Data-Efficient Deep Representation Learning for Brain-Computer Interface and Its Applications

Xiang Zhang

A thesis in fulfillment of the requirements for the degree of

Doctor of Philosophy

School of Computer Science and Engineering

Faculty of Engineering

The University of New South Wales

January 2020

Thesis/Dissertation Sheet

Surname/Family Name	:	**Zhang**
Given Name/s	:	**Xiang**
Abbreviation for degree as give in the University calendar	:	**PhD**
Faculty	:	**Engineering**
School	:	**Computer Science and Engineering**
Thesis Title	:	**Data-Efficient Deep Representation Learning for Brain-Computer Interface and Its Applications**

Abstract 350 words maximum: (PLEASE TYPE)

Brain-Computer Interface (BCI) bridges the human being's neural world and the outer physical world by decoding individuals' brain signals into commands recognizable by computer devices, which has attracted increasing attention in recent years. This dissertation aims at overcoming the hurdles and stretching the horizons of data-efficient BCI systems by developing robust deep representation learning paradigms. First, we present a comprehensive introduction of BCI systems including the deep learning models, state-of-the-art studies adopting deep learning for BCI drawbacks, and the appealing real-world BCI applications. Moreover, we propose automatic high-level representation learning methods through deep architectures addressing the traditional time-consuming manually feature engineering and low signal-to-noise ratio data. In addition, we develop reinforced selective attention mechanism by combining reinforcement learning and deep neural network for capturing informative representations adaptive to different scenarios. Furthermore, we design a weakly-supervised predictive model to harness the deep generative model and generative adversarial networks collectively under a trainable unified framework addressing the shortage of labeled data. At last, upon the proposed models, we develop several real-world BCI applications such as an EEG-based person identification system and a prototype of a brain-controlled typing system which converts user's thoughts into text.

Signature	Witness Signature	Date
.....	.....	04/11/2019

FOR OFFICE USE ONLY Date of completion of requirements for Award:

ORIGINALITY STATEMENT

‘I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, or substantial proportions of material which have been accepted for the award of any other degree or diploma at UNSW or any other educational institution, except where due acknowledgement is made in the thesis. Any contribution made to the research by others, with whom I have worked at UNSW or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the project's design and conception or in style, presentation and linguistic expression is acknowledged.’

Signed ……..

04/11/2019

Date …….....

INCLUSION OF PUBLICATIONS STATEMENT

UNSW is supportive of candidates publishing their research results during their candidature as detailed in the UNSW Thesis Examination Procedure.

Publications can be used in their thesis in lieu of a Chapter if:

- The student contributed greater than 50% of the content in the publication and is the "primary author", ie. the student was responsible primarily for the planning, execution and preparation of the work for publication
- The student has approval to include the publication in their thesis in lieu of a Chapter from their supervisor and Postgraduate Coordinator.
- The publication is not subject to any obligations or contractual agreements with a third party that would constrain its inclusion in the thesis

Please indicate whether this thesis contains published material or not.

☐ *This thesis contains no publications, either published or submitted for publication (if this box is checked, you may delete all the material on page 2)*

☒ *Some of the work described in this thesis has been published and it has been documented in the relevant Chapters with acknowledgement (if this box is checked, you may delete all the material on page 2)*

☐ *This thesis has publications (either published or submitted for publication) incorporated into it in lieu of a chapter and the details are presented below*

CANDIDATE'S DECLARATION

I declare that:

- I have complied with the Thesis Examination Procedure
- where I have used a publication in lieu of a Chapter, the listed publication(s) below meet(s) the requirements to be included in the thesis.

Name	**Signature**	**Date (dd/mm/yy)**
Xiang Zhang		04/11/2019

Copyright Statement

Authenticity Statement

Abstract

Brain-Computer Interface (BCI) bridges the human being's neural world and the outer physical world by decoding individuals' brain signals into commands recognizable by computer devices, which has attracted increasing attention in recent years. This dissertation aims at overcoming the hurdles and stretching the horizons of data-efficient BCI systems by developing robust deep representation learning paradigms. First, we present a comprehensive introduction of BCI systems including the deep learning models, state-of-the-art studies adopting deep learning for BCI drawbacks, and the appealing real-world BCI applications. Moreover, we propose automatic high-level representation learning methods through deep architectures addressing the traditional time-consuming manually feature engineering and low signal-to-noise ratio data. In addition, we develop reinforced selective attention mechanism by combining reinforcement learning and deep neural network for capturing informative representations adaptive to different scenarios. Furthermore, we design a weakly-supervised predictive model to harness the deep generative model and generative adversarial networks collectively under a trainable unified framework addressing the shortage of labeled data. At last, upon the proposed models, we develop several real-world BCI applications such as an EEG-based person identification system and a prototype of a brain-controlled typing system which converts user's thoughts into text.

Acknowledgments

First, I would like to express my sincere gratitude to my primary supervisor, the best supervisor in the world, Dr. Lina Yao. This dissertation would not be possible without Lina. From the first day of my Ph.D., Lina taught me how to be a good researcher hand-by-hand including literature review, topic narrow-down, model development, experiment design, academic writing, and so on. I could achieve nothing without Lina's excellent research insights and careful guidance. The extraordinary characteristics which shine in Lina's beautiful soul such as helpfulness, patience, rigor, and enthusiasm have a great impact on me and constantly have positive influence on my future life. Moreover, I am grateful for my secondary supervisor Prof. Wenjie Zhang for her warm encouragement and advices. Also I want to thank my co-supervisor Prof. Boualem Benatallah for his supporting.

Beside my supervisors, I would like to thank the senior teammate and my best friend Dr. Chaoran Huang. Chaoran gave me lots of great advices on administration, programming, experimental equipment test, daily lives, etc. In addition, I appreciate other lab members: Manqing Dong, Xiaodong Ning, Shuai Zhang for the stimulating discussions and all the fun we have had together.

Last but not the least, I want to thank my family: my parents, my younger sister and my girlfriend for their supporting throughout my Ph.D. in general.

Abbreviations

BCI Brain-Computer Interface

EEG Electroencephalography

DL Deep Learning

SNR Signal-to-Noise Ratio

KNN K-Nearest Neighbor

SVM Support Vector Machine

LDA Linear Discriminant Analysis

AdaBoost Adaptive Boosting

RF Random Forest

DT Decision Tree

MI EEG Motor Imagery EEG

S-EEG Sleeping EEG

MD EEG Mental Disorder EEG

ERP Event-Related Potentials

fNIRS functional Near-Infrared Spectroscopy

ERD Event-Related Desynchronization

ERS Event-Related Synchronization

SSVEP Steady-State Visual Evoked Potentials

VEP Visual Evoked Potentials

AEP Auditory Evoked Potentials

RSVP Rapid Serial Visual Presentation

EMG Electromyography

EOG Electrooculography

fMRI functional Magnetic Resonance Imaging

MLP Multi-Layer Perceptron

RNN Recurrent Neural Networks

CNN Convolutional Neural Networks

AE AutoEncoder

RBM Restricted Boltzmann Machine

DBN Deep Belief networks

VAE Variational AutoEncoder

GAN Generative Adversarial Networks

LSTM Long Short-Term Memory

GRU Gated Recurrent Units

D-AE Deep AutoEncoder

D-RBM Deep Restricted Boltzmann Machine

DBN-AE DBN stacked by AE

DBN-RBM DBN stacked by RBM

CA Classification Aimed

NCA Non-Classification Aimed

CSP Common Spatial Pattern

GA Genetic Algorithm

DE Differential Entropy

HCNN Hierarchical Convolutional Neural Network

HMM Hidden Markov Model

PD ParkinsonâĂŹs Disease

ICA Independent Component Analysis

PCA Principal Component Analysis

PSD Power Spectral Density

IoT Internet of Things

BCI-C BCI Competition

LR Logistic Regression

OA Orthogonal Array

ROC Receiver Operating Characteristic

CM Confusion Matrix

AUC Area Under the Curve

F1 F1 Score

DWT Discrete Wavelet Transform

RS Replicate and Shuffle

WAS-LSTM Weighted Average Spatial LSTM

DQN Deep Q Networks

FCL Fully Connected Layer

FAR False Acceptance Rate

FRR False Rejective Rate

AVAE Adversarial Variational Embedding

OATM Orthogonal Array Tuning Method

RFID Radio Frequency IDentification

BO Bayesian Optimization

DGR Dynamical Graph Representation

Contents

List of Figures

List of Tables

List of Publications

This dissertation contains parts of contexts of the following publications:

[1] X. Zhang, L. Yao, X. Wang, J. Monaghan, D. Mcalpine, and Y. Zhang, "A survey on deep learning based brain computer interface: Recent advances and new frontiers," in *Journal of Neural Engineering*, 2018, (Under Review, SJR Q1).

[2] X. Zhang, L. Yao, C. Huang, Q. Z. Sheng, and X. Wang, "Intent recognition in smart living through deep recurrent neural networks," in *International Conference on Neural Information Processing (ICONIP 2017)*, pp. 748–758, Springer, 2017, (CORE Rank A).

[3] X. Zhang, L. Yao, D. Zhang, X. Wang, Q. Z. Sheng, and T. Gu, "Multi-person brain activity recognition via comprehensive eeg signal analysis," in *Proceedings of the 14th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (MobiQuitous 2017)*, pp. 28–37, ACM, 2017, (CORE Rank A).

[4] X. Zhang, L. Yao, C. Huang, S. Wang, M. Tan, G. Long, and C. Wang, "Multi-modality sensor data classification with selective attention," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI 2018)*, 2018, (CORE Rank A*).

[5] X. Zhang, L. Yao, X. Wang, W. Zhang, S. Zhang, and Y. Liu, "Know your mind: Adaptive brain signal classification with reinforced attentive convolutional neural networks," in *The 2019 IEEE International Conference on Data Mining (ICDM 2019)*, 2019, (CORE Rank A*).

[6] X. Zhang, L. Yao, and F. Yuan, "Adversarial variational embedding for robust semi-supervised learning," *The 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD 2019)*, 2019, (CORE Rank A*).

[7] X. Zhang, X. Chen, L. Yao, C. Ge, and M. Dong, "Deep neural network hyper-parameter optimization with orthogonal array tuning," *The 26th International Conference On Neural Information Processing (ICONIP 2019)*, 2019, (CORE Rank A).

[8] X. Zhang, L. Yao, S. S. Kanhere, Y. Liu, T. Gu, and K. Chen, "Mindid: Person identification from brain waves through attention-based recurrent neural network," *the ACM International Conference on Pervasive and Ubiquitous Computing (UbiComp 2018)*, vol. 2, no. 3, p. 149, 2018, (CORE Rank A*).

[9] X. Zhang, L. Yao, Q. Z. Sheng, S. S. Kanhere, T. Gu, and D. Zhang, "Converting your thoughts to texts: Enabling brain typing via deep feature learning of eeg signals," in *2018 IEEE International Conference on Pervasive Computing and Communications (PerCom 2018)*, pp. 1–10, IEEE, 2018, (CORE Rank A*).

[10] X. Zhang, L. Yao, C. Huang, S. S. Kanhere, D. Zhang, and Y. Zhang, "Brain2object: Printing your mind from brain signals with spatial correlation embedding," *Pattern Recognition*, 2018, (Under Review, SJR Q1).

[11] X. Zhang, X. Chen, M. Dong, H. Liu, C. Ge, and L. Yao, "Multi-task generative adversarial learning on geometrical shape reconstruction from eeg brain signals," in *The 26th International Conference On Neural Information Processing (ICONIP 2019)*, 2019, (CORE Rank A).

I have also participated in the following publications during my candidature:

[12] X. Zhang, L. Yao, S. Zhang, S. Kanhere, M. Sheng, and Y. Liu, "Internet of things meets brain–computer interface: A unified deep learning framework for enabling human-thing cognitive interactivity," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2084–2092, 2018, (SJR Q1).

[13] X. Zhang, "Context-aware human intent inference for improving human machine cooperation," in *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom 2018)*, pp. 456–457, IEEE, 2018, (CORE Rank A*).

[14] X. Zhang, L. Yao, M. Dong, Z. Liu, Y. Zhang, and Y. Li, "Adversarial representation learning for robust patient-independent epileptic seizure detection," in *Journal of Biomedical and Health Informatics (JBHI)*, 2019, (Major Revision, SJR Q1).

