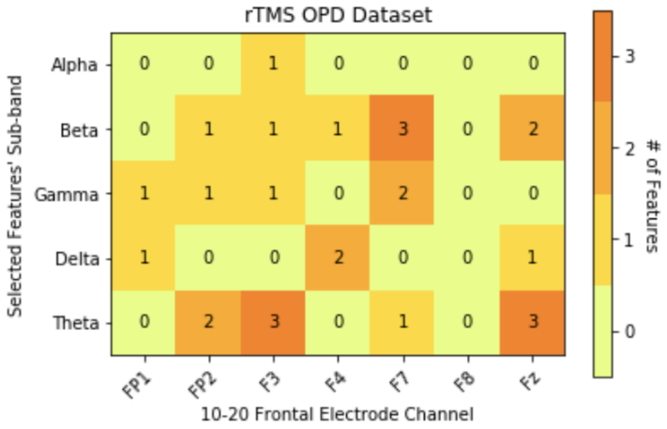
Channel-Feature	Sub-band
FP1-ApEn	Delta, Gamma
F3-DFA	Theta, Gamma
F7-DFA	Beta, Theta, Gamma
Fz-LLE	Beta, Theta
Fz-Welch	Delta, Theta

Table 4.9: The recurring statistically significant features in the rTMS OPD data.



	FP1	FP2	F3	F4	F7	F8	Fz
LLE	0	0	1	1	0	0	2
ApEn	2	1	1	0	1	0	1
DFA	0	1	2	1	3	0	1
KFD	0	1	1	0	1	0	0
HFD	0	0	0	0	1	0	0
Welch	0	1	1	1	0	0	2

Figure 4.2: The significant features-channels heatmap of the rTMS OPD dataset.



	FP1	FP2	F3	F4	F7	F8	Fz
Alpha	0	0	1	0	0	0	0
Beta	0	1	1	1	3	0	2
Gamma	1	1	1	0	2	0	0
Delta	1	0	0	2	0	0	1
Theta	0	2	3	0	1	0	3

Figure 4.3: The sub-bands-channels heatmap of the rTMS OPD dataset.

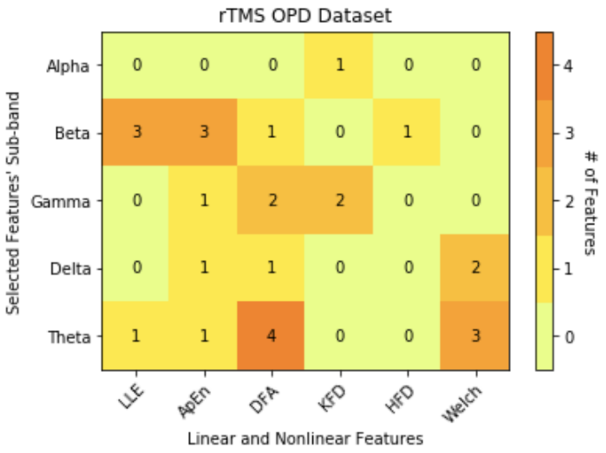


Figure 4.4: The sub-bands-significant features heatmap of the rTMS OPD dataset.

iTBS OPD Dataset		
Sub-band	**Electrode-Feature**	**P-Value**
Alpha	F3-Welch	0.0247*
Alpha	Fz-Welch	0.0454*
Beta	Fz-DFA	0.0015**
Beta	F4-DFA	0.0042**
Beta	F8-DFA	0.0454*
Delta	F3-KFD	0.0454*
Delta	F7-HFD	0.0213*
Delta	F3-HFD	0.0084**
Delta	FP1-ApEn	0.0349*
Delta	F7-LLE	0.0425*
Delta	F3-LLE	0.0373*
Delta	Fz-DFA	0.0425*
Theta	F7-KFD	0.0454*
Theta	F4-KFD	0.0230*
Theta	F7-ApEn	0.0091**
Gamma	FP1-DFA	0.0326*
Gamma	F3-DFA	0.0198*
Gamma	Fz-DFA	0.0326*
Gamma	F4-DFA	0.0425*

Table 4.10: The statistically significant features of the iTBS OPD data.

doi:10.6342/NTU202101201

Table 4.10 shows the statistically significant features of the iTBS OPD data after the Mann-Whitney U test. From the Figure 4.5, we can see that on this MDD iTBS dataset, F3, F7, and Fz channels have more significant features than other channels, and the DFA nonlinear feature has more significant features than other features, especially in Fz-DFA. From the Figure 4.6, Figure 4.7 and Table 4.11, the results have shown that on this rTMS OPD dataset, the delta and gamma band may can be the digital biomarker to predict the response of the iTBS, especially in beta-DFA and gamma-DFA.

Channel-Feature	Sub-band
F4-DFA	Beta, Gamma
Fz-DFA	Beta, Delta, Gamma

Table 4.11: The recurring statistically significant features in the iTBS OPD data.

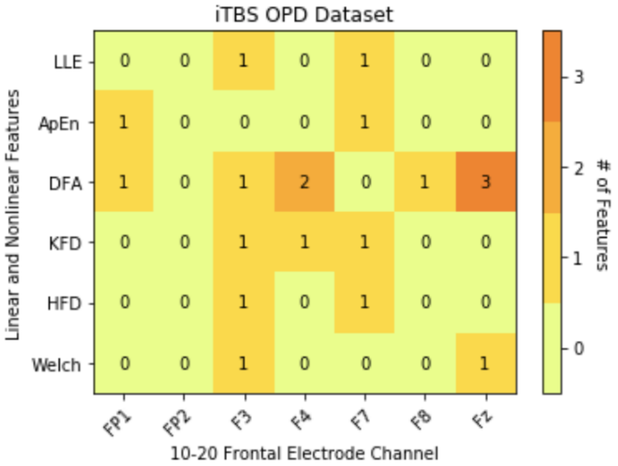


Figure 4.5: The significant features-channels heatmap of the iTBS OPD dataset.

doi:10.6342/NTU202101201

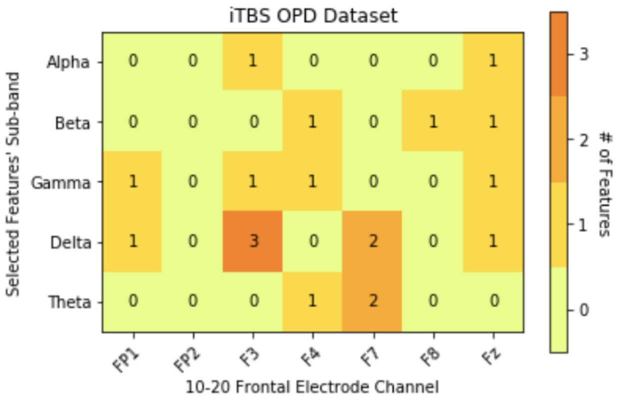


Figure 4. 6: The sub-bands-channels heatmap of the iTBS OPD dataset.

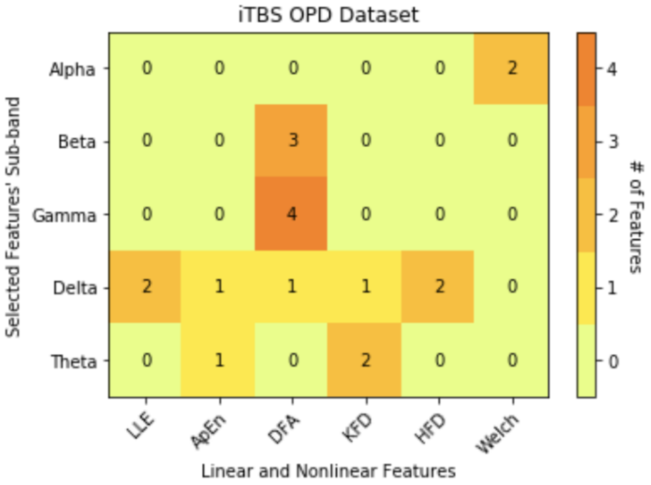


Figure 4. 7: The sub-bands-significant features heatmap of the iTBS OPD dataset.

4.3.2 Mixed Dataset

	rTMS Mixed Dataset	
Sub-band	**Electrode-Feature**	**P-Value**
Alpha	FP1-Welch	0.0091**
Alpha	FP2-Welch	0.0281*
Alpha	F7-Welch	0.0120*
Alpha	F3-Welch	0.0237*
Alpha	F4-KFD	0.0066**
Alpha	F4-LLE	0.0301*

doi:10.6342/NTU202101201

Alpha	F4-DFA	0.0365*
Beta	F7-Welch	0.0343*
Beta	F3-Welch	0.0263*
Beta	FP1-ApEn	0.0498*
Beta	F7-ApEn	0.0091**
Beta	F7-LLE	0.0263*
Beta	F3-LLE	0.0311*
Beta	F4-LLE	0.0072**
Beta	F8-LLE	0.0237*
Beta	F3-DFA	0.0377*
Beta	Fz-DFA	0.0144*
Delta	Fz-KFD	0.0229*
Delta	F4-KFD	0.0042**
Delta	FP1-HFD	0.0343*
Delta	FP2-HFD	0.0272*
Delta	FP1-ApEn	0.0414*
Delta	FP1-LLE	0.0173*
Delta	F3-LLE	0.0134*
Theta	FP2-KFD	0.0254*
Theta	FP1-HFD	0.0498*
Theta	F3-HFD	0.0081**
Theta	Fz-HFD	0.0069**
Theta	F4-HFD	0.0498*
Theta	F7-LLE	0.0144*
Theta	F3-LLE	0.0047**
Theta	Fz-LLE	0.0072**
Theta	F4-LLE	0.0134*
Theta	FP1-DFA	0.0150*
Theta	FP2-DFA	0.0179*
Theta	F3-DFA	0.0173*
Theta	Fz-DFA	0.0012**
Theta	F4-DFA	0.0173*
Theta	F8-DFA	0.0354*

doi:10.6342/NTU202101201

Gamma	FP1-Welch	0.0254*
Gamma	F7-Welch	0.0084**
Gamma	F3-Welch	0.0263*
Gamma	FP2-HFD	0.0229*
Gamma	F7-HFD	0.0483*
Gamma	F4-HFD	0.0107*

$Table\ 4.12$: The statistically significant features of the rTMS mixed data.

Table 4.12 shows the statistically significant features of the rTMS mixed data after the Mann-Whitney U test. From the Figure 4.8, we can see that on this rTMS mixed dataset, FP1, F3, F4 and F7 channels have more significant features than other channels, and the LLE, DFA, HFD nonlinear features and Welch linear feature have more significant features than other features, especially in LLE. From the Figure 4.9, Figure 4.10 and Table 4.13, the results have shown that on this rTMS mixed dataset, the beta and theta band may can be the digital biomarker to predict the response of the iTBS, especially in theta-DFA.

doi:10.6342/NTU202101201

Channel-Feature	Sub-band
FP1-Welch	Alpha, Gamma
FP1-HFD	Delta, Theta
FP2-HFD	Delta-Gamma
F3-Welch	Alpha, Beta, Gamma
F3-LLE	Beta, Delta, Theta
F3-DFA	Beta, Theta
F4-LLE	Alpha, Beta, Theta
F4-DFA	Alpha, Theta
F4-KFD	Alpha, Delta
F4-HFD	Theta, Gamma
F7-Welch	Alpha, Beta, Gamma
F7-LLE	Beta, Theta
Fz-DFA	Beta, Theta

$Table\ 4.13$: The recurring statistically significant features in the rTMS mixed data.

doi:10.6342/NTU202101201

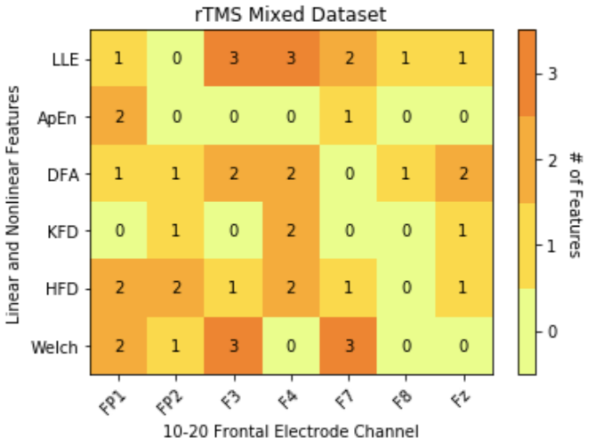


Figure 4.8: The significant features-channels heatmap of the rTMS mixed dataset.

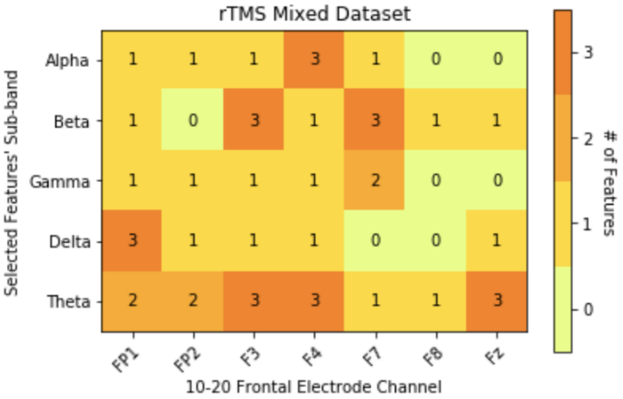


Figure 4.9: The sub-bands-channels heatmap of the rTMS mixed dataset.