[15] X. Zhang, L. Yao, C. Huang, T. Gu, Z. Yang, and Y. Liu, "Deepcode: A multimodal biometric authentication system via deep decoding gaits and brainwaves," *ACM Transaction on Intelligent Systems and Technology (TIST)*, 2019, (Major Revision, SJR Q1).

[16] X. Ning, L. Yao, X. Wang, B. Benatallah, S. Zhang, and X. Zhang, "Data-augmented regression with generative convolutional network," in *International Conference on Web Information Systems Engineering (WISE)*, pp. 301–311, Springer, 2018, (CORE Rank A).

[17] X. Chen, C. Huang, x. Zhang, W. Liu, and L. Yao, "Distributed expert representation learning in question answering community," in *The 15th Advanced Data Mining and Applications (ADMA 2019)*, 2019.

[18] M. Dong, L. Yao, X. Wang, B. Benatallah, X. Zhang, and Q. Sheng, "Dual-stream self-attentive random forest for false information detection," in *International Joint Conference on Neural Networks (IJCNN 2019)*, 2019, (CORE Rank A).

[19] W. Chen, S. Wang, X. Zhang, L. Yao, L. Yue, B. Qian, and X. Li, "Eeg-based motion intention recognition via multi-task rnns," in *Proceedings of the 2018 SIAM International Conference on Data Mining (SDM)*, pp. 279–287, SIAM, 2018, (CORE Rank A).

[20] D. Zhang, L. Yao, X. Zhang, S. Wang, W. Chen, and R. Boots, "Eeg-based intention recognition from spatio-temporal representations via cascade and parallel convolutional recurrent neural networks," *The 32nd AAAI Conference on Artificial Intelligence (AAAI 2018)*, 2018, (CORE Rank A*).

[21] C. Huang, L. Yao, X. Wang, B. Benatallah, and X. Zhang, "Software expert discovery via knowledge domain embeddings in a collaborative network," *Pattern Recognition Letters*, 2018, (SJR Q2).

[22] S. Zhang, L. Yao, B. Wu, X. Xu, X. Zhang, and L. Zhu, "Unraveling metric vector spaces with factorization for recommendation," *IEEE Transactions on Industrial Informatics*, 2019, (SJR Q1).

[23] L. Wang, X. Zhang, Y. Zhang, Q. Gu, and L. Yao, "Back-gesture: Gesture recognition on the back of mobile phones using acoustic signals," in *The IEEE International Conference on Computer Communications (INFOCOM 2020)*, 2019, (Under Review, CORE Rank A*).

[24] W. Xu, X. Zhang, C. Luo, L. Yao, J. Li, Z. Ming, and B. Wei, "A multi-view cnn-based acoustic classificationsystem for automatic animal specie identification," in *Ad Hoc Networks*, 2019, (Major Revision, SJR Q1).

Part I

Introduction and Literature Review

This part contains work published in:

[1] **X. Zhang**, L. Yao, X. Wang, J. Monaghan, D. Mcalpine, and Y. Zhang. A Survey on Deep Learning based Brain Computer Interface: Recent Advances and New Frontiers. *Journal of Neural Engineering*, 2019. (Under Review, SJR Q1)

Chapter 1

Introduction

1.1 Background

Brain-Computer Interface (BCI)[1] is a system that translates activity patterns of the human brain into messages or commands to communicate with the outer world [25]. BCI underpins many novel applications that are important to people's daily life, especially to people with psychological/physical disorders or disabilities. BCI can assist the disabled, elders and people with limited motion ability (e.g., people with muscle diseases) in controlling wheelchairs, home appliances, and robots. For instance, a BCI system can control household appliances through patients' brain signals. In addition, ordinary individuals can enjoy enhanced entertainment and security when brain waves-based techniques are applied for high fake-resistant user identification [8]. Generally, a BCI system contains a bidirectional communication between the human brain and the computer. However, as introduced in [26], sys-

[1]There are several terms similar to BCI, e.g., Brain Machine Interface (BMI), Brain Interface (BI), Direct Brain Interface (DBI), and Adaptive Brain Interface (ABI). They all describe machines that are directly controlled by human brain signals.

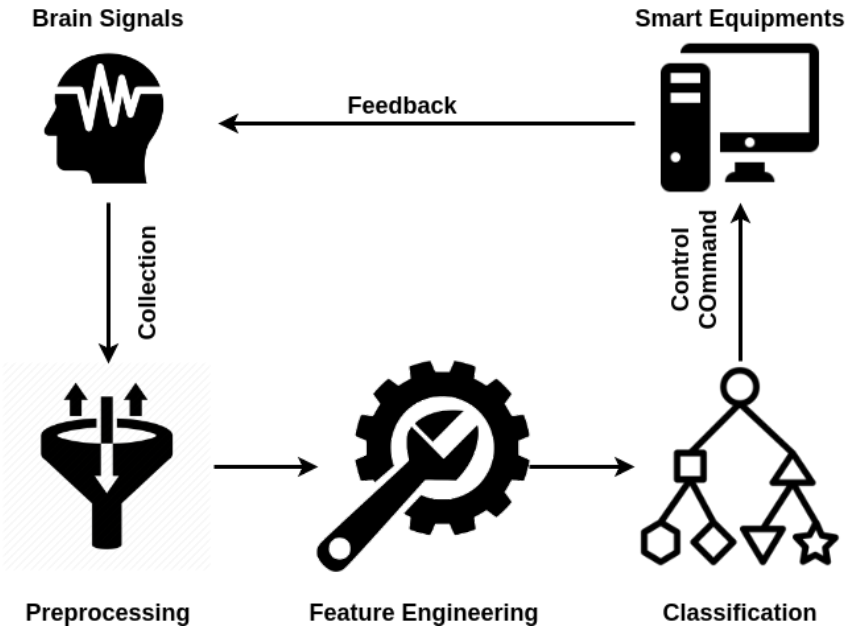


Figure 1.1: Generally workflow of BCI system.

tems based on brain signal analysis (such as mental disease diagnosis, emotional computation, sleeping state scoring, etc.) can also be considered as a more generalized BCI. This thesis investigates the generalized BCI systems to provide a more comprehensive and vast scope of BCI applications

Figure 1.1 shows the general paradigm of a BCI system, which receives brain signals and converts them into control commands for computers. The system includes several key components: brain signal collection, signal preprocessing, feature engineering, classification, and smart equipment. The brain signals are collected from humans and sent to the preprocessing component for denoising and enhancement. Then, the discriminating features are extracted from the processed signals and sent to the classifier, which recognizes the signals and convert then into external device commands.

The collection methods differ from signal to signal. For example, EEG signals measure the voltage fluctuation resulting from ionic current within the neurons of the brain. Collecting EEG signals requires placing a series of electrodes on the scalp of the human head to record the electrical activity of the brain. Since the ionic current

generated within the brain is measured at the scalp, obstacles (e.g., skull) greatly decrease the signal quality—the fidelity of the collected EEG signals, measured as Signal-to-Noise Ratio (SNR), is approximately 5% of that of original brain signals [27]. Therefore, brain signals are usually preprocessed before feature engineering to increase the SNR. The preprocessing component contains multiple steps such as signal cleaning (e.g., smoothing the noisy signals or resolving the inconsistencies), signal normalization (e.g., normalizing each channel of the signals along time-axis), signal enhancement (e.g., removing direct current), and signal reduction (presenting a reduced representation of the signal).

Feature engineering refers to the process of extracting discriminating features from the input signals through domain knowledge. Traditional features are extracted from time-domain (e.g., variance, mean value, kurtosis), frequency-domain (e.g., fast Fourier transform), and time-frequency domains (e.g., discrete wavelet transform). They will enrich distinguishable information regarding user intention. Feature engineering is highly dependent on the domain knowledge. For example, biomedical knowledge is required to extract features from brain signals of epileptic seizures. Manual feature extraction is also time-consuming and difficult. Recently, deep learning provides a better option to automatically extract distinguishable features.

The classification component refers to the machine learning algorithms that classify the extracted features into logical control signals recognizable by external devices. Deep learning algorithms are shown to be more powerful than traditional classifiers such as Support Vector Machine (SVM) and Linear Discriminant Analysis (LDA).

1.2 Challenges

Although traditional BCI systems have made tremendous progress [26, 28] in the past decades, constructing a reliable BCI system that continuously interacts with an ever-changing environment can be challenging due to the high uncertainties of

capturing human brainwaves and complexity of human-machine interactions. We identify the following several major challenges:

First, brain signals are easily corrupted by various biological (e.g., eye blinks, muscle artifacts, fatigue and concentration level) and environmental artifacts (e.g., environmental noise) [26]. Moreover, BCI has a low SNR due to the non-stationary nature of electro-physiological brain signals [29]. Although several preprocessing and feature engineering methods have been developed to decrease the noise level, such methods (e.g., feature selection and extraction both in the time domain and frequency domain) are time-consuming and may cause information loss in the extracted features [9]. Therefore, it is crucial to automatically distill informative data from corrupted brain signals and build a robust BCI system.

Second, the feature engineering stage highly depends on human expertise in the specific domain. For example, it requires basic knowledge of biology to investigate the sleep state through EEG signals. Human experience may help capture features on some particular aspects but prove insufficient in more general conditions. For instance, the model performs well on sleep state recognition may fail in human intention recognition. In addition, the domain-specific feature extraction decrease the generalization ability of BCI systems. Therefore, an algorithm is required to automatically extract representative features adapting to the specific scenarios.

Third, the labeling of brain signals is expensive in terms of finance and human source. The acquisition of labeled neurological data generally requires a skilled experiment conductor and professional equipment. The cost associated with the labeling process thus may render a fully labeled training set infeasible, whereas acquisition of unlabeled data is relatively inexpensive. For instance, the labels of the epileptic seizure signals are manually judged by at least three neurologists while the unlabelled signals can be easily collected from the hospital. As a result, the BCI researchers occasionally handle with a large set of brain signal observations but only have a small proportion labels. Thus, it is necessary to explore an effective algorithm to achieve a competitive performance even in the situation where only a

few labels are available.

1.3 Contributions

Addressing the aforementioned challenges, we develop a set of machine learning algorithms and frameworks to enable and enhance data-Efficient deep representation learning in BCI systems and its applications. In this dissertation, we mark the following contributions:

- First, we propose novel deep representation learning frameworks to automatically discover the high-level discriminative features from the raw brainwaves. Until now, deep learning has been applied extensively in BCI scenarios and shown success in addressing the above challenges [30, 31]. Deep learning possess two advantages. To begin with, it avoids the time-consuming preprocessing and feature engineering steps by working directly on raw brain signals to learn distinguishable information through back-propagation. In addition, deep neural networks can capture both representative high-level features and latent dependencies through deep structures. (Section 3 to Section 4)

- Second, aiming at the adaptive representative feature learning, we proposed a reinforced selective attention framework by combining the benefits of reinforcement learning, deep neural network (e.g., recurrent neural networks and convolutional neural networks), and attention mechanism for capturing informative temporal and spatial representations from the raw EEG signals. The propose approach is enabled to pay attention to the most distinctive features according to the specific scenario. (Section 5 to Section 6)

- Third, we investigate the weakly-supervised problem and proposed a novel adversarial variational embedding approach to harness the deep generative

model and generative adversarial networks collectively under a trainable unified framework. The proposed algorithm is sought for leveraging the unlabelled data when labelled data is difficult or expensive to acquire, which is also demonstrated in several application scenarios such as neurological diagnosis. (Section 7 to Section 8)

- At last, upon the proposed data-efficient representation learning frameworks, we develop several real-world BCI applications including person identification, brain typing, and geometrical shape reconstruction. We build end-to-end prototypes and conduct extensive experiments to demonstrate the effectiveness and efficiency. (Section 9 to Section 12)

The rest of this dissertation is structured as followed. Part 2 provides a comprehensively introduction of recent advances on deep representation learning-based BCI studies, along with the corresponding applications. Part II presents the proposed deep learning frameworks to explore the latent inter-subject and intra-subject dependencies targeting intention recognition in the context of person-dependent and cross-person situations, respectively. Part III shows the reinforced selective attention frameworks which can adaptively learn the discriminative features. Part IV introduces the proposed weakly-supervised learning framework and a novel hyper-parameter tuning method in order to learn the optimal parameter setting. Moreover, Part V presents several end-to-end real-world BCI applications. At last, Part VI summarizes the key conclusions of this dissertation and points out the future directions.

Chapter 2

Literature Review

In this chapter, we present a comprehensive review of the recent advances in deep learning-based BCI systems including the basic knowledge of deep learning models, the state-of-the-art studies on deep learning-powered BCI, and common-used applications.

Table 2.1 shows a summary of the existing survey on BCI. To the best of our knowledge, the limited existing surveys [32, 33, 34, 26, 35, 25, 28, 36] only focus on partial Electroencephalography (EEG) signals. For example, Lotte et al. [35] and Wang et al. [37] focus on EEG without analyzing EEG signal types; Cecotti et al. [38] focus on Event-Related Potentials (ERP); Haseer et al. [39] focus on functional Near-Infrared Spectroscopy (fNIRS); Fatourechi et al. [33] only focus on EMG and EOG; Mason et al. [36] brief the neurological phenomenons like Event-Related Desynchronization (ERD), P300, SSVEP, Visual Evoked Potentials (VEP), Auditory Evoked Potentials (AEP) but have not organized them systematically; Abdulkader et al. [26] present a topology of brain signals but have not mentioned spontaneous EEG and Rapid Serial Visual Presentation (RSVP).; Lotte et al. [25] have not considered ERD and RSVP; Roy et al. [40] list some deep learning based EEG studies but provide little analysis.

Moreover, although some overviews have conducted in deep learning ([41, 42, 43]) and BCI ([33, 26, 35, 25, 28, 36]), few focus on their combination. To the best of our knowledge, this thesis provides the first comprehensive summary of the recent advances on deep learning-based BCI.

Lastly, all previous BCI surveys focus on specific areas or applications without given an overview of the broad scenarios. For example, Litjens et al. [44] review some leading deep learning concepts pertinent to medical image analysis without covering many other deep learning models; Soekadar et al. [45] review the BCI systems and machine learning methods that help overcome stroke-related motor paralysis and focus on Somatosensory Rhythms (SMR); Vieira et al. [46] investigate the application of BCI systems on neurological and psychiatric disorders. This chapter provide a wide range of BCI applications from healthcare, smart environment, to security.

In summary, addressing the aforementioned issues, we present a comprehensive and systematic introduction of the recent advances and new frontiers of deep learning based brain-computer interface techniques. We summarize over 230 contributions in this field, most of which were published in the last five years.

2.1 Deep Learning Models

In this section, we formally introduce the deep learning models including concepts, architectures, and techniques commonly used in the BCI field. Deep learning is a class of machine learning techniques that uses many layers of information-processing stages in hierarchical architectures for pattern classification and feature/representation learning [42].

In this thesis, we will give relative detail introduction of various deep learning models for the reason that a part of the potential readers who are from non-computer area (e.g., biomedical) are not familiar to deep learning.

Table 2.1: The existing survey on BCI in the last decade. The column 'comprehensive on signals' indicates whether the survey has summarized all the BCI signals or not. fMRI refers to functional Magnetic Resonance Imaging.

No.	Reference	Comprehensive on Signals?	Signal	Deep Learning	Publication Time	Area
1	[33]	No	EMG, EOG	No	2007	
2	[32]	No	fMRI	Yes	2018	Mental Disease Diagnosis
3	[35]	Partial	MI EEG and P300	No	2007	Classification
4	[25]	Partial	MI EEG and P300	Partial	2018	Classification
6	[44]	No	MRI, CT	Partial	2017	Medical Image Analysis
7	[40]	No	EEG	Yes (but without any model introduction)	2019	
8	[28]	No	EEG	No	2007	Signal Processing
9	[37]	Partial	EEG	No	2016	BCI Applications
10	[26]	Yes		No	2015	
11	[47]	No	EEG	Partial (only DBN)	2018	
12	[45]	No	EEG, fMRI	No	2015	Neuro-rehabilitation of Stroke
13	[48]	No	MI EEG	No	2015	
14	[49]	No	fMRI	No	2014	
15	[50]	No	ERP (P300)	No	2017	Applications of ERP"
16	[34]	No	fMRI	Yes	2018	Applications of fMRI
17	[51]	No	ERP	No	2017	Classification
18	[52]	Partial	EEG	No	2019	Brain Biometrics
19	Ours	yes	Systematic EEG	Yes		

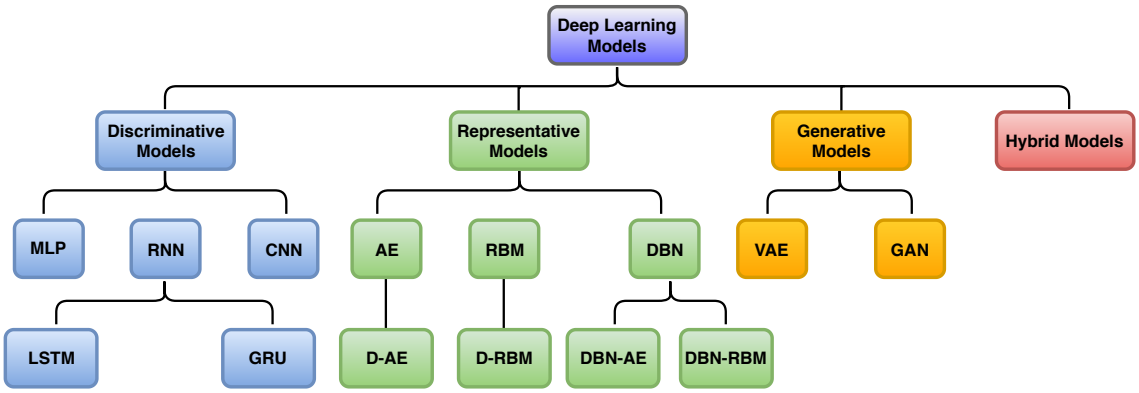


Figure 2.1: Deep learning models. They can be divided into discriminative, representative, generative and hybrid models based on the algorithm function. D-AE denotes Stacked-Autoencoder which refers to the autoencoder with multiple hidden layers. Deep Belief Network can be composed of AE or RBM, therefore, we divided DBN into DBN-AE (stacked AE) and DBN-RBM (stacked RBM).

Deep learning algorithms divide into several subcategories based on the aim of the techniques (as shown in Figure 2.1):

- Discriminative deep learning models, which classify the input data into a pre-known label based on the adaptively learned discriminative features. Discriminative algorithms are able to learn distinctive features by non-linear transformation, and classification through probabilistic prediction[1]. Thus these algorithms can play the role of both feature engineering and classification (corresponding to Figure 1.1). Discriminative architectures mainly include Multi-Layer Perceptron (MLP), Recurrent Neural Networks (RNN), Convolutional Neural Networks (CNN), along with their variations.

- Representative deep learning models, which learn the pure and representative features from the input data. These algorithms only have the function of feature engineering (corresponding to Figure 1.1) but fail to classify. Commonly used deep learning algorithms for representation are Autoencoder (AE), Restricted Boltzmann Machine (RBM), Deep Belief networks (DBN), along with their variations.

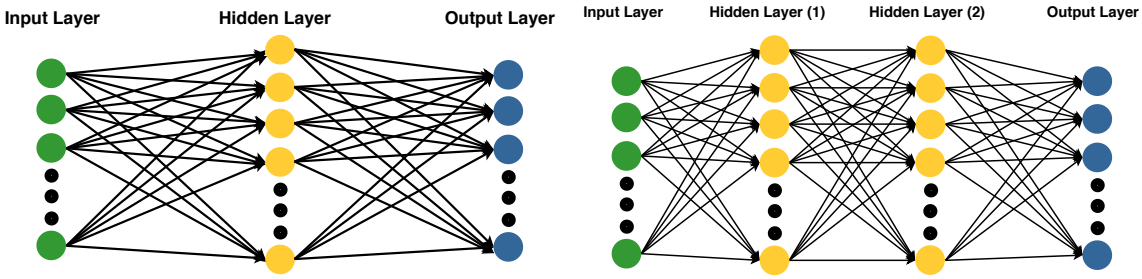
- Generative deep learning models, which learn the joint probability distribution of the input data and the target label. In the BCI scope, generative algorithms are mostly used in reconstruction or to generate a batch of brain signals samples to enhance the training set. Generative models commonly used in BCI include variational Autoencoder (VAE)[2], Generative Adversarial Networks (GANs), etc.

[1]The classification function is achieved by the combination of a softmax layer and one-hot label encoding. The one-hot label encoding refers to encoding the label by the one-hot method, which is a group of bits among which the only valid combinations of values are those with a single high (1) bit and all the others low (0) bits. For instance, a set of labels 0, 1, 2, 3 can be encoded as (1, 0, 0, 0), (0, 1, 0, 0), (0, 0, 1, 0), (0, 0, 0, 1).

[2]VAE is a variation of AE, but working on a different aspect. Therefore, we separately introduce AE and VAE.

Table 2.2: Summary of deep learning model types

Deep Learning	Input	Output	Function	Training method
Discriminative	Input data	Label	Feature extraction, Classification	Supervised
Representative	Input data	Representation	Feature extraction	Unsupervised
Generative	Input data	New Sample	Generation, Reconstruction	Unsupervised
Hybrid	Input data	–	–	–

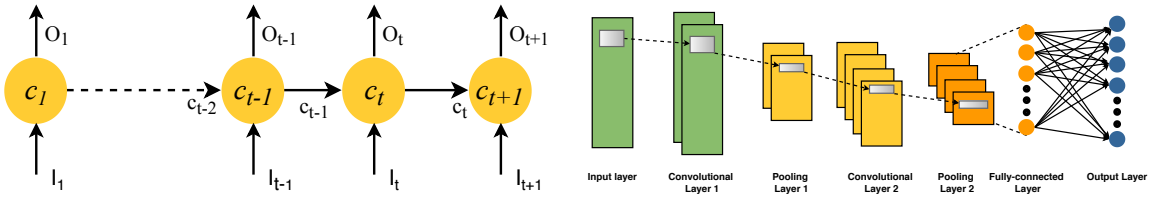


(a) Basic fully-connected neural network (b) Multi-Layer Perceptron

Figure 2.2: Illustration of standard neural network and multilayer perceptron. (a) The basic structure of the fully-connected neural network. The input layer receives the raw data or extracted features of brain signals while the output layer shows the classification results. The term 'fully-connected' denotes each node in a specific layer is connected with all the nodes in the previous and next layer. (b) MLP could have multiple hidden layers, the more, the deeper. This is an example of MLP with two hidden layers, which is the simplest MLP model.

- Hybrid deep learning models, which combine more than two deep learning models. For example, the typical hybrid deep learning model employs a representation algorithm for feature extraction and discriminative algorithms for classification.

The summary of the characteristics of each deep learning subcategories are listed in Table 2.2. Almost all the classification functions in neural networks are implemented by a softmax layer, which will not be regarded as an algorithmic component. For instance, a model combining a DBN, and a softmax layer will still be regarded as a representative model instead of a hybrid model.



(a) Recurrent Neural Networks (b) Convolutional Neural Networks

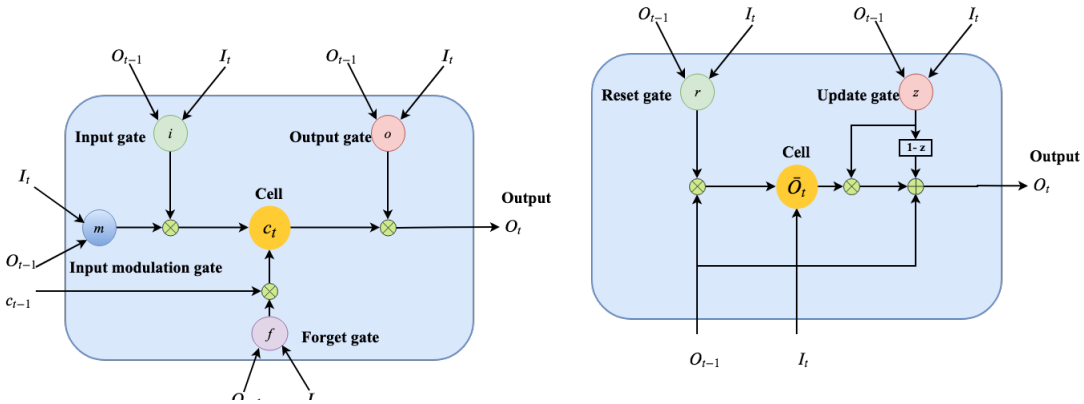
Figure 2.3: Illustration of RNN and CNN models. (a) The recurrent procedure of the RNN model. This procedure describes the recurrent procedure of a specific node in time range $[1, t+1]$. The node at time t receives two inputs variables (I_t denotes the input at time t and c_{t-1} denotes the hidden state at time $t-1$) and exports two variables (the output O_t and the hidden state c_t at time t). (b) The paradigm of CNN model which includes two convolutional layers, two pooling layers, and one fully-connected layer.