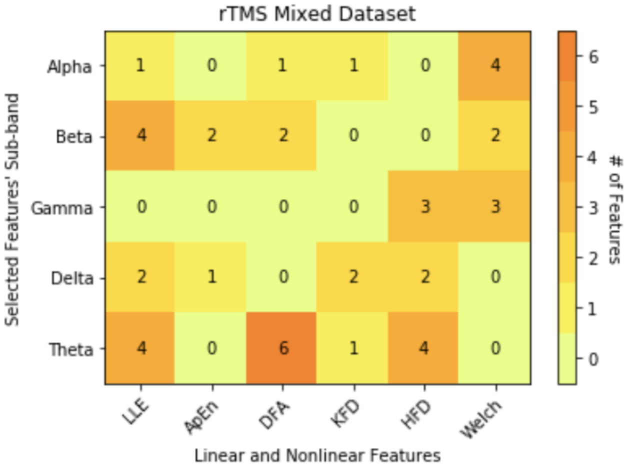


Figure 4.10: The sub-bands-significant features heatmap of the rTMS mixed dataset.

doi:10.6342/NTU202101201

	iTBS Mixed Dataset	
Sub-band	**Electrode-Feature**	**P-Value**
Alpha	F3-KFD	0.0313*
Alpha	FP2-HFD	0.0380*
Alpha	F4-HFD	0.0423*
Alpha	F8-HFD	0.0093**
Beta	FP2-Welch	0.0134*
Beta	Fz-Welch	0.0130*
Beta	F4-Welch	0.0058**
Beta	F8-Welch	0.0496*
Delta	F7-Welch	0.0458*
Delta	F3-HFD	0.0202*
Delta	F7-ApEn	0.0401*
Delta	F3-ApEn	0.0093**
Delta	F3-DFA	0.0279*
Theta	Fz-KFD	0.0360*
Gamma	FP1-Welch	0.0412*
Gamma	FP2-Welch	0.0221*
Gamma	F3-Welch	0.0263*
Gamma	Fz-Welch	0.0256*
Gamma	F4-Welch	0.0360*
Gamma	F8-Welch	0.0110*

Table 4.14: The statistically significant features of the iTBS mixed data.

Channel-Feature	**Sub-band**
FP2-Welch	Beta, Gamma
F4-Welch	Beta, Gamma
F8-Welch	Beta, Gamma
Fz- Welch	Beta, Gamma

Table 4.15: The recurring statistically significant features in the iTBS mixed data.

doi:10.6342/NTU202101201

Table 4.14 shows the statistically significant features of the iTBS mixed data after the Mann-Whitney U test. From the Figure 4.11, we can see that on this iTBS mixed dataset, F3 channel has more significant features than other channels, and the HFD nonlinear feature and Welch linear feature have more significant features than other features, especially in F3-delta. From the Figure 4.12, Figure 4.13 and Table 4.15, the results have shown that on this iTBS mixed dataset, the delta and gamma band may can be the digital biomarker to predict the response of the iTBS, especially in beta-Welch and gamma-Welch.

To sum up, this section described the statistics findings in this thesis both on the rTMS and the iTBS MDD EEG datasets. On the rTMS datasets, the common significant features are on the F3, F4 and F7 channels; DFA and Welch features; beta and theta bands. On the iTBS datasets, the common significant features are on the F3, channel; HFD and Welch features; delta and gamma bands.

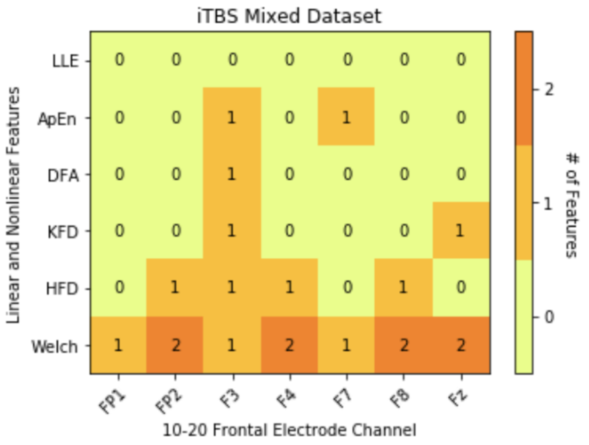


Figure 4. 11: The significant features-channels heatmap of the iTBS mixed dataset.

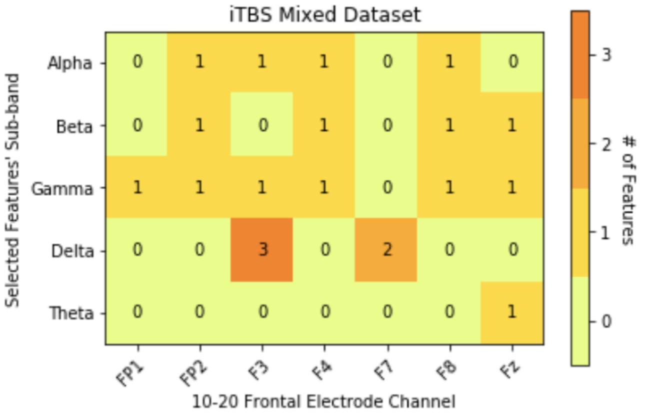


Figure 4. 12: The sub-bands-channels heatmap of the iTBS mixed dataset.

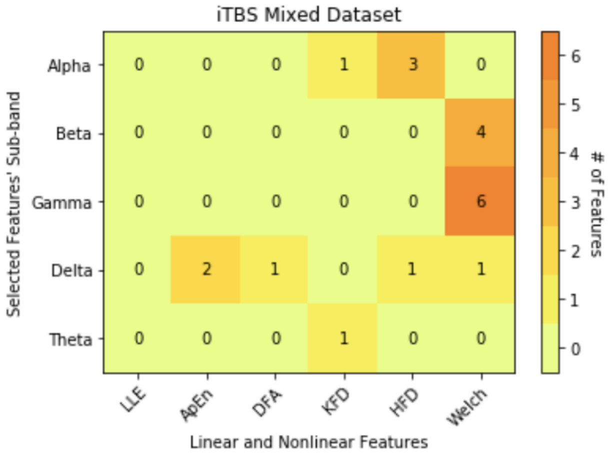


Figure 4. 13: The sub-bands-significant features heatmap of the iTBS mixed dataset.

4.3.3 Dataset Comparison

In this section we compared the OPD dataset and the previous dataset on the statistically significant features numbers of all sub-bands.

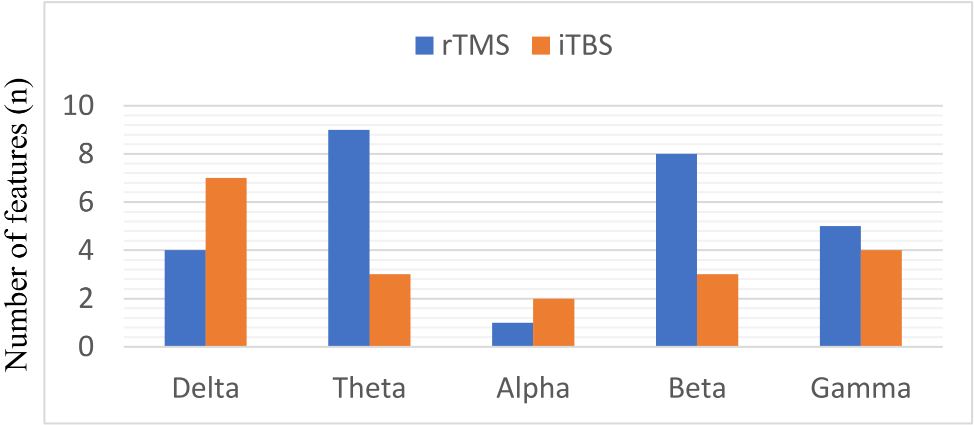


Figure 4. 14: Number of features under different band with p<0.05 of OPD dataset.

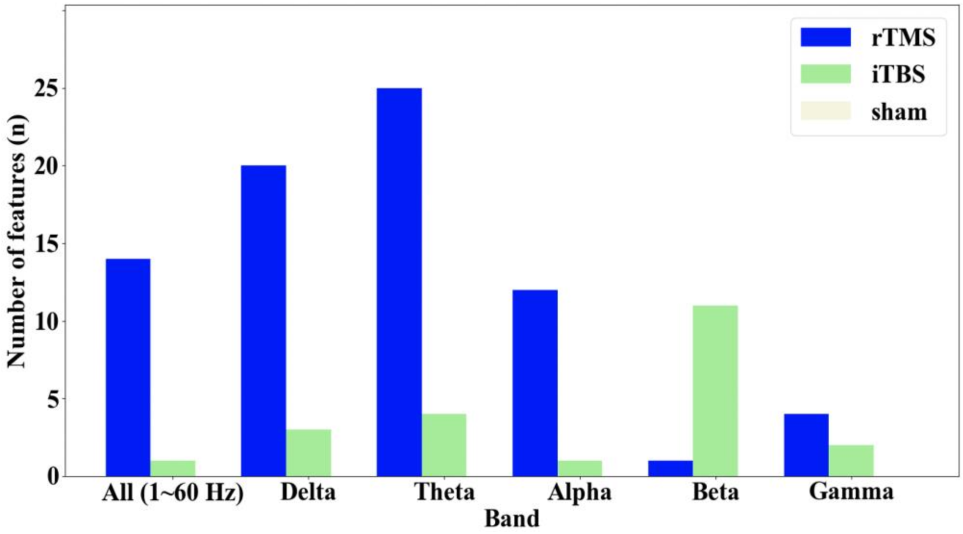


Figure 4. 15: Number of features under different band with p<0.05 of previous dataset [31].

We can observe that on the OPD dataset, these number of the features with p<0.05 on different sub-bands are very different. On the both datasets, the number of the features

doi:10.6342/NTU202101201

with $p<0.05$ in the theta band of rTMS datasets are both more than it on iTBS datasets, it proved that RECT test can really work to enhance the power of the theta band. On the OPD dataset, the statistically significant features of the delta band, and alpha band of iTBS is more than rTMS, and on the previous dataset, this phenomenon only observe on the beta band. The reason may be the research data (previous dataset) characteristics is relatedly ideal and the OPD dataset is the data from the real-world dataset has more heterogeneity.

doi:10.6342/NTU202101201

4.4 Classical Machine Learning Results

4.4.1 Classical Machine Learning Results

4.4.1.1 Compare with Linear Method and SVM on the Clinical Trial Dataset

Since SVM is the most widely used nonlinear kernel method classification model with good predictive effects in many fields on time series prediction and classification [179], we use EEG data to calculate and extract 5 non-linear features include LLE, DFA, KFD, HFD, ApEn, and a linear feature Welch. In this thesis, all the SVM models are trained with nonlinear radial basis function (rbf) kernel. The linear and nonlinear features are jointly trained the machine learning models, and the classification effect is compared with the linear multivariate analysis and prediction classification method, Logistic Regression, which is common in traditional statistics.

The new-trained results for the rTMS, iTBS, and sham control groups of clinical trial data are as follows:

doi:10.6342/NTU202101201

Biomarker	Frontal EEG signal		
Treatment	rTMS		
Patients (N)	32 (RCTD dataset)		
Features	Only Theta	Linear and Nonlinear features on Delta, Theta, Alpha, Beta, Gamma bands	
Classification	Linear method	Nonlinear Method (new trained on previous data for the information in [31] is lacking)	
	Logistic Regression (Optimal Threshold)	Previous work SVM [31]	SVM_{new}
Accuracy	33.3%	91.1%	91.7%
Sensitivity	16.7%	83.3%	100.0%
Specificity	66.7%	95.0%	75.0%

Table 4.16: The comparison results of linear versus nonlinear methods on rTMS clinical trial dataset.

Biomarker	Frontal EEG signal	
Treatment	iTBS	
Patients (N)	30 (RCTD dataset)	
Features	Only Theta	Linear and Nonlinear features on Delta, Theta, Alpha, Beta, Gamma bands
Classification	Linear method	Nonlinear Method
	Logistic Regression (Optimal Threshold)	SVM
Accuracy	44.4%	58.3%
Sensitivity	80.0%	57.1%
Specificity	0.0%	60.0%

Table 4.17: The comparison results of linear versus nonlinear methods on iTBS clinical trial dataset.

Biomarker	Frontal EEG signal	
Treatment	Sham control	
Patients (N)	28 (RCTD dataset)	
Features	Only Theta	Linear and Nonlinear features on Delta, Theta, Alpha, Beta, Gamma bands
Classification	Linear method	Nonlinear Method
	Logistic Regression (Optimal Threshold)	SVM
Accuracy	90.9%	100.0%
Sensitivity	90.9%	100.0%
Specificity	nan	nan

Table 4.18: The comparison results of linear versus nonlinear methods on sham clinical trial dataset.

The above experimental results (Table 4.16-Table 4.18) prove that the results of nonlinear machine learning analysis and prediction are better than traditional linear methods, and the results of SVM machine learning algorithm are better than those of linear analysis and prediction. The machine learning model can not only predict rTMS, but also predict the response effect of iTBS.

However, we found that in such SVM model, there is an overfitting phenomenon in the new clinical data, that is, it has a high accuracy rate in the original data set, but it performs poorly in the new clinical OPD EEG dataset (Table 4.19).

Biomarker	Frontal EEG signal			
Treatment	rTMS			
Patients (N)	32 (RCTD dataset)		30 (OPD dataset)	
Features	Linear and Nonlinear features on Delta, Theta, Alpha, Beta, Gamma bands			
Classification	Previous work		Previous work saved model	
	SVM	SVM$_{new}$	SVM test on OPD	SVM$_{new}$ test on OPD
Accuracy	91.1%	91.7%	**50.0%**	**48.4%**
Sensitivity	83.3%	100.0%	**36.3%**	**73.7%**
Specificity	95.0%	75.0%	**57.9%**	**8.3%**

Table 4.19: The overfitting results of the SVM trained on clinical trial dataset.

4.4.1.2 SVM and the Overfitting Problem

In order to increase the robustness and generalization of the model, we added clinical OPD data to form a larger sample data set for SVM model training. Because clinical OPD data is clinical data, there is no control in the group, the heterogeneity among patients is higher than that of the original clinical trial data, which is closer to the actual clinical situation. Therefore, we train the machine learning model by adding clinical OPD data and clinical trial data to enrich the data sample space expectation to avoid overfitting to a certain extent and obtain a more accurate SVM model. The computational experiment results are in the Table 4.20. (The iTBS clinical trial data results are new-trained on the same dataset, for the previous work [31] do not record the TPR and TNR of this SVM model.)