2.1.1 Discriminative Deep Learning Models

Since the main task of BCI is brain signal recognition, the discriminative deep learning models are the most popular and powerful algorithms. Suppose we have a dataset of brain signal samples $\{\mathbb{X}, \mathbb{Y}\}$ where $\mathbb{X}$ denotes the set of brain signal observations and $\mathbb{Y}$ denotes the set of sample ground truth (i.e., labels). Suppose a specific sample-label pair $\{\boldsymbol{x} \in \mathbb{R}^N, \boldsymbol{y} \in \mathbb{R}^M\}$ where N and M denote the dimension of observations and the number of sample categories, respectively. The aim of discriminative deep learning models is to lean a function with the mapping: $\boldsymbol{x} \rightarrow \boldsymbol{y}$. In short, the discriminative models receive the input data and output the corresponding category or label. All the discriminative models introduced in this section are supervised learning techniques which require the information of both the observations and the ground truth.

Multi-Layer Perceptron (MLP)

Multilayer Perceptron, one of the simplest and the most basic deep learning model, is modified based on the standard neural network (Figure 2.2a) which contains three neuron layers (i.e., an input layer, a hidden layer, and an output layer). The key



(a) Structure of LSTM cell (b) Structure of GRU cell

Figure 2.4: Illustration of detailed LSTM and GRU cell structures. (a) LSTM cell receives three inputs (I_t denotes the input at time t, O_{t-1} denotes the output of previous time, and c_{t-1} denotes the hidden state of the previous time) and exports two outputs (the output of this time O_t and the hidden state of this time c_t). (b) GRU cell receives two inputs (the input of this time I_t and the output of the previous time O_{t-1}) and exports its output O_t. Unlike the hidden state c_t in LSTM cell, there is no transmittable hidden state in GRU cell except one intermediate variable $\bar{O}_t$.

difference between MLP and the standard neural network is that MLP has more than one hidden layers. All the nodes are fully-connected with the nodes of the adjacent layers but without connection with the other nodes of the same layer. MLP includes multiple hidden layers. As shown in Figure 2.2b, we take a structure with two hidden layers as an example to describe the data flow in MLP. First, we define an operation $\mathcal{T}(\cdot)$ as

$$\mathcal{T}(\boldsymbol{x}) = \boldsymbol{w} * \boldsymbol{x} + \boldsymbol{b} \tag{2.1}$$

$$\mathcal{T}(\boldsymbol{x}, \boldsymbol{x'}) = \boldsymbol{w} * \boldsymbol{x} + \boldsymbol{b} + \boldsymbol{w'} * \boldsymbol{x'} + \boldsymbol{b'} \tag{2.2}$$

where $\boldsymbol{x}$ and $\boldsymbol{x'}$ denote two variables while $\boldsymbol{w}$, $\boldsymbol{w'}$, $\boldsymbol{b}$, and $\boldsymbol{b'}$ denote the corresponding weights and basis.

The input layer receives the observation $\boldsymbol{x}$ and feeds forward to the first hidden layer,

$$\boldsymbol{x^{h1}} = \sigma(\mathcal{T}(\boldsymbol{x})) \tag{2.3}$$

where $\boldsymbol{x^{h1}}$ denotes the data flow in the first hidden layer and σ represents the non-

linear activation function. There several commonly used activation function such as sigmoid/Logistic, Tanh, ReLU, we choose sigmoid activation function as an example in this section. Then, the data flow to the second hidden layer and the output layer,

$$\boldsymbol{x^{h2}} = \sigma(\mathcal{T}(\boldsymbol{x^{h1}})) \tag{2.4}$$

$$\boldsymbol{y'} = \sigma(\mathcal{T}(\boldsymbol{x^{h2}})) \tag{2.5}$$

where $\boldsymbol{y'}$ denotes the predict results in one-hot format. The error (i.e., loss) could be calculated based on the distance between $\boldsymbol{y'}$ and the ground truth $\boldsymbol{y}$. For instance, the Euclidean-distance based error can be calculated by

$$error = \|\boldsymbol{y'} - \boldsymbol{y}\|_2 \tag{2.6}$$

where $\|\cdot\|_2$ denotes the Euclidean norm. Afterward, the error will be back-propagated and optimized by a suitable optimizer. The optimizer will adjust all the weights and basis in the model until the error converges. The most widely used loss functions includes cross-entropy, negative log likelihood, mean square estimation, etc. The most widely used optimizers include Adaptive moment estimation (Adam), Stochastic Gradient Descent (SGD), Adagrad (Adaptive sub-gradient method), etc.

Several terms may be easily confused with each other: Artificial Neural Network (ANN), Deep Neural Network (DNN), and MLP. These terms have no strict difference and often mixed in literature. Generally, ANN represents neural networks with fewer hidden layers (shallow) while DNN have more (in this case, DNN is equivalent to MLP).

Recurrent Neural Networks (RNN)

Recurrent Neural Network is a specific subclass of discriminative deep learning model which are designed to capture temporal dependencies among input data. Figure 2.3a describes the activity of a specific RNN node in the time domain. At each time ranges from $[1, t+1]$, the node receives an input I [3] and a hidden state c from the

[3]The subscript represents the specific time.

previous time (except the first time). For instance, at time t it receives not only the input I_t but also the hidden state of the previous node c_{t-1}. The hidden state can be regarded as the 'memory' of the nodes which can help the RNN 'remember' the historical input.

Next, we will report two typical RNN architectures which have attracted much attention and achieved great success: long short-term memory and gated recurrent units. They both follow the basic principles of RNN, and we will pay our attention to the complicated internal structures in each node. Since the structure is much more complicated than general neural nodes, we call it a 'cell.' Cells in RNN are equivalent to nodes in MLP.

Long Short-Term Memory (LSTM). Figure 2.4a shows the structure of a single LSTM cell at time t. The LSTM cell has three inputs (I_t, O_{t-1}, and c_{t-1}) and two outputs (c_t and O_t). The operation is as follows:

$$I_t, O_{t-1}, c_{t-1} \rightarrow c_t, O_t \tag{2.7}$$

I_t denotes the input value at time t, O_{t-1} denotes the output at the previous time (i.e., time $t-1$), and c_{t-1} denotes the hidden state at the previous time. c_t and O_t separately denote the hidden state and the output at time t. Therefore, we can observe that the output O_t at time t not only related to the input I_t but also related to the information at the previous time. In this way, LSTM is empowered to remember the important information in the time domain. Moreover, the essential idea of LSTM is to control the memory of specific information. For this aim, LSTM cell adopts four gates: the input gate, forget gate, output gate, and input modulation gate. Each gate is a weight to control how much information can flow through this gate. For example, if the weight of the forget gate is zero, the LSTM cell would remember all the information passed from the previous time $t-1$; if the weight is one, the LSTM cell would remember nothing. The corresponding activation function determines the weight. The detailed data flow as follows:

$$f = \sigma(\mathcal{T}(I_t, O_{t-1})) \tag{2.8}$$

$$i = \sigma(\mathcal{T}(I_t, O_{t-1})) \tag{2.9}$$

$$o = \sigma(\mathcal{T}(I_t, O_{t-1})) \tag{2.10}$$

$$m = tanh(\mathcal{T}(I_t, O_{t-1})) \tag{2.11}$$

$$c_t = f * c_{t-1} + i * m \tag{2.12}$$

$$h_t = o * tanh(c_t) \tag{2.13}$$

where i, f, o and m represent the input gate, forget gate, output gate and input modulation gate, respectively.

Gated Recurrent Units (GRU). Another widely used RNN architecture is GRU. Similar to LSTM, GRU attempts to exploit the information from the past. GRU does not require hidden states, however, it receives temporal information only from the output of time $t-1$. Thus, as shown in Figure 2.4b, GRU has two inputs (I_t and O_{t-1}) and one output (O_t). The mapping can be described as:

$$I_t, O_{t-1} \rightarrow O_t \tag{2.14}$$

GRU contains two gates: reset gate r and update gate z. The former decides how to combine the input with previous memory. The latter decides how much of previous memory to keep around, which is similar to the forget gate of LSTM. The data flow as follows:

$$z = \sigma(\mathcal{T}(I_t, O_{t-1})) \tag{2.15}$$

$$r = \sigma(\mathcal{T}(I_t, O_{t-1})) \tag{2.16}$$

$$\bar{O}_t = tanh(\mathcal{T}(I_t, r * O_{t-1})) \tag{2.17}$$

$$O_t = (1 - z) * O_{t-1} + z * \bar{O}_t \tag{2.18}$$

It can be observed that there's a intermediate variable $\bar{O}_t$ which is similar to the hidden state of LSTM. However, $\bar{O}_t$ only works on this time point and unable to pass to the next time point.

We here give a brief comparison between LSTM and GRU since they are very similar. First, LSTM and GRU have comparable performance as studied by literature. For

any specific task, it is recommended to try both of them to determine which provides better performance. Second, GRU is lightweight since it only has two gates and without the hidden state. Therefore, GRU is faster to train and requires few data for generalization. Third, in contrast, LSTM generally works better if the training dataset is big enough.

Convolutional Neural Networks (CNN)

Convolutional Neural Networks is one of the most popular deep learning models specialized in spatial information exploration. This section will briefly introduce the working mechanism of CNN. CNN is widely used to discover the latent spatial information in applications such as image recognition, ubiquitous, and object searching due to their salient features such as regularized structure, good spatial locality, and translation invariance. In BCI, specifically, CNN is supposed to capture the distinctive dependencies among the patterns associated with different brain signals.

We present a standard CNN architecture as shown in Figure 2.3b. The CNN contains one input layer, two convolutional layers with each followed by a pooling layer, one fully-connected layer, and one output layer. The square patch in each layer shows the processing progress of a specific batch of input values. The key to the CNN is to reduce the input data into a form which is easier to recognize, with as little information loss as possible. CNN has three stacked layers: the convolutional Layer, pooling Layer, and fully-connected Layer.

The convolutional layer is the core block of CNN, which contains a set of filters to convolve the input data followed by a nonlinear transformation to extract the geographical features. In the deep learning implementation, there are several key hyper-parameters should be set in the convolutional layer, like the number of filters, the size of each filter, etc. The pooling layer generally follows the convolutional layer. The pooling layer aims to reduce the spatial size of the features progressively. In this way, it can help to decrease the number of parameters (e.g., weights and

basis) and the computing burden. There are three kinds of pooling operation: max, min, average. Take max pooling for example. The pooling operation outputs the maximum value of the pooling area as a result. The hyper-parameters in the pooling layer includes the pooling operation, the size of the pooling area, the strides, etc. In the fully-connected layer, as in the basic neural network, the nodes have full connections to all activations in the previous layer.

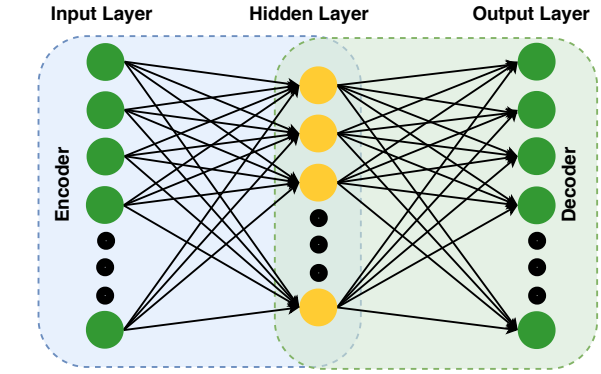
The CNN is the most popular deep learning model in BCI research, which can be used to exploit the latent spatial dependencies among the input brain signals like fMRI image, spontaneous EEG, and so on. More details will be reported in Section 2.2.

2.1.2 Representative Deep Learning Models

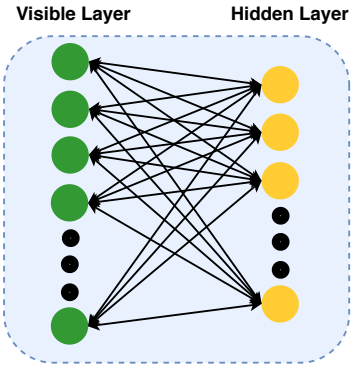
The essential blocks of representative deep learning models are Autoencoders, and restricted Boltzmann machines[4]. Deep Belief Networks are composed of AE or RBM. The representative models including AE, RBM[5], and DBN, are unsupervised learning methods. Thus, they can learn the representative features from only the input observations $\boldsymbol{x}$ without the ground truth $\boldsymbol{y}$. In short, representative models receive the input data and output a dense representation of the data. There are various definitions in different studies for several models (such as DBN, Deep RBM, and Deep AE), we choose the most understandable definitions and will present them in detail in this section.

[4]AE and RBM are generally regarded as kind of deep learning although they only have three and two layers, respectively.

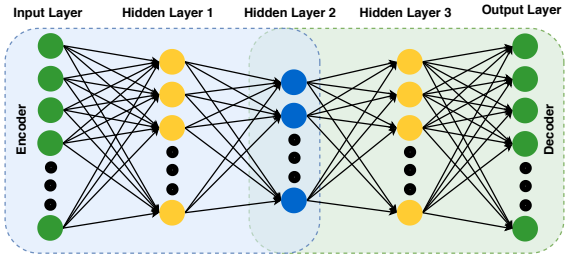
[5]We regard AE, and RBMas representative methods as most researches in BCI adopt them for feature representation.



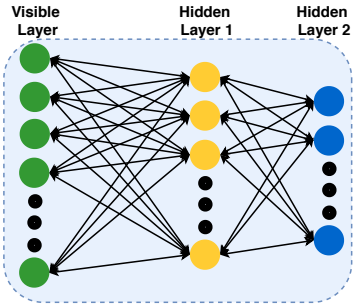
(a) Autoencoder



(b) Restricted Boltzmann Machine



(c) Deep AE



(d) Deep RBM

Figure 2.5: Illustration of several standard representative deep learning models. (a) A basic autoencoder contains three layers where the input layer and the output layer are supposed to have the same values. The process from the input layer to the hidden layer is an encoder while the process from the hidden layer to the output layer is a decoder. (b) In the Restricted Boltzmann Machine, the encoder and the decoder share the same transformation weights. The input layer and the output layer are merged into the visible layer. (c) The stacked autoencoder has more than one hidden layer. Generally, the number of hidden layers is odd, and the middle layer is the learned representative features. (d) The deep RBM has one visible layer and multiple hidden layers, the last layer is the encoded representation.

Autoencoder (AE)

As shown in Figure 2.5a, an autoencoder is a neural network that has three layers: the input layer, the hidden layer, and the output layer. It differs from the standard neural network, in that the AE is trained to reconstruct its inputs, which forces the hidden layer to try to learn good representations of the inputs.

The structure of AE contains two blocks. The first block is called the encoder, which embeds the observation to a latent representation (also called 'code'),

$$\boldsymbol{x^h} = \sigma(\mathcal{T}(\boldsymbol{x})) \tag{2.19}$$

where $\boldsymbol{x^h}$ represents the hidden layer. The second block is called the decoder, which decodes the representation into the original space,

$$\boldsymbol{y'} = \sigma(\mathcal{T}(\boldsymbol{x^h})) \tag{2.20}$$

where $\boldsymbol{y'}$ represents the output.

AE forces $\boldsymbol{y'}$ to be equal to the input $\boldsymbol{x}$ and calculates the error based on the distance between them. Thus, AE can compute the loss function only by $\boldsymbol{x}$ without the ground truth $\boldsymbol{y}$

$$error = \|\boldsymbol{y'} - \boldsymbol{x}\|_2 \tag{2.21}$$

Compared to Equation 2.6, this equation does not involve the variable $\boldsymbol{y}$ because it takes the input $\boldsymbol{x}$ as the ground truth. This is the reason why AE is able to perform unsupervised learning.

Naturally, one variant of AE is Deep-AE (D-AE) which has more than one hidden layer. We present the structure of D-AE with three hidden layers in Figure 2.5c. From the figure, we can observe that there is one more hidden layer in both the encoder and the decoder. The symmetrical structure ensures the smoothness of encoding and decoding procedure. Thus, D-AE generally has an odd number of hidden layers (e.g., $2n + 1$) where the first n layers belong to the encoder, the $(n + 1)$-th layer works as the code which belongs to both encoder and decoder, and

the last n layers belong to the decoder. The data flow of D-AE (Figure 2.5c) can be represented as

$$\boldsymbol{x^{h1}} = \sigma(\mathcal{T}(\boldsymbol{x})) \tag{2.22}$$

$$\boldsymbol{x^{h2}} = \sigma(\mathcal{T}(\boldsymbol{x^{h2}})) \tag{2.23}$$

where $\boldsymbol{x^{h2}}$ denotes the median hidden layer (the code). Then decode the hidden layer, we can get

$$\boldsymbol{x^{h3}} = \sigma(\mathcal{T}(\boldsymbol{x^{h2}})) \tag{2.24}$$

$$\boldsymbol{y'} = \sigma(\mathcal{T}(\boldsymbol{x^{h3}})) \tag{2.25}$$

It is almost the same as AE except that D-AE has more hidden layers. Apart from D-AE, AE has many other variants like denoising autoencoder, sparse autoencoder, contractive AE, etc. Here we only introduce the D-AE because it is easily confused with the AE-based deep belief network. The key difference between them will be provided in Section 2.1.2.

The core idea of AE and its variants is simple, which is that condensing the input data $\boldsymbol{x}$ into a code $\boldsymbol{x^h}$ (generally the code layer has lower dimension) and then reconstructing the data based on the code. If the reconstructed $\boldsymbol{y'}$ can approximate to the input data $\boldsymbol{x}$, it can be demonstrated that the condensed code $\boldsymbol{x^h}$ carries enough information about $\boldsymbol{x}$, thus, we can regard $\boldsymbol{x^h}$ as a representation of the input data for future operation (e.g., classification).

Restricted Boltzmann Machine (RBM)

Restricted Boltzmann Machine is a stochastic artificial neural network that can learn a probability distribution over its set of inputs. It contains two layers including one visible layer (input layer) and one hidden layer, as shown in Figure 2.5b. From the figure, we can see that the connection lines between the two layers are bidirectional. RBM is a variant of Boltzmann Machine with stronger restriction of being without intra-layer connections[6]. Similar to AE, the procedure of RBM also includes two

[6]In a general Boltzmann machine, the nodes in the same hidden layer will connect.

steps. The first step condenses the input data from the original space to the hidden layer in a latent space. After that, the hidden layer is used to reconstruct the input data in an identical way. Compared to AE, RBM has a stronger constraint which is that the encoder weights and the decoder weights should be equal. We have

$$\boldsymbol{x^h} = \sigma(\mathcal{T}(\boldsymbol{x})) \tag{2.26}$$

$$\boldsymbol{x'} = \sigma(\mathcal{T}(\boldsymbol{x^h})) \tag{2.27}$$

In the above two equations, the weights of $\mathcal{T}(\cdot)$ are the same. Then, the error for training can be calculated by

$$error = \|\boldsymbol{x'} - \boldsymbol{x}\|_2 \tag{2.28}$$

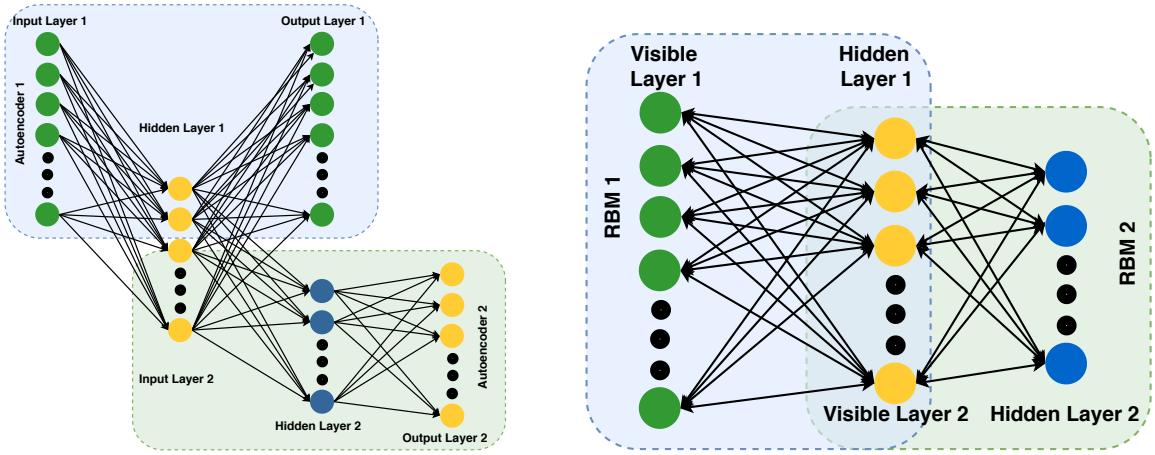
We can observe from the Figure 2.5d that the Deep-RBM (D-RBM) is an RBM with multiple hidden layers. The input data from the visible layer firstly flow to the first hidden layer and then the second hidden layer. Then, the code will flow backward into the visible layer for reconstruction.

Deep Belief Networks (DBN)

A Deep Belief Network (DBN) is a stack of simple networks, such as AEs or RBMs [53]. Thus, we divided DBN into DBN-AE (also called stacked AE) which is composed of AE and DBN-RBM (also called stacked RBM) which is composed of RBM.

As shown in Figure 2.6a, the DBN-AE contains two AE structures while the hidden layer of the first AE works as the input layer of the second AE. This diagram has two stages. In the first stage, the input data feed into the first AE follows the rules introduced in Section 2.1.2. The reconstruction error is calculated and back propagated to adjust the corresponding weights and basis. This iteration continues until the AE converges. We get the mapping,

$$\boldsymbol{x^1} \rightarrow \boldsymbol{x^{h1}} \tag{2.29}$$



(a) DBN-AE (b) DBN-RBM

Figure 2.6: Illustration of deep belief networks. (a) DBN composed of autoencoders. DBN-AE contains multiple AE components (in this case, two AE), with the hidden layer of the previous AE working as the input layer of the next AE. The hidden layer of the last AE is the learned representation. (b) DBN composed of RBM. The hidden layer of the first RBM working as the visible layer of the second RBM. The last hidden layer is the encoded representation. While DBN-RBM and D-RBM (Figure 2.5d) have similar architecture, the former is trained greedily while the latter is trained jointly .

Then, we move on to the second stage where the learned representative code in the hidden layer $\boldsymbol{x^{h1}}$ will be used as the input layer of the second AE, which is

$$\boldsymbol{x^2} = \boldsymbol{x^{h1}} \tag{2.30}$$

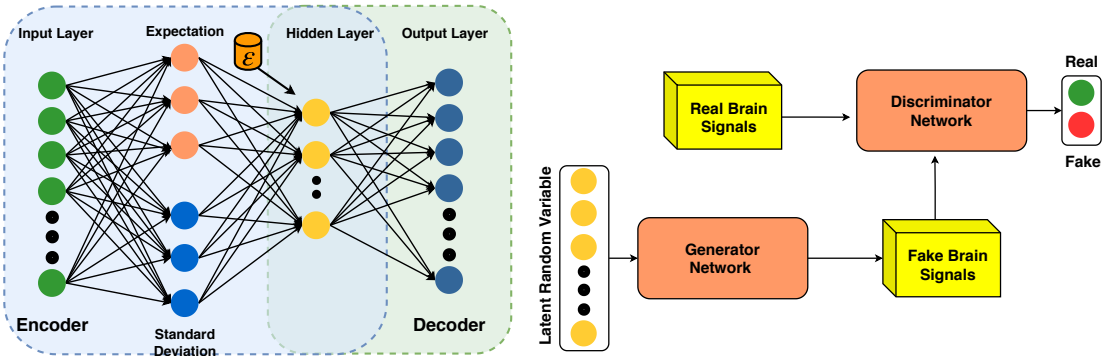
and then, after the second AE converges, we have

$$\boldsymbol{x^2} \rightarrow \boldsymbol{x^{h2}} \tag{2.31}$$

where $\boldsymbol{x^{h2}}$ denotes the hidden layer of the second AE, meanwhile, it is the final outcome of the DBN-AE.