Biomarker	Frontal EEG signal		Frontal EEG signal	
Treatment	rTMS	iTBS	rTMS	iTBS
Patient (N)	32（RCTD dataset）	30（RCTD dataset）	62（RCTD dataset + OPD dataset）	67（RCTD dataset + OPD dataset）
Feature	Linear and Nonlinear		Linear and Nonlinear	
EEG Band	Delta, Theta, Alpha, Beta, Gamma		Delta, Theta, Alpha, Beta, Gamma	
Classification	SVM	SVM	SVM	SVM
Accuracy	91.7%	58.3%	57.9%	40.0%
Sensitivity	100.0%	57.1%	75.0%	30.0%
Specificity	75.0%	60.0%	28.6%	50.0%

$Table\ 4.20$: The experiment results of SVM on the mixed dataset.

The above results can prove that the SVM model does overfitting on highly complex data such as depression EEG, so we need to find better machine learning algorithms that can learn well from the datasets. According to [45] and [46], in the comparison of different model training in major public data sets, they found that the machine learning algorithms based on bagging and boosting algorithms have better generalization ability and can reduce overfitting compared to SVM, so we choose these types of algorithms to predict the efficacy of rTMS and iTBS.

4.4.1.3 Compare with Previous Work on the Clinical Trial Dataset

To proof that bagging and boosting machine learning algorithms are better than support vector machine (SVM), we executed the computational experiments on the previous datasets. The results are shown in Table 4.21 and Table 4.22. We can observe that on the rTMS previous dataset (clinical trial dataset), only the random forest can perform better than SVM, but in the case of the iTBS previous dataset, random forest, XGBoost and CatBoost are more accurate than SVM on the same data and extracted feature set. The result of the rTMS previous dataset can be interpreted as the feature set is maybe underfitting on the boosting algorithms.

Biomarker	Frontal EEG signal			
Treatment	rTMS			
Patients (N)	32 (RCTD dataset)			
Features	Linear and Nonlinear features on Delta, Theta, Alpha, Beta, Gamma bands			
Classification		This work		
	SVM	Random Forest	XGBoost	CatBoost
Accuracy	91.7%	**91.6%**	88.9%	75.0%
Sensitivity	100.0%	100.0%	100.0%	80.0%
Specificity	75.0%	85.7%	83.3%	71.4%

Table 4.21: The classical machine learning results of the rTMS previous dataset.

Biomarker	Frontal EEG signal			
Treatment	iTBS			
Patients (N)	30 (RCTD dataset)			
Features	Linear and Nonlinear features on Delta, Theta, Alpha, Beta, Gamma bands			
Classification		This work		
	SVM	Random Forest	XGBoost	CatBoost
Accuracy	58.3%	**90.9%**	**89.0%**	**77.8%**
Sensitivity	57.1%	83.3%	100.0%	83.3%
Specificity	60.0%	100.0%	80.0%	66.7%

Table 4.22: The classical machine learning results of the iTBS previous dataset.

4.4.1.4 Compare with Previous Work on the OPD Dataset

In this thesis, different from previous work collected the study data (previous dataset, or clinical trial dataset), we collected real-world clinical data from outpatient clinic (clinical OPD dataset). To compare with the SVM method in the previous work, we also ran the SVM with radial basis function kernel which the same as the previous work on the OPD dataset.

Biomarker	Frontal EEG signal				
Treatment	rTMS				
Patients (N)	32 (RCTD dataset)	30 (OPD dataset)			
Features	Linear and Nonlinear features on Delta, Theta, Alpha, Beta, Gamma bands				
Classification	Previous data	This work			
	SVM	SVM	Random Forest	XGBoost	CatBoost
Accuracy	91.7%	53.8%	**84.6%**	**76.9%**	**53.8%**
Sensitivity	100.0%	62.5%	**80.0%**	**72.7%**	**58.3%**
Specificity	75.0%	40.0%	**100.0%**	**100.0%**	0.0%

Table 4.23: The classical machine learning results of the rTMS OPD dataset.

Biomarker	Frontal EEG signal				
Treatment	iTBS				
Patients (N)	30 (RCTD dataset)	37 (OPD dataset)			
Features	Linear and Nonlinear features on Delta, Theta, Alpha, Beta, Gamma bands				
Classification	Previous data	This work			
	SVM	SVM	Random Forest	XGBoost	CatBoost
Accuracy	58.3%	54.5%	**92.0%**	**72.7%**	**63.6%**
Sensitivity	57.1%	60.0%	**88.2%**	**66.7%**	**57.1%**
Specificity	60.0%	50.0%	**94.2%**	**80.0%**	**75.0%**

Table 4.24: The classical machine learning results of the iTBS OPD dataset.

doi:10.6342/NTU202101201

From the Table 4.23 and Table 4.24, we can see the results that the bagging (Random Forest) and boosting (XGBoost and CatBoost) methods are all better than the SVM method on the OPD dataset.

4.4.1.5 Compare with Previous Work on the Mixed Dataset

Biomarker	Frontal EEG signal				
Treatment	rTMS				
Patients (N)	32 (RCTD dataset)	62 (RCTD dataset + OPD dataset)			
Features	Linear and Nonlinear features on Delta, Theta, Alpha, Beta, Gamma bands				
Classification	Previous data	This work			
	SVM	SVM	Random Forest	XGBoost	CatBoost
Accuracy	91.7%	57.9%	**78.9%**	**73.7%**	**73.7%**
Sensitivity	100.0%	75.0%	**83.3%**	**73.3%**	**83.3%**
Specificity	75.0%	28.9%	71.4%	**75.0%**	57.1%

Table 4.25: The classical machine learning results of the rTMS mixed dataset.

Biomarker	Frontal EEG signal				
Treatment	iTBS				
Patients (N)	30 (RCTD dataset)	67 (RCTD dataset + OPD dataset)			
Features	Linear and Nonlinear features on Delta, Theta, Alpha, Beta, Gamma bands				
Classification	Previous work	This work			
	SVM	SVM	Random Forest	XGBoost	CatBoost
Accuracy	58.3%	40.0%	**80.0%**	**75.0%**	**60.0%**
Sensitivity	57.1%	30.0%	**75.0%**	**72.7%**	**66.7%**
Specificity	60.0%	50.0%	**87.5%**	**77.8%**	**57.1%**

Table 4.26: The classical machine learning results of the iTBS mixed dataset.

To have more data for machine learning model training, we mixed the previous dataset and OPD dataset and extracted the features. The models' training results are shown in Table 4.25 and Table 4.26. We can find that both on the rTMS and iTBS mixed dataset, the bagging and the boosting machine learning methods all performed well than SVM under the same feature set. The reason why they cause these phenomena maybe the previous dataset is too ideal to let the model to learn the real MDD features from the real-world EEG data.

4.4.2 Booster Transformation Results

In this section, we will prove by the computational experiments that the booster transformation I invented can really shift the distribution of the models' sensitivity (Sens) and specificity (Spec), and then change the models' accuracy (Acc) distribution on the same dataset. This mathematical skill can fine tune the model.

doi:10.6342/NTU202101201

4.4.2.1 Compare Booster with Non-Booster Methods on Random Forest with rTMS Mixed Dataset

For random forest model, compare with non–booster models, booster models which after 16300 trainings (the distribution were shown in Figure 4.16 and the detail is shown in Table 4.27) with the same dataset (rTMS mixed dataset) have some properties:

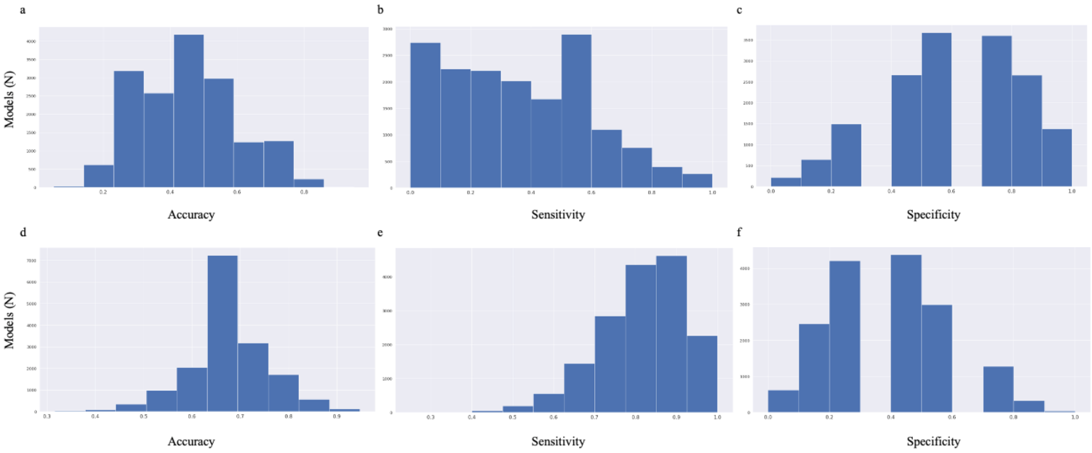


Figure 4.16: The model performance distribution of random forest-booster and original random forest.

The subfigure a, b, c is the accuracy, sensitivity, and specificity of the random forest-booster models on the rTMS mixed dataset, respectively. The subfigure d, e, f is the accuracy, sensitivity, and specificity of the original random forest models on the rTMS mixed dataset, respectively.

Performance of Random Forest	Booster (N)	Non-Booster (N)
$0.9 > Acc \geq 0.8$	65	685
$Acc \geq 0.9$	1	12
$0.9 > Sens \geq 0.8$	398	4347
$Sens \geq 0.9$	263	6884
$0.9 > Spec \geq 0.8$	2658	322
$Spec \geq 0.9$	1378	40

$Table\ 4.27$: The comparison detail of the random forest booster model performance distribution on rTMS dataset.

Random forest-booster can effectively reduce the average sensitivity, effectively increase the average specificity, and high accuracy models are 10 times less than non-booster models on the rTMS mixed dataset.

4.4.2.2 Compare Booster with Non-Booster methods on XGBoost with rTMS Mixed Dataset

For XGBoost model, compare with non–booster models, booster models which after 16700 trainings (the distribution were shown in Figure 4.17 and the detail is shown in Table 4.28) with the same dataset (rTMS mixed dataset) have some properties:

doi:10.6342/NTU202101201

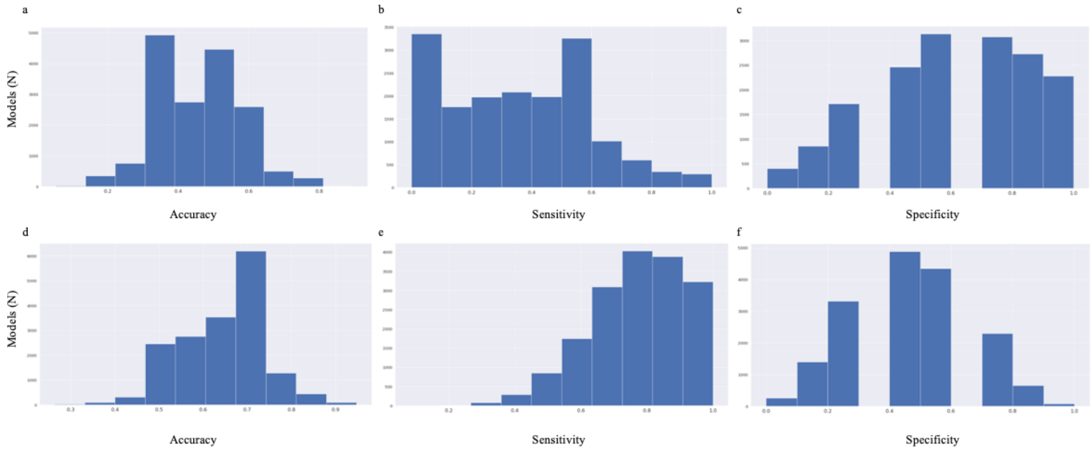


Figure 4.17: The model performance distribution of XGBoost-booster and original XGBoost.

The subfigure a, b, c is the accuracy, sensitivity, and specificity of the XGBoost-booster models on the rTMS mixed dataset, respectively. The subfigure d, e, f is the accuracy, sensitivity, and specificity of the original XGBoost models on the rTMS mixed dataset, respectively.

Performance of XGBoost	Booster (N)	Non-Booster (N)
$0.9 > Acc \geq 0.8$	12	540
$Acc \geq 0.9$	0	9
$0.9 > Sens \geq 0.8$	347	3884
$Sens \geq 0.9$	291	3231
$0.9 > Spec \geq 0.8$	2727	653
$Spec \geq 0.9$	2278	75

Table 4.28: The comparison detail of the XGBoost booster model performance distribution on rTMS dataset.

doi:10.6342/NTU202101201

XGBoost-booster can effectively reduce the average sensitivity, effectively increase the average specificity, and high accuracy models are 45 times less than non-booster models on the rTMS mixed dataset.