The core idea of AE is that of learning a representative code with lower dimensionality but containing most information of the input data. The idea behind DBN-AE is to learn a more representative and purer code.

Similarly, the DBN-RBM is composed of several single RBM structures. Figure 2.6b



(a) Variational Autoencoder (b) Generative Adversarial Networks

Figure 2.7: Illustration of generative deep learning models. (a) VAE contains two hidden layers. The first hidden layer is composed of two components: the expectation and the standard deviation, which are learned separately from the input layer. The second hidden layer represents the encoded information. ϵ denotes the standard normal distribution. (b) GAN mainly contain two crucial components: the generator and the discriminator network. The former receives a latent random variable to generate a fake brain signal while the latter receives both the real and the generated brain signals and attempts to determine if its generated or not. In BCI, GAN reconstructs or augments data instead of classification.

shows a DBN with two RBMs where the hidden layer of the first RBM is used as the visible layer of the second RBM.

Compare the DBN-RBM (Figure 2.6b) and D-RBM (Figure 2.5d). They almost have the same architecture. Moreover, DBN-AE (Figure 2.6a) and D-AE (Figure 2.5c) have similar architecture. The most important difference between the DBN and the deep AE/RBM is that the former is trained greedily while the latter is trained jointly. In particular, for the DBN, the first AE/RBM is trained first, after it converges, the second AE/RBM is trained[54]. For the deep AE/RBM, jointly training means that the whole structure is trained together, no matter how layers it has.

2.1.3 Generative Deep Learning Models

Generative deep learning models are mainly used to generate training samples or data augmentation. In other words, generative deep learning models play a support-

ing role in the BCI field to enhance the training data quality and quantity. After the data augmentation, the discriminative models will be employed for the classification. This procedure is created to improve the robustness and effectiveness of the trained deep learning networks, especially when the training data is limited. In short, the generative models receive the input data and output a batch of similar data. In this section, we will introduce two typical generative deep learning models: VAE and GAN.

Variational Autoencoder (VAE)

Variational Autoencoder, proposed in 2013 [55], is an important variant of AE, and one of the most powerful generative algorithms. The standard AE and its other variants can be used for representation but fail in generation for the reason that the learned code (or representation) may not be continuous. Therefore, we cannot generate a random sample which is similar to the input sample. In other words, the standard AE does not allow interpolation. Thus, we can replicate the input sample but cannot generate a similar one. VAE has one fundamentally unique property that separates it from other AEs, and it is this property that makes VAE so useful for generative modeling: the latent spaces are designed to be continuous which allows easy random sampling and interpolation. Next, we will introduce how VAE works.

Similar to the standard AE, VAE can be divided into an encoder and decoder where the former embeds the input data to a latent space and the latter transfers the data from the latent space to the original space. However, the learned representation in the latent space is forced to approximate a prior distribution $\boldsymbol{p}(\bar{\boldsymbol{z}})$ which is generally set as Standard Gaussian distribution. Based on the re-parameterization trick [55], the first hidden layer of VAE is designed to have two parts where one denotes the expectation $\boldsymbol{\mu}$ and another denotes the standard deviation $\boldsymbol{\sigma}$, thus we have

$$\boldsymbol{\mu} = \sigma(\mathcal{T}(\boldsymbol{x})) \tag{2.32}$$

$$\boldsymbol{\sigma} = \sigma(\mathcal{T}(\boldsymbol{x})) \tag{2.33}$$

Then, the latent code in the hidden layer is not directly calculated but sampled from a Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)$. The statistic code

$$\boldsymbol{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} * \boldsymbol{\varepsilon} \tag{2.34}$$

where $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$. The representation $\boldsymbol{z}$ is forced to a prior distribution, and the distance $error_{KL}$ is measured by KullbackâĂŞLeibler divergence,

$$error_{KL} = D_{KL}(z, \boldsymbol{p}(\bar{\boldsymbol{z}})) \tag{2.35}$$

where $\boldsymbol{p}(\bar{\boldsymbol{z}})$ denotes the prior distribution. In the decoder, $\boldsymbol{z}$ is decoded into the output $\boldsymbol{y}'$,

$$\boldsymbol{y'} = \sigma(\mathcal{T}(\boldsymbol{z})) \tag{2.36}$$

and the reconstruction error is

$$error_{recon} = \|\boldsymbol{y'} - \boldsymbol{x}\|_2 \tag{2.37}$$

The overall error for VAE is combined by the DL divergence and the reconstruction error,

$$error = error_{KL} + error_{recon} \tag{2.38}$$

The key point of VAE is that all the latent representations $\boldsymbol{z}$ are forced to obey the normal distribution. Thus, we can randomly sample a representation $\boldsymbol{z'} \in \boldsymbol{p}(\bar{\boldsymbol{z}})$ from the prior distribution and then reconstruct a sample based on $\boldsymbol{z'}$. This is why VAE is so powerful in generation.

Generative Adversarial Networks (GAN)

Generative Adversarial Networks [56] is proposed in 2014 and achieved great success in a wide range of research areas (e.g., computer vision and natural language processing). GAN is composed of two simultaneously trained neural networks with a generator and a discriminator. The generator captures the distribution of the input data, and the discriminator is used to estimate the probability that a sample

came from the training data. The generator aims to generate fake samples while the discriminator aims to distinguish whether the sample is genuine. The functions of the generator and the discriminator are opposite; that's why GAN is called 'adversarial.' After the convergence of both the generator and the discriminator, the discriminator ought to be unable to recognize the generated samples. Thus, the pre-trained generator can be used to create a batch of samples and use them for further operations such as as classification.

Figure 2.7b shows the procedure of a standard GAN. The generator receives a noise signal $\boldsymbol{s}$ which is randomly sampled from a multimodal Gaussian distribution and outputs the fake brain signals $\boldsymbol{x}_F$. The discriminator receives the real brain signals $\boldsymbol{x}_R$ and the generated fake sample $\boldsymbol{x}_F$, and then it predicts whether the received sample is real or fake. The internal architecture of the generator and discriminator are designed depending on the data types and scenarios. For instance, we can build the GAN by convolutional layers on fMRI images since CNN has an excellent ability to extract spatial features. The discriminator and the generator are trained jointly. After the convergence, numerous brain signals $\boldsymbol{x}_G$ can be created by the generator. Thus, the training set is enlarged from $\boldsymbol{x}_R$ to $\{\boldsymbol{x}_R, \boldsymbol{x}_G\}$ to train a more effective and robust classifier.

2.1.4 Hybrid Model

Hybrid deep learning models refers to models which are composed of at least two deep basic learning models where the basic model is a discriminative, representative, or generative deep learning model. Hybrid models comprise two subcategories based on their targets: classification-aimed (CA) hybrid models and the non-classification-aimed (NCA) hybrid models.

Most of the deep learning related studies in BCI are focused on the first category. Based on the existing literature, the representative and generative models are employed to enhance the discriminative models. The representative models can provide

more informative and low dimensional features for the discrimination while the generative models can help to augment the training data quality and quantity which supply more information for the classification. The CA hybrid models can be further subdivided into[7]: 1) several discriminative models combined to extract more distinctive and robust features (e.g., CNN+RNN); 2) a representative model followed by a discriminative model (e.g., DBN+MLP); 3) a generative model combined with a representative model followed by a discriminative model; 4) a generative model combined with a representative model followed by a non-deep learning classifier.

A few NCA hybrid models aim for brain signal reconstruction. For example, St-yves et al. [57] adopted GAN to reconstruct visual stimuli based on fMRI images.

2.2 State-of-The-Art DL Techniques for BCI

In this section, we will systematically summarize the existing state-of-the-art studies for BCI based on deep learning. Some literature combined deep learning and traditional machine learning methods are also listed.

More than half of the recent publications are related to EEG signals because this approach is non-invasive, high-portable and low-cost. In this section, we will summarize the state-of-the-art research based on three aspects: EEG oscillations, evoked potentials, and ERD/ERS.

2.2.1 EEG Oscillatory

Spontaneous EEG has a vast range of applications since it is well suited to a range of different scenarios. In particular, spontaneous EEG includes sleeping EEG, motor

[7]The representative model followed by a non-deep learning classifier is regarded as a representative deep learning model.

imagery EEG, emotional EEG, mental disease EEG, and others. Next, we will present the studies in each scenario and the deep learning models used.

Sleeping EEG

Sleep quality is significant for diagnosing sleep disorders and cultivating healthy habits. Sleep EEG is mainly used to recognize sleep stages (or sleep score/state) [58]. In Rechtschaffen and Kales (R&K) rules, the sleep stages include wakefulness, non-REM (rapid eye movement) 1, non-REM 2, non-REM 3, non-REM 4, and REM. However, there is no clear distinction between non-REM 3 and non-REM 4. Therefore, they are combined into slow wave sleep (SWS) [59]. The American Academy of Sleep Medicine (AASM) recommends segmentation of sleep in five stages: wakefulness, non-REM (rapid eye movement) 1, non-REM 2, SWS, and REM. Generally, in sleep stage analysis, the EEG signals are preprocessed by a filter which has various passband in different papers, but most of the studies notched at 50 Hz to remove power-line noise. The EEG signals are usually segmented into 30s windows.

(i) *Discriminative models.* Many publications have adopted CNN for sleep-stage classification on single-channel EEG [51, 60]. Viamala et al. [61] manually extracted time-frequency features from sleeping EEG signals and adopted a CNN algorithm to analyze them. The EEG signal collected from $Fpz - Cz$ and$Pz - Oz$ channels, was sliced into 30 s segments. The employed CNN achieved an accuracy of 86% in five-class classification. Shahin et al. [62] manually extract 57 features in the frequency domain and fed them into an MLP for classification, which obtained an accuracy of 90%in insomnia detection. Fernande et al. [63] adopted CNN to analyze physiological signals including EEG, EOG, and EMG. The model was evaluated over the Sleep Heart Health Scoring dataset and achieved a precision of 91%, recall of 90%, and F-1 score of 90%.

RNN is also often used in sleep disorder detection. Biswal et al. [64] demonstrated that RNN performed better than MLP, and CNN for sleep stage prediction. Tsiouris

et al. [65] extracted many features from the time domain, frequency domain, correlation, and graph theoretical features. An LSTM was employed to discover the latent dependencies of the features for seizure detection.

(ii) *Representative models.* Zhang et al. [59] combined a DBN-RBM with three RBMs for sleep feature extraction and traditional machine learning classifiers (e.g., SVM) for classification. Tan et al. [66] adopted a DBN-RBM algorithm to detect sleep spindles from the extracted PSD features of the sleeping EEG signals. They finally reached an F-1 measure of 92.78% in a local dataset.

(iii) *Hybrid models.* Manzano et al. [67, 68] proposed a multi-view model to predict sleep stage by combining CNN and MLP. The CNN was employed to receive the raw EEG data in the time domain while the MLP received the spectrum obtained by a Short-Time Fourier Transform (STFT) between 0.5-32 Hz. Supratak et al. [69] proposed a model by combining a multi-view CNN and LSTM for automatic sleep-stage scoring based on raw single-channel EEG. The proposed method utilized convolutional neural networks to extract time-invariant features, and bidirectional-long short-term memory to learn transition rules among sleep stages. Dong et al. [70] proposed a hybrid deep learning model aimed at temporal sleep stage classification. They have taken advantage of MLP for detecting hierarchical features and LSTM for sequential data learning to optimize classification performance with single-channel recordings.

MI EEG

Extreme Learning Machine (ELM) [71] Deep learning models have shown the superior on the classification of MI EEG and real-motor EEG [72, 73].

(i) *Discriminative models.* CNN is widely used for the recognition of MI EEG [74]. On the one hand, some studies CNN is only used as a classifier to recognize manually extracted features [75, 76]. Uktveris et al. [77] extracted a large number of

EEG features including Mean channel energy (MCE), Mean window energy (MWE), Channel variance (CV), Mean band power (BP), etc. All the extracted features were sent into a 2-D CNN for classification. Lee et al. [78] first processed the MI EEG signals through wavelet transformation and then manually extracted PSD from mu and beta bands. Finally, they employed a CNN model for recognition and achieved an accuracy of 78.93%. Apart from CNN, Zhang et al. [2] used a modified LSTM structure to learn affective information from EEG signals to control smart home appliances.

On the other hand, CNN deals with the raw EEG data based on feature engineering and classification results [79]. Wang et al. [80] designed a fast convolutional feature extraction approach based on CNN to learn the latent features from MI-EEG signals. Several weak classifiers are applied to choose important features for the final classification. Hartmann et al. [72] worked on the EEG signals corresponding to real motor action. They investigated how the CNN represented spectral features through the sequence of intermediate stages of the network, which showed higher sensitivity to EEG phase features at earlier stages and higher sensitivity to EEG amplitude features at later stages. Moreover, MLP is also applied for MI EEG recognition [81].

(ii) *Representative models.* DBN is widely employed for MI EEG classification because of its high representative ability [82, 83]. Ren et al. [84] applied a convolutional DBN based on RBM components. They claimed that the DBN worked better in feature representation than traditional hand-crafted features (e.g., CSP, band powers). Li et al. [85] processed EEG signals with discrete wavelet transformation and then applied a DBN-AE based on denoising AE. They achieved an accuracy of 73.86% over a local MI EEG dataset. The authors also used denoising AE to generate the missing values in incomplete EEG signals such as an EEG segment with a portion of data removed (unevenly spaced). Rekar et al. [86] employed an AE model for feature extraction followed by a KNN classifier, which achieved an accuracy of 72.38% in binary classification over a local dataset.

Nurse et al. [87] proposed a model combining MLP with Genetic Algorithm (GA)

where the GA was used for optimal hyper-parameter selection (e.g., the number of hidden layers in MLP) and the MLP worked as the classifier. Zhang et al. [3] combined AE with an XGBoost classifier to recognize the EEG signals in a multi-person scenario. The authors also proposed a complex framework by combining LSTM with reinforcement learning to classify multi-modality signals [4, 88].

(iii) *Hybrid models.* Several studies proposed hybrid models for the recognition of MI EEG [89]. Fraiwan et al. [90] combined DBN with MLP for neonatal sleep state identification. Twelve features were extracted from the time and frequency domain of the sleeping EEG signals, which were refined by a designed DBN-AE. After that, the MLP classifier gave an accuracy of 80.4% on a public dataset. Tabar et al. [91] combined the time, frequency and location information of the EEG signals as the input data which would be fed into a CNN for high-level feature extraction. The features were classified through a DBN-AE with seven AEs while the hidden layer of AE only had two nodes which corresponded to the probability of the two labels. Tan et al. [92] proposed a complicated system to achieve multimodal EEG classification. A denoising AE was employed for dimensional reduction. A multi-view CNN combined with RNN was proposed to discover the latent temporal and spatial information from the low-dimension representations. They obtained an average accuracy of 72.22% over the IIa dataset from BCI competition IV.

Emotional EEG

The emotion of an individual can be evaluated by three aspects: the valence, arousal, and dominance. Each aspect can be rated by an integer between 1 to 9 or can be divided into positive and negative. The combination of the three aspects forms the emotions which are familiar to us like fear, sadness, anger. The subject's EEG signals could be used to predict the affective state.

(i) *Discriminative models.* In the beginning, the basic MLP is adopted to classify manually extracted features when deep learning first arose [93]. Frydenlund et al.

[94] extracted the average and standard deviation of each EEG band and then fed them into an MLP for emotional affect estimation.

However, CNN is the most popular in the area of EEG based emotion prediction [95, 96]. Li et al. [95] proposed a hierarchical CNN to implement the EEG-based emotion classifier (positive, negative and neutral) in a movie-watching task. Differential Entropy (DE) is calculated as the main feature. This paper first proposes that converting multi-channel EEG signals into a 2-D matrix, which takes advantage of the spatial dependencies among EEG channels. For the emotion recognition task, this paper compared the proposed CNN with a DBN-AE and demonstrated that CNN has better performance than DBN, which is similar to [97]. Wang et al. [98] employed a CNN algorithm to classify emotional EEG signals. Of note is the fact that they augmented the training set by generating new EEG samples by adding Gaussian noise to the original samples. Li et al. [99] proposed a novel hierarchical convolutional neural network (HCNN) to recognize the subject's emotional state (positive, neutral, and negative) and obtained an accuracy of 88.2%. In the HCNN structure, each convolutional kernel only has localized receptive field, so the kernels can capture the correlations among adjacent electrodes, which might be of great value for the recognition task.

RNN and its variants are another group of widely used discriminative models. Talathi [100] utilized a discriminative deep learning model composed of GRU cells to detect early seizure disease and achieved competitive performance. Zhang et al. [101] proposed a spatial-temporal recurrent neural network (STRNN) to integrate the feature learning from both spatial and temporal information. To capture those spatially co-occurrent variations of human emotions, a multi-directional RNN layer can capture long-range contextual cues by traversing the spatial regions of each temporal slice along with different directions. Then, a bi-directional temporal RNN layer is further used to learn the discriminative features characterizing the temporal dependencies of the sequences produced by the spatial RNN layer.

(ii) *Representative models.* DBN, especially DBN-RBM, is widely used for unsuper-

vised representation ability in emotion recognition [102, 103, 104]. For instance, Xu et al. [105] proposed a DBN-RBM algorithm with three RBMs and an RBM-AE to predict the subject's affective state. Nevertheless, it is not a strictly semi-supervised method: the model reported by [105] is composed of unsupervised feature representation and a supervised softmax layer. The authors also tried to manually extract the PSD features from 14 narrow-down bands of the EEG signals and then fed them into DBN-RBM for classification [106]. For Alzheimer's Disease diagnosis, Zhao et al. [107] adopted DBN-RBM with three RBMs to extract informative representations after filtering ($0.5 \sim 30$ Hz). The proposed representative model is combined with a traditional classifier (SVM) and achieved an accuracy of 92%. Another work combined DBN-RBM with Hidden Markov Model (HMM) and achieved an accuracy of 87.62% in a local dataset [108].

Compared to other repetitive models, D-RBM only appears in a few studies. Zheng et al. [109, 110] introduced a D-RBM with five hidden RBM layers to investigate critical frequency bands and channels in emotion recognition. The authors claimed that they employed a DBN-RBM; however, the RBMs are trained jointly. Thus it is regarded as D-RBM in this chapter. Jia et al. [111] proposed an interesting algorithm which is composed of RBMs. The algorithm contains a channel selection component and an RBM classifier. The data from each EEG channel are reconstructed through RBM; then, the channels with high error are eliminated. Then the representative features of the residual channels are sent to D-RBM for affective state recognition.

Emotion is affected by many subjective and environmental factors, such as gender, fatigue, etc. Yan et al. [112, 113] investigated the differences between males and females in emotion recognition using EEG and eye movement data. They proposed a novel model called Bimodal Deep AutoEncoder (BDAE) which is, however, actually formed by RBMs. The BDAE received both EEG and eye movement features and shared the information in a fusion layer which connected with an SVM classifier. The results showed that the fearful emotion is more diverse among women compared with

men, and men behave more diversely on the sad emotion compared with women. Moreover, individual differences in fear are more pronounced than in the other three emotions for females.

To overcome the mismatched distribution among the samples collected from different subjects or different experimental sessions, Chai et al. [114] proposed an unsupervised domain adaptation technology which is called the subspace alignment autoencoder (SAAE). SAAE combined an AE and a subspace alignment solution, which could take advantage of both nonlinear transformation and a consistency constraint. The proposed approach obtained a mean accuracy of 77.88% in a person-independent scenario.

(iii) *Hybrid models.* One commonly-used hybrid model is a combination of RNN and MLP. For example, Alhagry et al. [115] employed an LSTM architecture for feature extraction from emotional EEG signals, and the features are forwarded into an MLP for classification, which got 85.65%, 85.45%, and 87.99% accuracy on arousal, valence, and liking classes, respectively. Furthermore, Yin et al. [116] proposed a multi-view ensemble classifier to recognize emotions using multimodal physiological signals. The ensemble classifier contains several D-AEs with three hidden layers and a fusion structure. Each D-AE receives one physiological signal (e.g., EEG, EOG, EMG) and then sends the outputs of D-AE to a fusion structure which is composed of another D-AE. At last, an MLP classifier classifies the mixed features. Kawde et al. [117] implemented an affect recognition system by combining a DBN-RBM for effective feature extraction and an MLP for classification.

Mental Disease EEG

A large number of researchers exploited EEG signals to diagnose neurological disorders, especially epileptic seizures [118].

(i) *Discriminative models.* CNN is widely used in the automatic detection of epileptic

seizures [119, 120, 121, 122]. For example, Johansen et al. [123] adopted CNN to work on the high-passed filtered (>1 Hz) EEG signals of epileptic spikes and achieved an AUC of 94.7%. Acharya et al. [124] employed a CNN model with 13 layers (5 convolutional layers, five pooling layers, and three fully-connected layers) on depression detection. The method was evaluated on a local dataset with 30 subjects (15 normal and 15 depressed) and achieved the accuracies of 93.5% and 96.0% using EEG signals from the left and right hemisphere, respectively. Morabito et al. [125] exploited a CNN structure to extract suitable features of multi-channel EEG signals to classify Alzheimer's Disease from a prodromal version of dementia (Mild Cognitive Impairment, MCI) and age-matched Healthy Controls (HC). The EEG signals are filtered in bandpass ($0.1 \sim 30$ Hz) and finally achieved an accuracy of around 82% for three-class classification.

In some research, the discriminative model is only employed for feature extraction. For example, Ansari et al. [126] used CNN to extract the latent features which are fed into a Random Forest classifier for the final seizure detection in neonatal babies. Chu et al. [127] employed CNN for feature extraction which was sent to a random forest for schizophrenia recognition.

REM Behavior Disorder (RBD) may cause many mental disorder diseases like Parkinson's disease (PD). Ruffini et al. [128] described an Echo State Networks (ESNs) model to distinguish RBD from healthy individuals. ESN, as a particular class of RNN, implements nonlinear dynamics with memory and seem ideally poised for the classification of complex time series data. The central concept in ESNs and related types of so-called âĂIJreservoir computationâĂİ systems is to have data inputs drive a semi-randomly connected, large, fixed recurrent neural network (the âĂIJreservoirâĂİ) where each node/neuron in the reservoir is activated in a nonlinear way.

(ii) *Representative models.* For disease detection, one commonly used method is adopting a representative model (e.g., DBN) followed by a softmax layer for classification [129, 130]. Page et al. [131] adopted DBN-AE to extract useful features

from seizure EEG signals. The extracted features were fed into a traditional logistic regression classifier for seizure detection. Al et al. [132] proposed a multi-view DBN-RBM structure to analyze EEG signals from depressed patients. The proposed approach contains multiple input pathways, composed of two RBMs, while each corresponded to one EEG channel. All the input pathways would merge into a shared structure which is composed of another RBMs. The results showed that the multi-view DBN-RBM achieved competitive results. Yuan et al. [133] extract EEG context features in parallel by using global principal component analysis (GPCA), deep denoising AE, and EEG embeddings, respectively. The multi-features are concatenated into a fixed-length feature vector for seizure classification.

Some papers favor preprocessing the EEG signals through dimensionality reduction methods such as PCA and ICA [134] while others prefer to direct fed the raw signals to the representative model [135]. Lin et al. [135] proposed a sparse D-AE with three hidden layers to extract the representative features from epileptic EEG signals while Hosseini et al. [134] adopted a similar sparse D-AE with two hidden layers.