4.4.2.3 Compare Booster with Non-Booster Methods on CatBoost with rTMS Mixed Dataset

For CatBoost model, compare with non–booster models, booster models which after 16759 trainings (the distribution were shown in Figure 4.18 and the detail is shown in Table 4.29) with the same dataset (rTMS mixed dataset) have some properties:

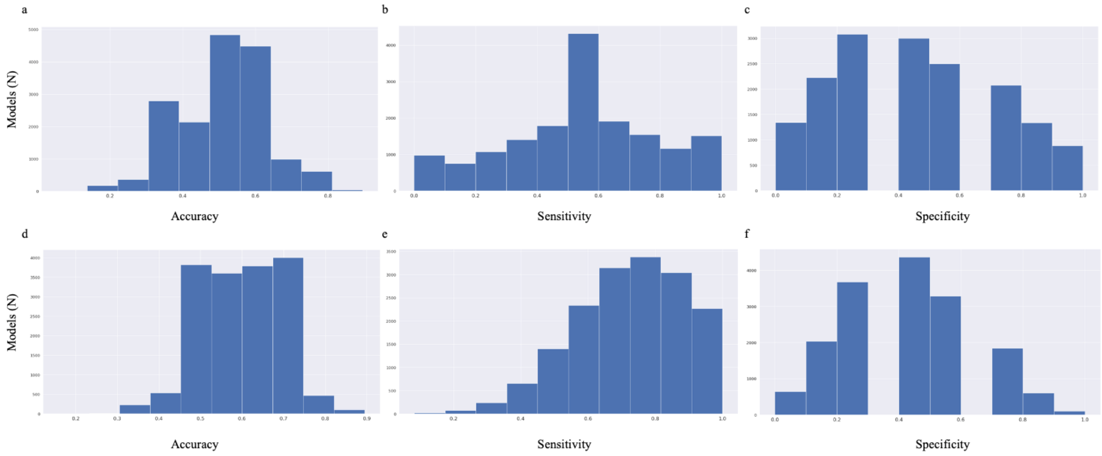


Figure 4. 18: The model performance distribution of CatBoost-booster and original CatBoost.

The subfigure a, b, c is the accuracy, sensitivity, and specificity of the CatBoost-booster models on the rTMS mixed dataset, respectively. The subfigure d, e, f is the accuracy, sensitivity, and specificity of the original CatBoost models on the rTMS mixed dataset, respectively.

Performance of CatBoost	Booster (N)	Non-Booster (N)
$0.9 > Acc \geq 0.8$	37	101
$Acc \geq 0.9$	0	0
$0.9 > Sens \geq 0.8$	1161	3042
$Sens \geq 0.9$	1510	2272
$0.9 > Spec \geq 0.8$	1335	599
$Spec \geq 0.9$	886	100

$Table\ 4.29$: The comparison detail of the CatBoost booster model performance distribution on rTMS dataset.

CatBoost-booster can effectively reduce the average sensitivity, effectively increase the average specificity, and high accuracy models are 3 times less than non-booster models on the rTMS mixed dataset. From Figure 4.16 to Figure 4.18, we can observe that booster transformation can shift the distribution both on bagging and boosting machine learning algorithms on the rTMS EEG data.

4.4.2.4 Compare Booster with Non-Booster Methods on Random Forest with iTBS Mixed Dataset

Similarly, for random forest model, compare with non–booster models, booster models which after 16708 trainings (the distribution were shown in Figure 4.19 and

doi:10.6342/NTU202101201

the detail is shown in Table 4.30) with the same dataset (iTBS mixed dataset) have some properties:

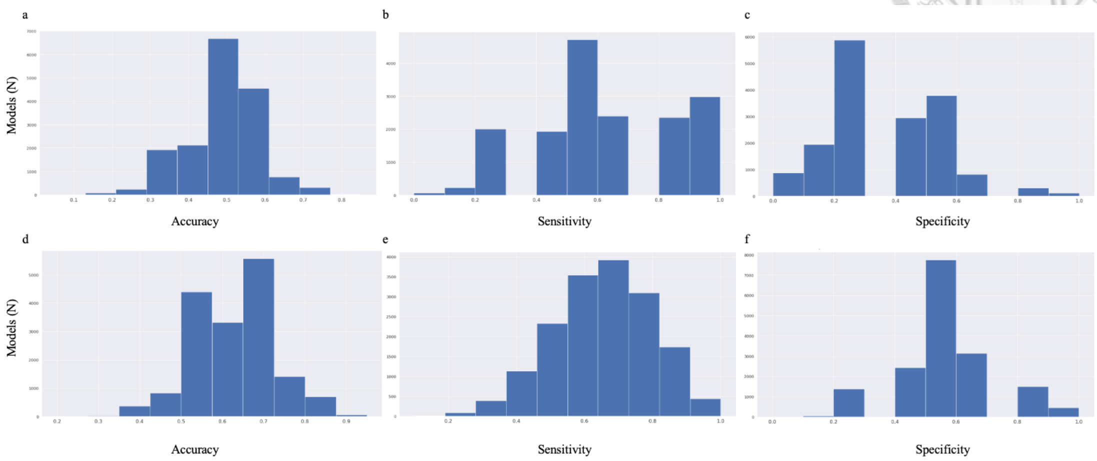


Figure 4.19: The model performance distribution of random forest-booster and original random forest.

The subfigure a, b, c is the accuracy, sensitivity, and specificity of the random forest-booster models on the iTBS mixed dataset, respectively. The subfigure d, e, f is the accuracy, sensitivity, and specificity of the original random forest models on the iTBS mixed dataset, respectively.

Performance of Random Forest	Booster (N)	Non-Booster (N)
$0.9 > Acc \geq 0.8$	1	158
$Acc \geq 0.9$	0	4
$0.9 > Sens \geq 0.8$	0	0
$Sens \geq 0.9$	1186	445
$0.9 > Spec \geq 0.8$	0	0
$Spec \geq 0.9$	16	451

Table 4.30: The comparison detail of the random forest booster model performance distribution on iTBS dataset.

doi:10.6342/NTU202101201

Random forest-booster can effectively reduce the average sensitivity, effectively increase the average specificity, and high accuracy models are 158 times less than non-booster models on the iTBS mixed dataset.

4.4.2.5 Compare Booster with Non-Booster Methods on XGBoost with iTBS Mixed Dataset

For XGBoost model, compare with non–booster models, booster models which after 16697 trainings (the distribution were shown in Figure 4.20 and the detail is shown in Table 4.31) with the same dataset (iTBS mixed dataset) have some properties:

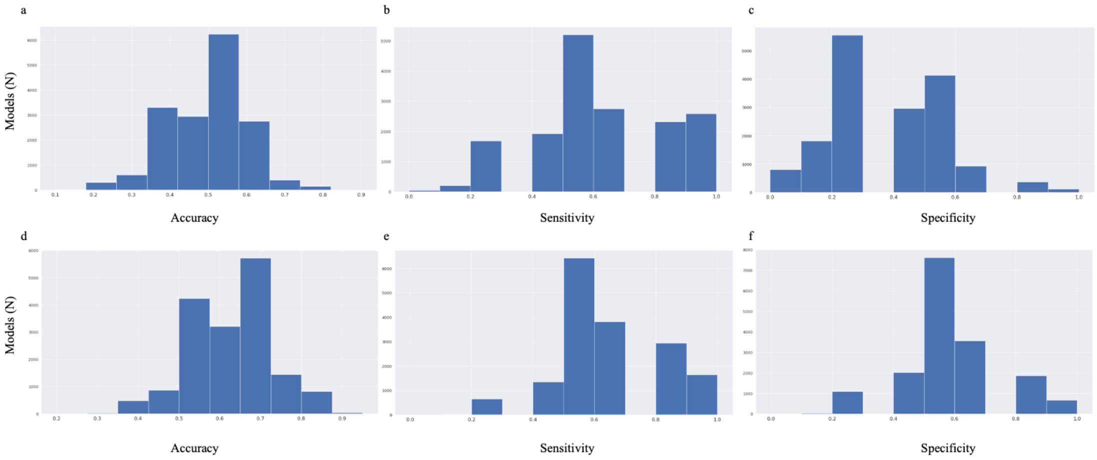


Figure 4.20: The model performance distribution of XGBoost-booster and original XGBoost.

The subfigure a, b, c is the accuracy, sensitivity, and specificity of the XGBoost-booster models on the iTBS mixed dataset, respectively. The subfigure d, e, f is the

doi:10.6342/NTU202101201

accuracy, sensitivity, and specificity of the original XGBoost models on the iTBS mixed dataset, respectively.

Performance of XGBoost	Booster (N)	Non-Booster (N)
$0.9 > Acc \geq 0.8$	6	178
$Acc \geq 0.9$	0	4
$0.9 > Sens \geq 0.8$	0	0
$Sens \geq 0.9$	927	264
$0.9 > Spec \geq 0.8$	0	0
$Spec \ \geq \ 0.9$	25	90

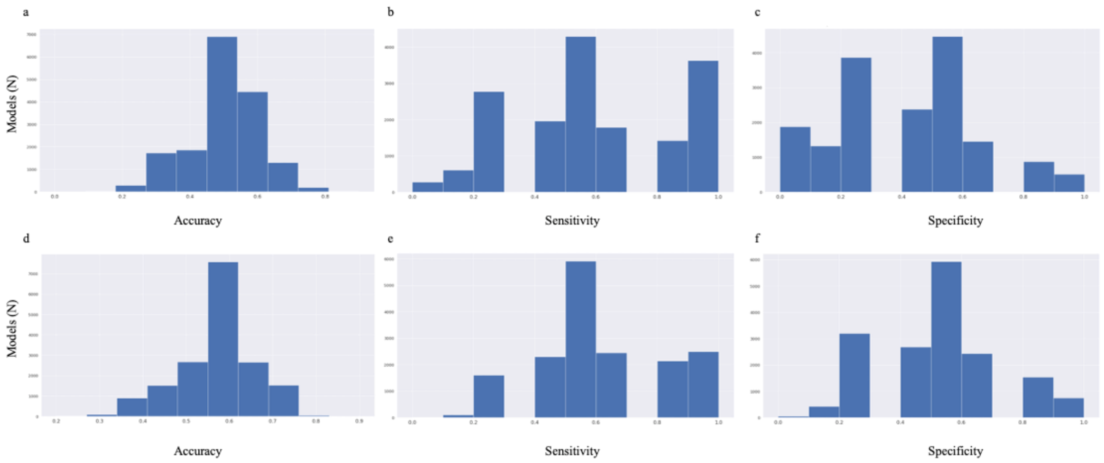
$Table\ 4.31$: The comparison detail of the XGBoost booster model performance distribution on iTBS dataset.

XGBoost-booster can effectively reduce the average sensitivity, effectively increase the average specificity, and high accuracy models are 29 times less than non-booster models on the iTBS mixed dataset.

4.4.2.6 Compare Booster with Non-Booster Methods on CatBoost with iTBS Mixed Dataset

For CatBoost model, compare with non–booster models, booster models which after 16759 trainings (the distribution were shown in Figure 4.21 and the detail is shown in Table 4.32) with the same dataset (iTBS mixed dataset) have some properties:

doi:10.6342/NTU202101201



$Figure\ 4.21$: The model performance distribution of CatBoost-booster and original CatBoost.

The subfigure a, b, c is the accuracy, sensitivity, and specificity of the CatBoost-booster models on the iTBS mixed dataset, respectively. The subfigure d, e, f is the accuracy, sensitivity, and specificity of the original CatBoost models on the iTBS mixed dataset, respectively.

Performance of CatBoost	Booster (N)	Non-Booster (N)
$0.9 > Acc \geq 0.8$	9	2
$Acc \geq 0.9$	0	0
$0.9 > Sens \geq 0.8$	0	0
$Sens \geq 0.9$	2516	973
$0.9 > Spec \geq 0.8$	0	0
$Spec\ \geq\ 0.9$	166	101

$Table\ 4.32$: The comparison detail of the CatBoost booster model performance distribution on iTBS dataset.

CatBoost-booster can effectively enhance the average sensitivity, effectively increase the average specificity, and high accuracy models are 4 times more than non-booster models on the iTBS mixed dataset.

From Figure 4.19 to Figure 4.21, we can observe that booster transformation can shift the distribution both on bagging and boosting machine learning algorithms on the iTBS EEG data.

doi:10.6342/NTU202101201

4.4.3 Booster-Classical Machine Learning Results

After introduced booster transformation, in this section we will show the application of booster transformation on the classical machine learning methods on the MDD EEG datasets.

4.4.3.1 Compare with Previous Work on the OPD Dataset

Table 4.33 and Table 4.34 shows the machine learning methods used booster transformation on the OPD dataset and compare the results with the previous work. From the Table 4.33 and Table 4.34, we can see that the sensitivity performance of the random forest-booster, XGBoost-booster, CatBoost-booster are all better than the original models especially on the iTBS OPD dataset.

Biomarker	Frontal EEG signal				
Treatment	rTMS				
Patients (N)	32 (RCTD dataset)	30 (OPD dataset)			
Features	Linear and Nonlinear features on Delta, Theta, Alpha, Beta, Gamma bands				
Classification	Previous data	This work			
	SVM	SVM-Booster	Random Forest-Booster	XGBoost-Booster	CatBoost-Booster
Accuracy	91.7%	**76.9%**	**90.0%**	**76.9%**	**61.5%**
Sensitivity	100.0%	**66.7%**	**85.7%**	**77.8%**	**100.0%**
Specificity	75.0%	**85.7%**	**100.0%**	75.0%	0.0%

Table 4.33: The experiment results of boostered machine learning methods on the OPD rTMS dataset.

doi:10.6342/NTU202101201

Biomarker	Frontal EEG signal				
Treatment	iTBS				
Patients (N)	32 (RCTD dataset)	37 (OPD dataset)			
Features	Linear and Nonlinear features on Delta, Theta, Alpha, Beta, Gamma bands				
Classification	Previous work	This work			
	SVM	SVM-Booster	Random Forest-Booster	XGBoost-Booster	CatBoost-Booster
Accuracy	58.3%	**72.7%**	**93.3%**	**90.9%**	**90.9%**
Sensitivity	57.1%	**83.3%**	**100.0%**	**83.8%**	**80.0%**
Specificity	60.0%	**60.0%**	**88.9%**	**100.0%**	**100.0%**

Table 4.34: The experiment results of boostered machine learning methods on the OPD iTBS dataset.

4.4.3.2 Compare with Previous Work on the Mixed Dataset

Biomarker	Frontal EEG signal				
Treatment	rTMS				
Patients (N)	32 (RCTD dataset)	62 (RCTD dataset + OPD dataset)			
Features	Linear and Nonlinear features on Delta, Theta, Alpha, Beta, Gamma bands				
Classification	Previous data	This work			
	SVM	SVM-Booster	Random Forest-Booster	XGBoost-Booster	CatBoost-Booster
Accuracy	91.7%	**63.2%**	**89.5%**	**89.5%**	**94.7%**
Sensitivity	100.0%	**57.1%**	**85.7%**	**91.7%**	**92.3%**
Specificity	75.0%	**66.7%**	**100.0%**	**85.7%**	**100.0%**

Table 4.35: The experiment results of boostered machine learning methods on the mixed OPD rTMS dataset.

doi:10.6342/NTU202101201

Biomarker	Frontal EEG signal				
Treatment	iTBS				
Patients (N)	30 (RCTD dataset)	67 (RCTD dataset + OPD dataset)			
Features	Linear and Nonlinear features on Delta, Theta, Alpha, Beta, Gamma bands				
Classification	Previous data	This work			
	SVM	SVM-Booster	Random Forest-Booster	XGBoost-Booster	CatBoost-Booster
Accuracy	58.3%	**60.0%**	**85.0%**	**90.0%**	**85.0%**
Sensitivity	57.1%	**100.0%**	**80.0%**	**100.0%**	**80.0%**
Specificity	60.0%	**20.0%**	**90.0%**	**80.0%**	**90.0%**

Table 4.36: The experiment results of boostered machine learning methods on the mixed OPD iTBS dataset.

Table 4.35 and Table 4.36 shows the machine learning methods used booster transformation on the mixed MDD EEG dataset and compare the results with the previous work. From the Table 4.35 and Table 4.36, we can see that the sensitivity performance of the random forest-booster, XGBoost-booster, CatBoost-booster are all better than the original models and also especially on the iTBS OPD dataset. The boostered models' performance are more balance and better than the original models on the iTBS dataset.

4.4.4 Feature Importance and Digital Biomarkers

To make the machine learning model more explainable, the classification and regression tree (CART) based bagging and boosting algorithms have the characteristics that output the feature importance. In this section, I will analysis the feature importance from the output of the bagging and boosting algorithms with p-value < 0.05 statistical significance by the feature extraction step.

4.4.4.1 rTMS Data on Bagging and Boosting Machine Learning Algorithms

Table 4.37 to Table 4.39 are the feature importance list of the bagging and boosting machine learning methods applied on the rTMS datasets. The value of the feature importance in the tables are calculated by the scikit-learn python toolkit, the value is bigger, the feature is more important in that decision tree based machine learning model.

doi:10.6342/NTU202101201

Random Forest Feature Importance on the rTMS dataset			
Feature Importance	**Sub-band**	**Electrode-Feature**	**P-Value**
0.1081	Delta	Fz-Welch	0.0166*
0.1033	Beta	FP2-ApEn	0.0325*
0.0864	Theta	Fz-LLE	0.0166*
0.0847	Beta	F7-ApEn	0.0296*
0.0655	Delta	FP1-ApEn	0.0355*

$Table\ 4.37$: The feature importance in the random forest model on the rTMS dataset.

XGBoost Feature Importance on the rTMS dataset			
Feature Importance	**Sub-band**	**Electrode-Feature**	**P-Value**
0.1710	Theta	FP2-Welch	0.0203*
0.1319	Beta	F4-LLE	0.0325*
0.1112	Beta	F3-LLE	0.0098**
0.0929	Alpha	F3-KFD	0.0270*
0.0765	Delta	F4-Welch	0.0150*

$Table\ 4.38$: The feature importance in the XGBoost model on the rTMS dataset.

CatBoost Feature Importance on the rTMS dataset			
Feature Importance	**Sub-band**	**Electrode-Feature**	**P-Value**
0.5259	Beta	F7-DFA	0.0246*
0.4039	Theta	FP2-Welch	0.0203*
0.0568	Beta	F3-LLE	0.0098**
0.0132	Delta	FP1-ApEn	0.0355*

$Table\ 4.39$: The feature importance in the CatBoost model on the rTMS dataset.

And then, we can analyze the feature importance by electrode channel-wise, feature wise, and the subband-wise to get some detail findings. According to Figure 4.22 to Figure 4.24 we can know that in the bagging and boosting models, the models prefer

doi:10.6342/NTU202101201

to use FP1, FP2, F3, F4, F7 and Fz channels. In the part of linear and nonlinear features, the models prefer to LLE, ApEn and Welch features. In addition, the models prefer to choose the beta, delta, theta bands to make the prediction of response of rTMS.

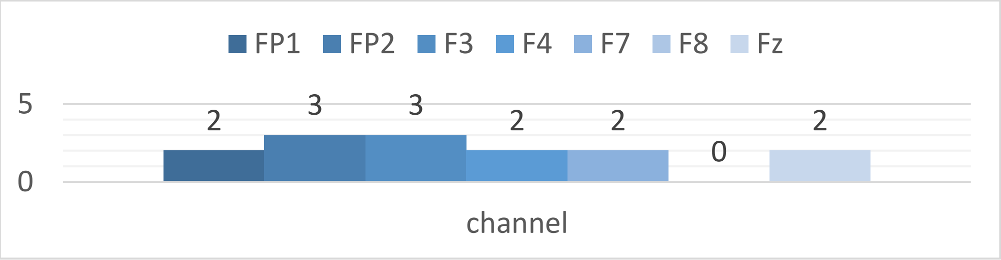


Figure 4.22: The statistics of the channel-wise digital biomarkers of bagging and boosting model on the rTMS dataset.

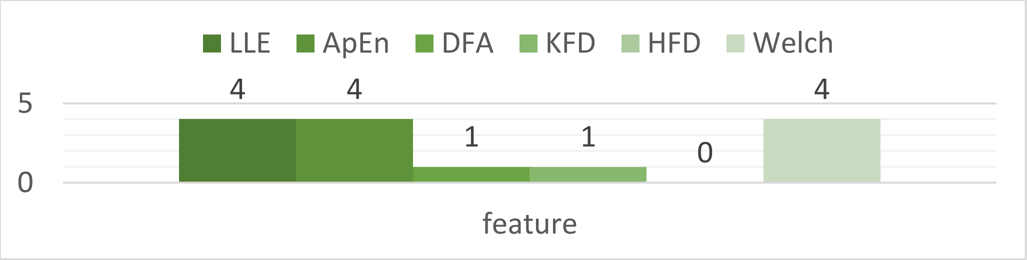


Figure 4.23: The statistics of the feature-wise digital biomarkers of bagging and boosting model on the rTMS dataset.

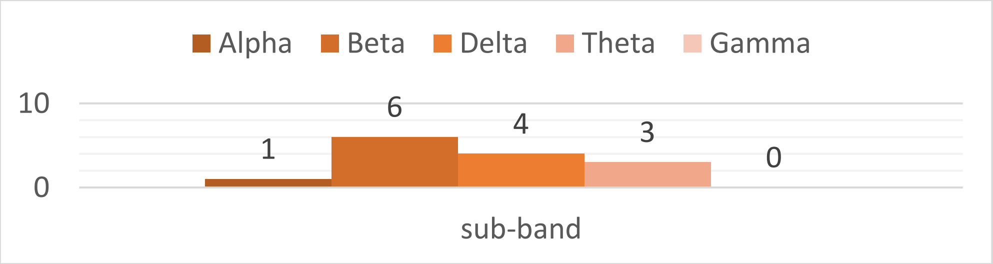


Figure 4.24: The statistics of the subband-wise digital biomarkers of bagging and boosting model on the rTMS dataset.

doi:10.6342/NTU202101201

4.4.4.2 iTBS OPD Data on Bagging and Boosting Machine Learning Algorithms

Similarly, Table 4.40 to Table 4.42 are the feature importance list of the bagging and boosting machine learning methods applied on the iTBS datasets. The value of the feature importance in the tables are calculated by the scikit-learn python toolkit, the value is bigger, the feature is more important in that decision tree based machine learning model. Interestingly, there are four features extracted from the iTBS MDD EEG often selected to use in the prediction by the models: the Welch feature of alpha band, F3 channel, the DFA feature of beta band, F8 channel, the DFA feature extracted from beta band, Fz channel, the KFD feature extracted from theta band, F4 channel. These features maybe can looked as the digital biomarker for iTBS response prediction in the future.

Random Forest Feature Importance on the iTBS dataset

Feature Importance	Sub-band	Electrode-Feature	P-Value
0.1213	Alpha	F3-Welch	0.0247*
0.0949	Beta	F8-DFA	0.0454*
0.0927	Beta	Fz-DFA	0.0015**
0.0745	Theta	F7-KFD	0.0454*
0.0728	Delta	Fz-DFA	0.0425*

Table 4. 40: The feature importance in the random forest model on the iTBS dataset.

doi:10.6342/NTU202101201

XGBoost Feature Importance on the iTBS dataset			
Feature Importance	**Sub-band**	**Electrode-Feature**	**P-Value**
0.1912	Beta	Fz-DFA	0.0015**
0.1138	Theta	F4-KFD	0.0230*
0.0949	Beta	F8-DFA	0.0454*
0.0862	Delta	F7-LLE	0.0425*
0.0809	Theta	F7-ApEn	0.0091**

Table 4. 41: The feature importance in the XGBoost model on the iTBS dataset.

CatBoost Feature Importance on the iTBS dataset			
Feature Importance	**Sub-band**	**Electrode-Feature**	**P-Value**
0.5505	Delta	F3-KFD	0.0454*
0.2578	Alpha	F3-Welch	0.0247*
0.1917	Beta	Fz-DFA	0.0015**

Table 4. 42: The feature importance in the CatBoost model on the iTBS dataset.

And similarly, we can analyze the feature importance by electrode channel-wise, feature wise, and the subband-wise to get some detail findings. According to Figure 4.25 to Figure 4.27 we can know that in the bagging and boosting models, the models prefer to use F3, F7, F8 and Fz channels. In the part of linear and nonlinear features, the models prefer to DFA, KFD and Welch features. In addition, the models prefer to choose the beta, delta, theta bands to make the prediction of response of iTBS.

doi:10.6342/NTU202101201

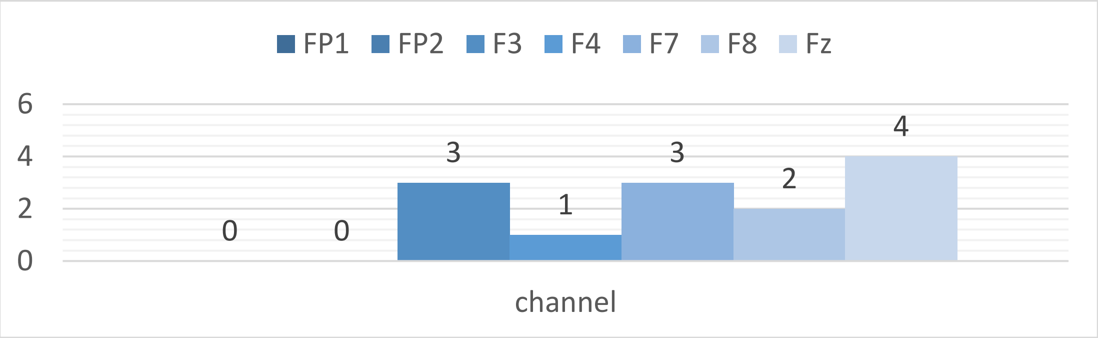


Figure 4.25: The statistics of the channel-wise digital biomarkers of bagging and boosting model on the iTBS dataset.

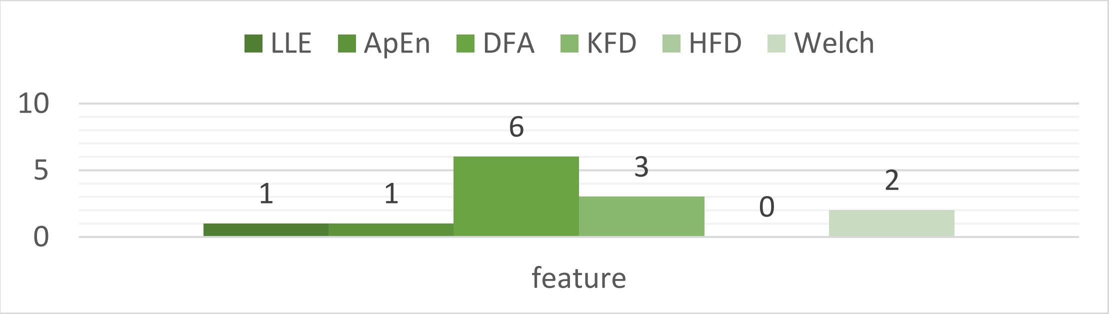


Figure 4.26: The statistics of the feature-wise digital biomarkers of bagging and boosting model on the iTBS dataset.

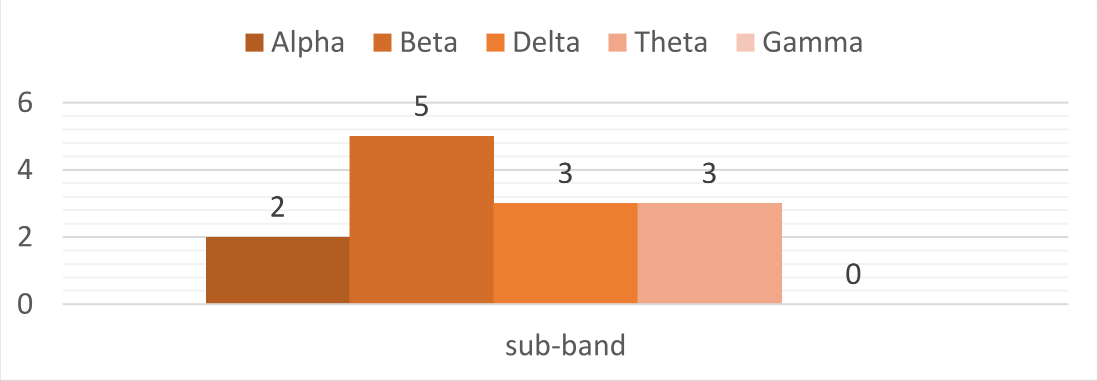


Figure 4.27: The statistics of the subband-wise digital biomarkers of bagging and boosting model on the iTBS dataset.

doi:10.6342/NTU202101201

4.5 Quantum Machine Learning Results

Table 4.43 and Table 4.44 show the quantum machine learning results, the QSVM algorithms were executed on the real backend, IBMQ quantum computer. We can see that under the same feature set, the QSVM are better than classical SVM both on the rTMS and the iTBS dataset, it represents that the quantum feature space can do feature embedding more efficiently than classical computing.

Biomarker	Frontal EEG signal	
Treatment	rTMS	
Patients (N)	62 (RCTD dataset + OPD dataset)	
Features	Linear and Nonlinear features on Delta, Theta, Alpha, Beta, Gamma bands	
Classification	This work	This work
	SVM	QSVM
Accuracy	57.9%	85.7%
Sensitivity	75.0%	87.5%
Specificity	28.6%	83.3%

Table 4.43: The experiment results of QSVM on rTMS dataset.

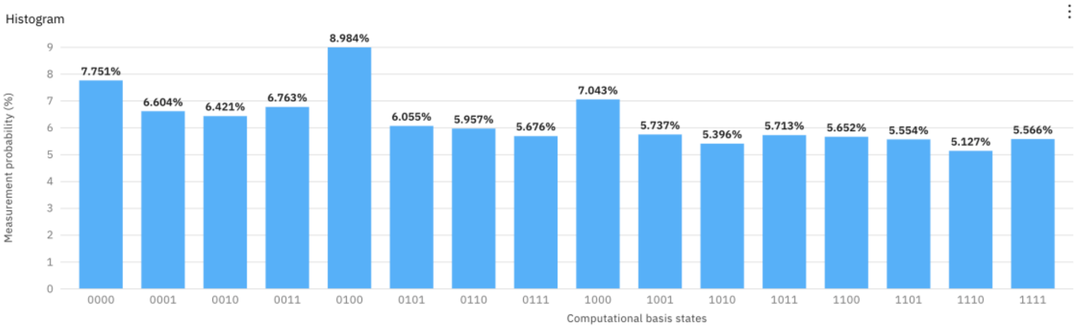
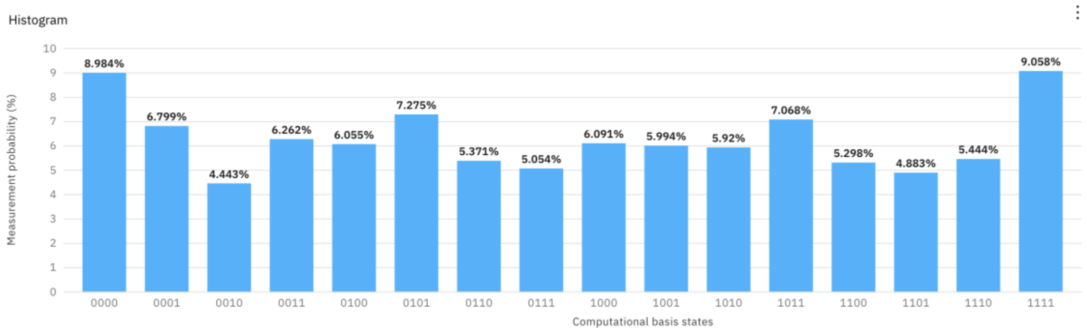


Figure 4.28: The measurement probability histogram of QSVM on the rTMS dataset with 8192 shots.

Biomarker	Frontal EEG signal	
Treatment	iTBS	
Patients (N)	67 (RCTD dataset + OPD dataset)	
Features	Linear and Nonlinear features on Delta, Theta, Alpha, Beta, Gamma bands	
Classification	This work	This work
	SVM	QSVM
Accuracy	40.0%	85.7%
Sensitivity	50.0%	85.7%
Specificity	30.0%	85.7%

$Table\ 4.44$: The experiment results of QSVM on iTBS dataset.



$Figure\ 4.29$: The measurement probability histogram of QSVM on the iTBS dataset with 8192 shots.

Chapter 5 Conclusion

In this thesis, I proposed some new approaches to achieve the goal that enhance the models' performance of response prediction both on rTMS and iTBS under the premise of prevent overfitting. To our knowledge, the present study is the first to demonstrate the combined use of linear and non-linear EEG features along with modern ML, DL, and QML models could enhance the predictive rate of antidepressant responses to not only rTMS, but also iTBS. In the deep learning part, I suggested a new neural network model, ACTSNet, which is inspired by TapNet, can prevent model overfitting by its prototype learning mechanism. The LSTM part in TapNet does not perform well on the long noisy time series data, thus I modified the LSTM part to the attentional convolution architecture and received the state-of-the-art result (rTMS accuracy=83.5% and iTBS accuracy=93.6% both on the OPD dataset). In the classical machine learning part, I tried the bagging and boosting algorithms for their performance are better than SVM according to several publications. The findings in the machine learning part is that the bagging machine learning algorithm outperform than the boosting algorithms on the EEG data (mix dataset random forest accuracy=78.9% on rTMS, accuracy=80.0% on iTBS and OPD dataset random forest accuracy=84.6% on rTMS, accuracy=92.0% on iTBS). In addition, I invented booster

doi:10.6342/NTU202101201

transformation to enhance the sensitivity and shift the performance distribution. Moreover, to train the model more efficiently, I ran the quantum machine learning on the real quantum computer, it can find more feature space by spin and entanglement, and decrease the time complexity than the same structure classical machine learning methods on rTMS and iTBS effect prediction on MDD patients.

doi:10.6342/NTU202101201

Chapter 6 Future Work

To continue this work in the future, firstly, we may can combine more EEG data (more patients data. Strictly speaking, the data samples of this research are still too small for machine learning, which is easy to cause instability and non-repeatability in cross validation results because of the sampling space is quite small.) and with other neural signal time series or image data (e.g., computed tomography (CT), positron emission tomography (PET), magnetic resonance imaging (MRI), PET-CT, PET-MRI, or functional near infra-red spectroscopy (fNIRS), etc.) to get more information and make more solidly prediction multi-models. Secondly, try to know the real mechanism of TMS, we may can try to do some mathematical nonstationary chaos analysis [173] on MDD EEG in signal processing and construct the equations with brain dynamics modeling [177] in neurophysics and simulate by some computational tools like Nengo [172], etc. Thirdly, we can try the semi-supervise learning [157, 158], meta learning [154], few-shot learning [148, 159, 160] or zero-shot learning [155] methods to utilize the unlabeled data in improving the classification performance when training samples are scarce. Moreover, for the lack of the clinical MDD EEG data, generate EEG brain signals by generative adversarial network (GAN) [161] may be the interesting topic in the future. For the machine learning part,

doi:10.6342/NTU202101201

we may can use other EEG features [162, 163] and other feature selection methods like classical emergent computing and evolutionary computation algorithms [164], and their quantum computing versions [165, 166, 167] to improve the model performance. We also can use supercomputing techniques (with bigger RAM of GPU) to train the machine learning and deep learning models on EEG data directly without down-sampling and reducing the information loss [171] or use some model compression tricks like network pruning [174], tensor train [175, 176] methods to reduce the model parameters or quantum computing-native algorithms [43, 44, 103, 104, 106, 132, 144] to achieve the goals of speed-up the computing process or reduce the computation consumption. To actually know the characteristics in mathematics and data science, maybe we can explore the answer from information geometry [183]. Information geometry uses Fisher information matrix to construct $\alpha -$geometry to describe the distance between the data points in flat manifold, is more general than Kullback-Leibler divergence, which is defined only in the Euclidean space. It may can solve the problem of EEG noise reduction and get the more mathematical explanations on the differences between responder and nonresponder of MDD TMS EEG data.

Reference

[1] K. R. Krishnan, “Comorbidity and depression treatment,” *Biological Psychiatry*, vol. 53, no. 8, pp. 701-706, 2003.

[2] B. Kennard et al., “Remission and residual symptoms after short-term treatment in the Treatment of Adolescents with Depression Study (TADS),” *Journal of the American Academy of Child and Adolescent Psychiatry*, vol. 45, no. 12, pp. 1404-1411, 2006.

[3] N. Kennedy et al., “Residual symptoms at remission from depression: impact on long-term outcome,” *Journal of affective disorders*, vol. 80, no. 2-3, pp. 135-144, 2004.

[4] S.P. Roose et al., “Relationship between depression and other medical illnesses,” *JAMA: the journal of the American Medical Association*, vol. 286, no. 14, pp. 1687-1690, 2001.

[5] World Health Organization (WHO), Available: https://www.who.int/mental_health/management/depression/en/, 2020.

[6] World Health Organization (WHO), Available: https://www.who.int/news-room/fact-sheets/detail/depression , 2017.

[7] M. S. Reddy, “Depression: the disorder and the burden,” *Indian J Psychol Med*, vol. 32, no. 1, pp. 1-2, 2010.

[8] National Institute of Mental Health (NIH), Available: https://www.nimh.nih.gov/health/statistics/major-depression.shtml.

[9] Warden D et al., “The STAR*D Project results: a comprehensive review of findings,” *Curr Psychiatry Rep*, vol. 9, no. 6, pp. 449-459, 2007.

doi:10.6342/NTU202101201

[10] A.J. Rush et al., “Acute and longer-term outcomes in depressed outpatients requiring one or several treatment steps: a STAR*D report,” *Am J Psychiatry*, vol. 163, no. 11, pp. 1905-1917, 2006.

[11] C.T. Li, *Class Lecture*, Topic: “Treatment-Resistant Depression (TRD) and Antidepressant Mechanisms of Theta-Burst Stimulation (TBS),” Department of Psychiatry, Taipei Veterans General Hospital, 2021.

[12] C.T. Li et al., “Critical role of glutamatergic and GABAergic neurotransmission in the central mechanisms of theta-burst stimulation,” *Human Brain Mapping*, vol. 40, no. 6, pp. 2001-2009, 2019.

[13] C.T. Li et al, “Antidepressant mechanism of add-on repetitive transcranial magnetic stimulation in medication-resistant depression using cerebral glucose metabolism,” *Journal of Affective Disorders*, vol. 127, no. 1-3, pp. 219-229, 2010.

[14] C.T. Li et al., “Efficacy of prefrontal theta-burst stimulation in refractory depression: a randomized sham-controlled study,” *Brain*, vol. 137, no. 7, pp. 2088-2098, 2014.

[15] C.T. Li et al., “Antidepressant Efficacy of Prolonged Intermittent Theta Burst Stimulation Monotherapy for Recurrent Depression and Comparison of Methods for Coil Positioning: A Randomized, Double-Blind, Sham-Controlled Study,” *Biological Psychiatry*, vol. 87, no. 5, pp. 443-450, 2020.

[16] Qingqing Liu et al., “Changes in the global burden of depression from 1990 to 2017: Findings from the Global Burden of Disease study,” *Journal of Psychiatric Research*, vol. 126, pp. 134-140, 2020.

[17] A. L. Brody et al., “Prefrontal-subcortical and limbic circuit mediation of major depressive disorder,” *Seminars in Clinical Neuropsychiatry*, vol. 6, no. 2, pp. 102-112, 2001.

[18] W. C. Drevets, “Neuroimaging studies of mood disorders,” *Biological Psychiatry*, vol. 48, no. 8, pp. 813-829, 2000.

[19] R. J. Davidson et al., “Depression: perspectives from affective neuroscience,” *Annual Review of Psychology*, vol. 53, pp. 545-574, 2002.

[20] J. F. Thayer et al., “A model of neurovisceral integration in emotion regulation and dysregulation,” *Journal of Affective Disorders*, vol. 61, no. 3, pp. 201-216, 2000.

[21] O. Devinsky et al., “Contributions of anterior cingulate cortex to behavior,” *Brain*, vol. 118, no. 1, pp. 279-306, 1995.

[22] D. A. Pizzagalli et al., “Anterior Cingulate Activity as a Predictor of Degree of Treatment Response in Major Depression: Evidence From Brain Electrical Tomography Analysis,” *The American Journal of Psychiatry*, vol. 158, no. 3, pp. 405-415, 2001.

[23] H. Asada et al., “Frontal midline theta rhythms reflect alterative activation of prefrontal cortex and anterior cingulate cortex in humans,” *Neuroscience Letters*, vol. 274, no. 1, pp. 29-32, 1999.

[24] P. S. Cooper et al., “Theta frontoparietal connectivity associated with proactive and reactive cognitive control processes,” *Neuroimage*, vol. 108, pp. 354-363, 2015.

[25] J. F. Cavanagh et al., “Frontal theta as a mechanism for cognitive control,” *Trends Cogn Sci*, vol. 18, no. 8, pp. 414-421, 2014.

[26] Y. Noda et al., “Neurobiological mechanisms of repetitive transcranial magnetic stimulation of the dorsolateral prefrontal cortex in depression: a systematic review,” *Psychol Med*, vol. 45, no. 16, pp. 3411-3432, 2015.

[27] M. Tik et al., “Towards understanding rTMS mechanism of action: Stimulation of the DLPFC causes network-specific increase in functional connectivity,” *Neuroimage*, vol. 162, pp. 289-296, 2017.

[28] Kevin C. Bickart et al., “The amygdala as a hub in brain networks that support social life,” *Neuropsychologia,* vol. 63, pp. 235-248, 2014.

[29] C.T. Hsu, "探討經由認知作業程式驅動後的前額葉 Theta 波預測憂鬱症病患未來之療效以及認知作業程式開發與驗證," *Master Thesis,* National Taiwan University, 2017.

[30] Ina Wu, “Real Time Computer Aided Detection System for the Prediction of Clinical Antidepressant Responses,” *Master Thesis,* National Taiwan University, 2017.

[31] Yi-Chen Li, “Real Time EEG Analysis for Prediction of Antidepressant Responses of Transcranial Magnetic Stimulation in Major Depressive Disorder Based on Machine Learning,” *Master Thesis,* National Taiwan University, 2019.

[32] C.T. Li et al., “Cognition-Modulated Frontal Activity in Prediction and Augmentation of Antidepressant Efficacy: A Randomized Controlled Pilot Study,” *Cerebral Cortex*, vol. 26, no. 1, pp. 202-210, 2014.

[33] Fatemeh Hasanzadeh et al., “Prediction of rTMS treatment response in major depressive disorder using machine learning techniques and nonlinear features of EEG signal,” *Journal of Affective Disorders*, vol. 256, pp. 132-142, 2019.

[34] Amin Zandvakili et al., “Use of machine learning in predicting clinical response to transcranial magnetic stimulation in comorbid posttraumatic stress disorder and major depression: A resting state electroencephalography study,” *Journal of Affective Disorders*, vol. 252, pp. 47-54, 2019.

[35] Turker Tekin Erguzel et al., “Classification of major depressive disorder subjects using Pre-rTMS electroencephalography data with support vector machine approach,” *2014 IEEE Science and Information Conference*, pp. 410-414, 2014.

[36] Nathan Bakker, “Resting-State Functional Connectivity Predicts

Individual Treatment Outcomes of Repetitive Transcranial Magnetic Stimulation for Major Depressive Disorder,” *Master Thesis*, University of Toronto, 2014.

[37] Jue Wang et al., “High-Frequency rTMS of the Motor Cortex Modulates Cerebellar and Widespread Activity as Revealed by SVM,” *Frontiers in Neuroscience*, vol. 14, pp. 186, 2020.

[38] Turker Tekin Erguzel et al., “Machine Learning Approaches to Predict Repetitive Transcranial Magnetic Stimulation Treatment Response in Major Depressive Disorder,” *Proceedings of SAI Intelligent Systems Conference (IntelliSys) 2016*, pp. 391-401, 2016.

[39] Turker Tekin Erguzel et al., “Feature Selection and Classification of Electroencephalographic Signals: An Artificial Neural Network and Genetic Algorithm Based Approach,” *Clinical EEG and Neuroscience*, vol. 46, no. 4, pp. 321-326, 2015.

[40] Wei Wu et al, “An electroencephalographic signature predicts antidepressant response in major depression,” *Nature Biotechnology*, vol. 38, pp. 439-447, 2020.

[41] Zhijiang Wan et al., “HybridEEGNet: A Convolutional Neural Network for EEG Feature Learning and Depression Discrimination,” *IEEE Access*, vol. 8, pp. 30332-30342, 2020.

[42] U. RajendraAcharya et al., “Automated EEG-based screening of depression using deep convolutional neural network,” *Computer Methods and Programs in Biomedicine*, vol. 161, pp. 103-113, 2018.

[43] Javier Alcazar et al., “Classical versus quantum models in machine learning: insights from a finance application,” *Machine Learning: Science and Technology*, vol. 1, no. 3, 035003, 2020.

[44] YaoChong Li et al., “A quantum mechanics-based framework for EEG signal feature extraction and classification,” *IEEE Transactions on Emerging Topics in Computing*, vol. 14, no. 8, pp. 1-1, 2020.

[45] Bárbara M. de Andrade et al., “Comparison of the performance of multiclass classifiers in chemical data: Addressing the problem of overfitting with the permutation test,” *Chemometrics and Intelligent Laboratory Systems*, vol. 201, 104013, 2020.

[46] RaviGarg et al., “Automating Ischemic Stroke Subtype Classification Using Machine Learning and Natural Language Processing,” *Journal of Stroke and Cerebrovascular Diseases*, vol. 28, no. 7, pp. 2045-2051, 2019.

[47] Natalia Jaworska et al., “Leveraging Machine Learning Approaches for Predicting Antidepressant Treatment Response Using Electroencephalography (EEG) and Clinical Data,” *Frontiers in Psychiatry*, vol. 9, pp. 768, 2019.

[48] Milena Cukic et al., “EEG machine learning with Higuchi fractal dimension and Sample Entropy as features for successful detection of depression,” *arXiv preprint arXiv:1803.05985v1*, 2018.

[49] Thigo M. Nunes et al., “EEG signal classification for epilepsy diagnosis via optimum path forest – A systematic assessment,” *Neurocomputing*, vol. 136, pp. 103-123, 2014.

[50] G. Ratsch et al., “Constructing boosting algorithms from SVMs: an application to one-class classification,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 9, pp. 1184-1199, 2002.

[51] Dixita Mali et al., “A Machine Learning Technique to Analyze Depressive Disorders,” *Research Square preprint Research Square: rs.3.rs-322564/v1*, 2021.

doi:10.6342/NTU202101201

[52] Abraham J. Wyner et al., “Explaining the Success of AdaBoost and Random Forests as Interpolating Classifiers,” *Journal of Machine Learning Research*, vol. 18, no. 18, pp. 1-33, 2017.

[53] Qiao Yuanhua et al., “Machine Learning Approaches for MDD Detection and Emotion Decoding Using EEG Signals,” *Frontiers in Human Neuroscience*, vol. 14, pp. 284, 2020.

[54] Turker Tekin Erguze et al., “Neural Network Based Response Prediction of rTMS in Major Depressive Disorder Using QEEG Cordance,” *Psychiatry Investig,* vol. 12, no. 1, pp. 61-65, 2015.

[55] Amin Zandvakili et al., “Changes in functional connectivity after theta-burst transcranial magnetic stimulation for post-traumatic stress disorder: a machine-learning study,” *European Archives of Psychiatry and Clinical Neuroscience,* vol. 271, pp. 29-37, 2021.

[56] Behshad Hosseinifard et al., “Classifying depression patients and normal subjects using machine learning techniques and nonlinear features from EEG signal,” *Comput Methods Programs Biomed*, vol. 109, pp. 339-345, 2013.

[57] Wajid Mumtaz et al., “A machine learning framework involving EEG-based functional connectivity to diagnose major depressive disorder (MDD),” *Medical & Biological Engineering & Computing*, vol. 56, no. 2, pp. 233-246, 2018.

[58] Xiao-WeiWang et al., “Emotional state classification from EEG data using machine learning approach,” *Neurocomputing*, vol. 129, pp. 94-106, 2014.

[59] Dan V. Iosifescu, “Are Electroencephalogram-Derived Predictors of Antidepressant Efficacy Closer to Clinical Usefulness?,” *Invited Commentary, JAMA Psychiatry*, 2020.

doi:10.6342/NTU202101201

[60] Y. P. Lin et al., “Fusion of electroencephalographic dynamics and musical contents for estimating emotional responses in music listening,” *Frontiers in Neuroscience*, vol. 8, pp. 94, 2014.

[61] Diego Alvarez-Estevez, “European Data Format,” Available: European Data Format (EDF) (edfplus.info)

[62] Pierre Comon et al., “Independent Component Analysis,” *Higher-Order Statistics*, pp. 29-38, 1992.

[63] Jutten et al., “Independent component analysis versus principal component analysis,” *Signal Processing IV: Theo. and Appl.*, 1988.

[64] Peizhen Peng et al., “Epileptic Seizure Prediction in Scalp EEG Using an Improved HIVE-COTE Model,” *39th IEEE Chinese Control Conference (CCC)*, pp. 6450-6457, 2020.

[65] Chenglong Dai et al., “CenEEGs: Valid EEG Selection for Classification,” *ACM Transactions on Knowledge Discovery from Data*, vol. 14, no. 2, pp. 18:1-18:25, 2020.

[66] Alejandro Pasos Ruiz et al., “Benchmarking Multivariate Time Series Classification Algorithms,” *arXiv preprint arXiv:2007.13156*, 2020.

[67] Chenglong Dai et al., “Shapelet-transformed Multi-channel EEG Channel Selection,” *ACM Transactions on Intelligent Systems and Technology*, vol. 11, no. 5, pp. 58:1-58:27, 2020.

[68] Yann LeCun et al., “Convolutional Networks for Images, Speech, and Time-Series,” in *The handbook of brain theory and neural networks*, MIT Press, 1995.

[69] J. Long et al., “Fully Convolutional Networks for Semantic Segmentation,” *Conference on Computer Vision and Pattern Recognition 2015 (CVPR 2015)*, pp. 3431-3440, 2015.

[70] Jian Bo Yang et al., “Deep Convolutional Neural Networks On Multichannel Time Series For Human Activity Recognition,” *Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015)*, pp. 3995-4001, 2015.

[71] Kaiming He et al., “Deep Residual Learning for Image Recognition,” *arXiv preprint arXiv:1512.03385*, 2015.

[72] Le Guennec et al., “Data Augmentation for Time Series Classification using Convolutional Neural Networks,” *ECML/PKDD Workshop on Advanced Analytics and Learning on Temporal Data*, 2016.

[73] Hassan Ismail Fawaz et al., “InceptionTime: Finding AlexNet for Time Series Classification,” *arXiv preprint arXiv:1909.04939*, 2019.

[74] Christian Szegedy et al., “Going deeper with convolutions,” *arXiv preprint arXiv:1409.4842v1*, 2014.

[75] Hinton et al., “Learning representations by back-propagating errors,” *Nature*, vol. 323, pp. 533-536, 1986.

[76] Sepp Hochreiter et al., “LONG SHORT-TERM MEMORY,” *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, 1997.

[77] Xuchao Zhang et al., “TapNet: Multivariate Time Series Classification with Attentional Prototypical Network,” *Association for the Advancement of Artificial Intelligence Conference 2020 (AAAI 2020)*, vol. 34, no. 04, pp. 6845-6852, 2020.

[78] Joan Serrà et al., “Towards a Universal Neural Network Encoder for Time Series,” *arXiv preprint arXiv:1805.03908*, 2018.

[79] Fazle Karim et al., “LSTM Fully Convolutional Networks for Time Series Classification,” *IEEE Access*, vol. 6, pp. 1662-1669, 2017.

[80] Fazle Karim et al., “Multivariate LSTM-FCNs for Time Series Classification,” *Neural Networks*, vol. 116, pp. 237-245, 2019.

doi:10.6342/NTU202101201

[81] Dmitry Ulyanov et al., “Instance Normalization: The Missing Ingredient for Fast Stylization,” *arXiv preprint arXiv:1607.08022*, 2016.

[82] Bing Xu et al., “Empirical Evaluation of Rectified Activations in Convolution Network,” *arXiv preprint arXiv:1505.00853v2*, 2015.

[83] He et al., “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” *arXiv preprint arXiv:1502.01852*, 2015.

[84] Patrick Schäfer, “Bag-Of-SFA-Symbols in Vector Space (BOSS VS),” *ZIB-Report*, vol. 30, 2015.

[85] Patrick Schäfer, “Scalable time series classification,” *Data Mining and Knowledge Discovery*, vol. 30, no. 5, pp. 1273-1298, 2015.

[86] Patrick Schäfer et al., “The BOSS is concerned with time series classification in the presence of noise,” *Data Mining and Knowledge Discovery*, vol. 29, pp. 1505-1530, 2015.

[87] Patrick Schäfer et al., “SFA: A Symbolic Fourier Approximation and Index for Similarity Search in High Dimensional Datasets,” *EDBT '12: Proceedings of the 15th International Conference on Extending Database Technology*, pp. 516-527, 2012.

[88] Zellig Harris, “Distributional Structure,” *WORD*, vol. 10, pp. 146-162, 1954.

[89] Xueqi Zhang et al., “Time-Series Prediction of Environmental Noise for Urban IoT Based on Long Short-Term Memory Recurrent Neural Network,” *Applied Sciences*, vol. 10, no. 3, pp. 1144, 2020.

[90] C. Lee Giles et al., “Noisy Time Series Prediction using Recurrent Neural Networks and Grammatical Inference,” *Machine Learning*, vol. 44, pp. 161-183, 2001.

[91] A. Wolf et al., “Determining Lyapunov Exponents From a Time Series,” *Physica D: Nonlinear Phenomena*, vol. 16, no. 3, pp. 285-317, 1985.

doi:10.6342/NTU202101201

[92] L. F. Márton et al., “Detrended Fluctuation Analysis of EEG Signals,” *Procedia Technology*, vol. 12, pp. 125-132, 2014.

[93] S. M Pincus et al., “A regularity statistic for medical data analysis,” *Journal of Clinical Monitoring and Computing*, vol. 7, pp. 335-345, 1991.

[94] M. J. Katz, “Fractals and the analysis of waveforms,” *Computers in Biology and Medicine*, vol. 18, no. 3, pp. 145-156, 1988.

[95] T. Higuchi, “Approach to an irregular time series on the basis of the fractal theory,” *Physica D: Nonlinear Phenomena*, vol. 31, no. 2, pp. 277-283, 1988.

[96] P. D. Welch, “The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms,” *IEEE Trans. Audio Electroacoust*, vol. 15, no. 2, pp. 70-73, 1967.

[97] L. Breiman, “Random Forests,” *Machine Learning*, vol. 45, pp. 5-32, 2001.

[98] Freund et al., “A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting,” *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119-139, 1997.

[99] Tianqi Chen et al., “XGBoost: A Scalable Tree Boosting System,” *KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785-794, 2016.

[100] Cortes et al., “Support-Vector Networks,” *Machine Learning*, vol. 20, pp. 273-297, 1995.

[101] Anna Veronika Dorogush et al., “CatBoost: gradient boosting with categorical features support,” *arXiv preprint arXiv:1810.11363*, 2018.

[102] Zhang, K. et al., “Domain adaptation under target and conditional shift,” *Proceedings of Machine Learning Research (PMLR)*, vol. 28, no. 3, pp. 819-827, 2013.

[103] Patrick Rebentrost et al., "Quantum Support Vector Machine for Big Data Classification," *Physical Review Letters*, vol. 113, no. 130503, 2014.

[104] Peter Wittek, *Quantum Machine Learning: What Quantum Computing Means to Data Mining*, Academic Press, 2014.

[105] Christopher Havenstein et al., "Comparisons of Performance between Quantum and Classical Machine Learning," *SMU Data Science Review*, vol. 1, no. 4, pp. 11, 2018.

[106] Vojtech Havlicek et al. "Supervised learning with quantum enhanced feature spaces," *Nature*, vol. 567, pp. 209-212, 2019.

[107] Aram W. Harrow et al., "Quantum algorithm for solving linear systems of equations," *arXiv preprint arXiv:0811.3171*, 2008.

[108] P V Zahorodko et al., "Comparisons of performance between quantum-enhanced and classical machine learning algorithms on the IBM Quantum Experience," *Journal of Physics: Conference Series*, no. 1840, 2021.

[109] Andrew Cross, "The IBM Q experience and QISKit open-source quantum computing software," *American Physical Society March Meeting (APS)*, no. L58.003, 2018.

[110] James Large et al., "sktime-dl," *Github*; https://github.com/sktime/sktime-dl (Accessed 2020 November-2021 March).

[111] Fazle Karim et al., "MLSTM-FCN," *Github*; https://github.com/titu1994/MLSTM-FCN (Accessed 2020 November-2021 March).

[112] Zhang et al., "tapnet," *Github*; https://github.com/xuczhang/tapnet (Accessed 2021 February-2021 March).

doi:10.6342/NTU202101201

[113] Ingo Mierswa, “Controlling overfitting with multi-objective support vector machines,” *GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pp. 1830-1837, 2007.

[114] Henry Hen et al., “Overcome Support Vector Machine Diagnosis Overfitting,” *Cancer Informatics*, vol. 13, no. s1, 2014.

[115] C.T. Li et al., “Effects of prefrontal theta-burst stimulation on brain function in treatment-resistant depression: A randomized sham-controlled neuroimaging study,” *Brain Stimulation*, vol. 11, no. 5, pp. 1054-1062, 2018.

[116] Davide Anguita et al., “Model Selection for Support Vector Machines: Advantages and Disadvantages of the Machine Learning Theory,” *International Joint Conference on Neural Networks (IJCNN)*, 2010.

[117] R. Zhang et al., “An improved SVM method P-SVM for classification of remotely sensed data,” *International Journal of Remote Sensing*, vol. 29, no. 20, pp. 6029-6036, 2008.

[118] I. V. Tetko et al., “Neural network studies. 1. Comparison of overfitting and overtraining,” *Journal of Chemical Information and Modeling*, vol. 35, no. 5, pp. 826-833, 1995.

[119] Vladimir Vapnik, *The nature of statistical learning theory*, Springer, 2000.

[120] Vladimir Vapnik et al., “On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities,” *Theory of Probability & Its Applications*, vol. 16, no. 2, pp.264, 1971.

[121] Mikhail Belkin et al., “Overfitting or perfect fitting? Risk bounds for classification and regression rules that interpolate,” *Advances in Neural Information Processing Systems 31 (NeurIPS 2018)*, 2018.

doi:10.6342/NTU202101201

[122] Arthur Jacot et al., “Neural Tangent Kernel: Convergence and Generalization in Neural Networks,” *Advances in Neural Information Processing Systems 31 (NeurIPS 2018),* 2018.

[123] Sanjeev Arora et al., “Fine-Grained Analysis of Optimization and Generalization for Overparameterized Two-Layer Neural Networks,” *arXiv preprint arXiv:1901.08584*, 2019.

[124] Kenji Kawaguchi et al., “Generalization in Deep Learning,” *arXiv preprint arXiv:1710.05468*, 2020.

[125] Fengxiang He et al., “Recent advance in deep learning theory,” *arXiv preprint arXiv:2012.10931*, 2021.

[126] Photograph of Qubit and Bit, *depositphotos*, Accessed on: May, 5, 2021. [Online]. Available: https://cn.depositphotos.com/419582472/stock-illustration-qubit-bit-states-classical-bit.html

[127] Sebastian Ruder, “An overview of gradient descent optimization Algorithms,” *arXiv preprint arXiv:1609.04747v2*, 2016.

[128] Haoyan Xu et al., “Multivariate Time Series Classification with Hierarchical Variational Graph Pooling,” *arXiv preprint arXiv:2010.05649*, 2020.

[129] LinWang et al., “An effective multivariate time series classification approach using echo state network and adaptive differential evolution algorithm,” *Expert Systems with Applications*, vol. 43, pp. 237-249, 2016.

[130] Lines et al., “Time Series Classification with HIVE-COTE: The Hierarchical Vote Collective of Transformation-based Ensembles,” *ACM Transactions on Knowledge Discovery from Data*, vol. 12, no. 5, 2018.

[131] R. P. Feynman, “Simulating physics with computers,” *International Journal of Theoretical Physics*, vol. 21, no. 6-7, pp. 467-488, 1982.

doi:10.6342/NTU202101201

[132] J. Biamonte et al., “Quantum machine learning,” *Nature*, vol. 549, no. 7671, pp. 195–202, 2017.

[133] V. Dunjko et al., “Quantum-Enhanced Machine Learning,” *Physical Review Letters*, vol. 117, no. 13, pp. 130501, 2016.

[134] I. Cong et al., “Quantum discriminant analysis for dimensionality reduction and classification,” *New Journal of Physics*, vol. 18, no. 7, pp. 073011, 2016.

[135] M. Schuld et al., “Implementing a distance-based classifier with a quantum interference circuit,” *EPL (Europhysics Letters)*, vol. 119, no. 6, pp. 60002, 2017.

[136] M. Schuld et al., “Quantum ensembles of quantum classifiers,” *Scientific reports*, vol. 8, no. 1, pp. 2772, 2018.

[137] Y. Dang et al., “Image classification based on quantum K-Nearest-Neighbor algorithm,” *Quantum Information Processing*, vol. 17, no. 9, pp. 239, 2018.

[138] J. Zhao et al., “Building quantum neural networks based on a swap test,” *Physical Review A*, vol. 100, no. 1, pp. 012334, 2019.

[139] N. Wiebe et al., “Quantum algorithm for data fitting,” *Physical Review Letters*, vol. 109, no. 5, pp. 050505, 2012.

[140] M. Schuld et al., “Prediction by linear regression on a quantum computer,” *Physical Review A*, vol. 94, no. 2, pp. 022342, 2016.

[141] G. Wang et al., “Quantum algorithm for linear regression,” *Physical Review A*, vol. 96, no. 1, pp. 012335, 2017.

[142] C.-H. Yu et al., “An improved quantum algorithm for ridge regression,” *IEEE Transactions on Knowledge and Data Engineering*, 2019.

[143] D. Horn et al., “Algorithm for data clustering in pattern recognition problems based on quantum mechanics,” *Physical Review Letters*, vol. 88, no. 1, pp. 187 021–187 024, 2002.

doi:10.6342/NTU202101201

[144] E. Aımeur et al., “Quantum speed-up for unsupervised learning,” *Machine Learning*, vol. 90, no. 2, pp. 261-287, 2013.

[145] J. Romero et al., “Quantum autoencoders for efficient compression of quantum data,” *Quantum Science and Technology*, vol. 2, no. 4, pp. 045001, 2017.

[146] C.-H. Yu et al., “Quantum data compression by principal component analysis,” *Quantum Information Processing*, vol. 18, no. 8, p. 249, 2019.

[147] B. Neyshabur et al., “In search of the real inductive bias: On the role of implicit regularization in deep learning,” *arXiv preprint arXiv:1412.6614*, 2014.

[148] J. Snell et al., “Prototypical networks for few-shot learning,” *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, pp. 4077-4087, 2017.

[149] Ulf Grenander, “The Nyquist frequency is that frequency whose period is two sampling intervals,” *Probability and Statistics: The Harald Cramér Volume*, 1959.

[150] H. M. Yang et al., “Robust Classification with Convolutional Prototype Learning,” *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, no. 18347865, 2018.

[151] M. Ilse et al., “Attention-based deep multiple instance learning,” *arXiv preprint arXiv:1802.04712*, 2018.

[152] Arindam Banerjee et al, “Clustering with Bregman Divergences,” *Journal of Machine Learning Research*, vol. 6, pp. 1705-1749, 2005.

[153] D. P. Kingma et al., “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[154] Joaquin Vanschoren, “Meta-Learning: A Survey,” *arXiv preprint arXiv:1810.03548*, 2018.

[155] Richard Socher et al., “Zero-Shot Learning Through Cross-Modal Transfer,” *arXiv preprint arXiv:1301.3666*, 2013.

[156] Abhinav Kandala et al., “Hardware-efficient Variational Quantum Eigensolver for Small Molecules and Quantum Magnets,” *arXiv preprint arXiv:1704.05018*, 2017.

[157] O. Chapelle et al., “Semi-Supervised Learning,” *IEEE Transactions on Neural Networks*, vol. 20, no. 3, pp. 542, 2009.

[158] Jesper E. van Engelen et al., “A survey on semi-supervised learning,” *Machine Learning*, vol. 109, pp. 373-440, 2020.

[159] Flood Sung et al. “Learning to Compare: Relation Network for Few-Shot Learning,” *the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1199-1208, 2018.

[160] Sachin Ravi et al., “Optimization as a Model for Few-Shot Learning,” *International Conference on Learning Representations (ICLR)*, 2017.

[161] Kay Gregor Hartmann et al., “EEG-GAN: Generative adversarial networks for electroencephalograhic (EEG) brain signals,” *arXiv preprint arXiv:1806.01875*, 2018.

[162] Shiliang Sun et al., “A review of adaptive feature extraction and classification methods for EEG-based brain-computer interfaces,” *2014 International Joint Conference on Neural Networks (IJCNN)*, no. 14563793, 2014.

[163] Wan Amirah W Azlan et al., “Feature extraction of electroencephalogram (EEG) signal - A review,” *2014 IEEE Conference on Biomedical Engineering and Sciences (IECBES)*, no. 14950668, 2014.

[164] 李士勇等。*智能優化演算法與湧現計算*。清華大學出版社，2019。

[165] Donald A. Sofge et al., “Toward a Framework for Quantum Evolutionary Computation,” *2006 IEEE Conference on Cybernetics and Intelligent Systems*, no. 9231748, 2006.

[166] Gexiang Zhang, “Quantum-inspired evolutionary algorithms: a survey and empirical study,” *Journal of Heuristics*, vol. 17, pp. 303-351, 2010.

[167] R.K.Agrawal et al., “Quantum based Whale Optimization Algorithm for wrapper feature selection,” *Applied Soft Computing*, vol. 89, no. 106092, 2020.

[168] X. D. Cai et al., “ “Entanglement-based machine learning on a quantum computer,” *Physical Review Letters*, vol.114, no. 11, p. 110504, 2015.

[169] Z. Li et al., “Experimental realization of a quantum support vector machine,” *Physical Review Letters*, vol. 114, no. 14, p. 140504, 2015.

[170] F. Tacchino et al., “An artificial neuron implemented on an actual quantum processor,” *npj Quantum Information*, vol. 5, no. 1, p. 26, 2019.

[171] Albert Reuther et al., “Interactive Supercomputing on 40,000 Cores for Machine Learning and Data Analysis,” *2018 IEEE High Performance extreme Computing Conference (HPEC)*, no. 18290412, 2018.

[172] Trevor Bekolay et al., “Nengo: a Python tool for building large-scale functional brain models,” *Frontiers in Neuroimformatics*, vol. 7, pp. 48, 2014.

[173] W.S. Pritchard et al., “Measuring Chaos in the Brain - A Tutorial Review of EEG Dimension Estimation,” *Brain and Cognition*, vol. 27, no. 3, pp. 353-397, 1995.

[174] Yu Cheng et al., “A Survey of Model Compression and Acceleration for Deep Neural Networks,” *arXiv preprint arXiv:1710.09282*, 2017.

[175] Alexander Novikov et al., “Tensorizing Neural Networks,” *Advances in Neural Information Processing Systems 28 (NIPS 2015)*, 2015.

[176] Yinchong Yang et al., “Tensor-Train Recurrent Neural Networks for Video Classification,” *arXiv preprint arXiv:1707.01786*, 2017.

[177] P. A. Robinson et al., “Neurophysical Modeling of Brain Dynamics,” *Nature Neuropsychopharmacology*, vol. 28, pp. S74-S79, 2003.

[178] Felix Bloch, “Nuclear induction,” *Phys. Rev.*, vol. 70, no. 7-8, pp. 460–474, 1946.

[179] Nicholas I. Sapankevych et al., “Time Series Prediction Using Support Vector Machines: A Survey,” *IEEE Computational Intelligence Magazine*, vol. 4, no. 2, pp. 24-38, 2009.

[180] Juliana Tolles et al., “Logistic Regression Relating Patient Characteristics to Outcomes,” *JAMA Guide to Statistics and Methods*, vol. 316, no. 5, pp. 533-534, 2016.

[181] Fletcher et al., *Practical Methods of Optimization (2nd ed.)*, New York: John Wiley & Sons, 1987.

[182] Yaser S. et al., *Learning From Data: A Short Course (Hardcover),* AMLBook, 2012.

[183] S. Amari, *Information Geometry and Its Applications,* Springer, 2016.

[184] A. M. Chekroud et al., “Cross-trial prediction of treatment outcome in depression: a machine learning approach,” *Lancet Psychiatry*, vol. 3, no. 3, pp. 243-250, 2016.

[185] Bulat Ibragimov et al., “Minimal Variance Sampling in Stochastic Gradient Boosting,” *arXiv preprint arXiv:1910.13204,* 2019.

[186] Mohammad Teshnehlab et al., “Feature Extraction and Classification of EEG Signals Using Wavelet Transform, SVM and Artificial Neural Networks for Brain Computer Interfaces,” *International Joint Conferences on Bioinformatics, Systems Biology and Intelligent Computing*, pp. 352-355, 2009.

doi:10.6342/NTU202101201