(iii) *Hybrid models.* A popular hybrid method is a combination of RNN and CNN. Shah et al. [136] investigated the performance of CNN-LSTM on seizure detection after channel selection. They used a reduced number of channels ranging from 8 to 20, and achieved sensitivities between 33% and 37% with false alarms in the range of 38% and 50%. Golmohammadi et al. [137] proposed a hybrid architecture for automatic interpretation of EEG that integrates temporal and spatial context for sequential decoding of EEG events. 2D and 1D CNNs capture the spacial features while LSTM networks capture the temporal features. The authors claimed sensitivity of 30.83% and a specificity of 96.86% on the well-known TUH EEG seizure corpus.

In the detection of early-stage Creutzfeldt-Jakob Disease (SJD), Morabito et al. [138] combined D-AE and MLP together. The EEG signals of SJD were first filtered by bandpass (0.5$\sim$70 Hz) and then fed into a D-AE with two hidden layers for feature representation. At last, the MLP classifier obtained the accuracy of 81$\sim$ 83% in a

local dataset. Convolutional autoencoder, replacing the fully-connected layers in a standard AE by convolutional and de-convolutional layers, is applied to extract the seizure features in an unsupervised manner [139].

(5) **Data augmentation.** Generative models such as GAN can be used for data augmentation in BCI classification [140]. Palazzo et al. [141] first demonstrated that brain activity EEG signals encode visually-related information that enables to discriminate between visual object categories accurately. Then, they extracted a more compact class-dependent representation of EEG data using recurrent neural networks. At last, they used the learned EEG manifold to condition image generation employing GANs, which, during inference, will read EEG signals and convert them into images. Kavasidis et al. [142] aiming at converting EEG signals into images. The EEG signals were collected when the subjects were observing images on a screen. An LSTM layer was employed to extract the latent features from the EEG signals, and the extracted features were regarded as the input of a GAN structure. The generator and the discriminator of the GAN were both composed of convolutional layers. The generator was supposed to generate an image based on the input EEG signals after the pre-training. Abdelfattach et al. [140] adopted a GAN on seizure data augmentation. The generator and discriminator are both composed of fully-connected layers. The authors demonstrated that GAN outperforms AE and VAE. After the augmentation, the classification accuracy increased dramatically from 48% to 82%.

(6) **Others** Other researchers have explored a wide range of interesting topics. The first one is how EEG affected by audio/visual stimuli. This differs from the potentials evoked by audio/visual stimulations because the stimuli in this phenomenon are constant instead of fluctuating at a particular frequency. Stober et al. [143, 144] claimed that EEG signals of rhythm perception might contain enough information to distinguish different rhythm types/genres or even identify the rhythms themselves. The authors conducted an experiment where 13 participants were stimulated by 23 rhythmic stimuli including 12 East African and 12 Western stimuli. For the

24-category classification, the proposed CNN achieved a mean accuracy of 24.4%. After that, the authors exploited convolutional AE for feature learning and CNN for classification and achieved an accuracy of 27% for 12-class classification [145]. Sternin et al. [146] adopted CNN to extract discriminative features from the EEG signals to distinguish whether the subject was listening or imaging music. Similarly, Sarkar et al. [147] designed two deep learning models to recognize the EEG signals invoked by audio or visual stimuli. For this binary classification task, the proposed CNN and DBN-RBM with three RBMs achieved the accuracy of 91.63% and 91.75%, respectively. Furthermore, the spontaneous EEG could be used to distinguish the user's mental state (logical versus emotional) [148].

Moreover, some researchers focus on the impact of cognitive load [149] or physical workload [150] on EEG . Bashivan et al. [151] first extracted informative features through wavelet entropy and band-specific power which were fed into a DBN-RBM for further refining. At last, an MLP is employed for cognitive load level recognition. The authors, in another work [152], also aimed to find representations that are invariant to inter- and intra-subject differences from multi-channel EEG time-series in the context of the mental load classification task. They transformed EEG activities into a sequence of topology-preserving multi-spectral images and then trained a recurrent-convolutional network to preserve the spatial, spectral, and temporal features of the EEG signals. Yin et al. [153] collected the EEG signals from different mental workload levels (e.g., high and low) for binary classification. The EEG signals were filtered by a low-pass filter, transformed to the frequency domain and the power spectral density (PSD) was calculated. The extracted PSD features were fed into a denoising D-AE structure for future refining. They finally achieved an accuracy of 95.48%. Li et al. [154] worked on the recognition of mental fatigue level including alert, slight fatigue, and severe fatigue. They adopted a simple DBN-RBM to extract the related features from single-channel EEG.

In addition, EEG based driver fatigue detection is a popular area of research[155, 156, 157, 157, 158]. Huang et al. [159] designed a 3D CNN to predict reaction

time in drowsy driving. This is useful to reduce traffic accidents. Hajinoroozi et al. [160] adopted a DBN-RBM to handle the EEG signals which were processed by ICA. They achieved an accuracy of around 85% in binary classification ('drowsy' or 'alert'). The strength of this paper is that they evaluated the DBN-RBM on three levels: time samples, channel epochs, and windowed samples. The experiments showed that the channel epoch level provided the best performance. San et al. [161] combined deep learning models with a traditional classifier to detect driver fatigue. The model contains a DBN-RBM structure followed by an SVM classifier, which achieved a detection accuracy of 73.29%. Almogbel et al. [162] investigated the drivers' mental state under high workload and low workload. A proposed CNN is claimed to detect the driver's cognitive workload directly based on the raw EEG signals.

Research into detection of eye state has shown exceedingly high accuracy. Narejo et al. [163] explored the detection of eye state (closed or open) based on EEG signals. They tried a DBN-RBM with three RBMs and a DBN-AE with three AEs and achieved a very high accuracy of 98.9%. Reddy et al. [164] tried a simpler structure, MLP, for eye state detection and got a slightly lower accuracy of 97.5%.

There are still a lot of promising areas that have not drawn much attention to date. Baltatzis et al. [165] adopted CNN to detect school bullying through EEG when watching the specific video. They achieved 93.7% and 88.58% for binary and four-class classification. Khurana et al. [166] proposed deep dictionary learning that outperformed several deep learning methods. Volker et al. [167] evaluated the use of Deep CNN in a flanker task, which achieved an averaging accuracy of 84.1% within subject and 81.7 on unseen subjects. Zhang et al. [10] combined CNN and graph network to discover the latent information from the EEG signal.

Miranda-Correa et al. [168] proposed a cascaded framework by combing RNN and CNN to predict individuals' affective level and personal factors (Big-five personality traits, mood, and social context). An experiment conducted by Putten et al. [169] attempted to identify the user's gender based on their EEG signals. They employed a

standard CNN algorithm and achieved the binary classification accuracy of 81% over a local dataset. The detection of emergency braking intention could help to reduce the responses time. Hernandez et al. [170] demonstrated that the driver's EEG signals could distinguish braking intention and normal driving state. They combined a CNN algorithm which achieved the accuracy of 71.8% in binary classification. Behncke et al. [171] applied deep learning, a CNN model, in the context of robot assistive devices. They attempted to use CNN to improve the accuracy of decoding robot errors from EEG while the subject watching the robot both during an object grasping and a pouring task.

Teo et al. [172, 173] tried to combine the BCI and recommender system, which predicted the user's preference by EEG signals. A cohort of 16 users was shown 60 bracelet-like objects as rotating visual stimuli (a 3D object) on a computer display while their preferences and EEGs were recorded. Then, an MLP algorithm was adopted to classify whether the user liked or disliked the object. This exploration got the prediction accuracy of 63.99%. Some researchers have tried to explore a common framework which can be used for various BCI paradigms. Lawhern et al. [174] introduced a compact CNN for EEG-based BCI. The authors described the use of depth-wise and separable convolutions to construct an EEG-specific model which encapsulates well-known EEG feature extraction concepts for BCI. The proposed EEGNet is evaluated on four BCI paradigms: P300 visual-evoked potentials, error-related negativity responses (ERN), movement-related cortical potentials (MRCP), and sensory-motor rhythms (SMR).

2.3 BCI Applications

Deep learning models have contributed to various BCI applications including health care, smart environments, security, affective computing, etc. In Table 2.3, we summarized deep learning based BCI paradigms. The papers focused on signal classification without a specific application are not listed in this table.

Table 2.3: Summary of deep learning based BCI applications. The 'local' dataset refers to private or not publicly available dataset and the public datasets (with links) will be introduced in Section 2.4. In the signals, S-EEG, MD EEG, and E-EEG separately denote sleeping EEG, mental disease EEG, and emotional EEG. The single 'EEG' refers to the other subcategory of spontaneous EEG. In the models, RF and LR denote to random forest and logistic regression algorithms, respectively. In the performance column, 'N/A', 'sen', 'spe', 'aro', 'val', 'dom', and 'like' denote not-found, sensitivity, specificity, arousal, valence, dominance, and liking, respectively.

BCI Applications		Reference	Signals	Deep Learning Models	Dataset	Performance
Health Care	Sleeping Quality Evaluation	Vilamala et al. [61]	S-EEG	CNN	Sleep-EDF	0.86
		Chambon et al. [58]	S-EEG	Multi-view CNN	MASS session 3	N/A
		Zhang et al. [59]	S-EEG	DBN + voting	UCD	0.9131
		Tsinalis et al. [51]	S-EEG	CNN	Sleep-EDF	0.82
		Sors et al. [60]	S-EEG	CNN	SHHS	0.87
		Manzano et al. [67]	S-EEG	CNN + MLP	Sleep-EDF	0.732
		Shahin et al. [62]	S-EEG	MLP	University Hospital in Berlin	0.9
		Manzano et al. [68]	S-EEG	CNN, MLP	Sleep-EDF	0.686/0.689
		Supratak et al. [69]	EEG	CNN + LSTM	MASS/ Sleep-EDF	0.862/0.82
		Ruffini et al. [128]	S-EEG	RNN	Local	0.85
		Fraiwan et al. [90]	S-EEG	DBN-AE + MLP	Local	0.804
		Tan et al. [66]	S-EEG	DBN-RBM	Local	0.9278 (F1)
		Fernandez et al. [63]	EEG	CNN	SHHS	0.9 (F1)
		Biswai et al. [64]	S-EEG	RNN	Local	0.8576
	AD Detection	Morabito et al. [125]	MD EEG	CNN	Local	0.82
		Zhao et al. [107]	MD EEG	DBN-RBM	Local	0.92
	Seizure Detection	Tsiouris et al. [65]	MD EEG	LSTM	CHB-MIT	>0.99
		Yuan et al. [118]	MD EEG	Attention-MLP	CHB-MIT	0.9661
		Yuan et al. [133]	MD EEG	D-AE + SVM	CHB-MIT	0.95
		Ullah et al. [119]	MD EEG	CNN + voting	UBD	0.954
		Lin et al.[135]	MD EEG	D-AE	UBD	0.96
		Hosseini et al. [134]	MD EEG	D-AE + MLP	Local	0.94
		Page et al. [131]	MD EEG	DBN-AE + LR	N/A	$0.8 \sim 0.9$
		Golmohammadi et al. [137]	MD EEG	RNN+CNN	TUH	Sen: 0.3083; Spe: 0.9686
		Wen et al. [139]	MD EEG	AE	Local	0.92
		Acharya et al. [120]	MD EEG	CNN	UBD	0.8867
		Schirmeister et al. [121]	MD EEG	CNN	TUH	0.854
		Hosseini et al. [175]	MD EEG	CNN	Local	N/A
		Talathi et al. [100]	MD EEG	GRU	BUD	0.996
		Johansen et al. [123]	MD EEG	CNN	Local	0.947 (AUC)
		Ansari et al. [126]	MD EEG	CNN + RF	Local	0.77

Table 2.3: Summary of deep learning based BCI applications (Continued).

BCI Applications		Reference	Signals	Deep Learning Models	Dataset	Performance
Health Care	Seizure Detection	Hosseini et al. [97]	EEG	CNN	Local	0.96
		Shah et al. [136]	MD EEG	CNN+ LSTM	TUH	Sen: 0.39; Spe: 0.9037
		Turner et al. [130]	MD EEG	DBN-RBM + LR	Local	N/A
	Others:					
	Cardiac Detection	Garg [176]	MEG	CNN	Local	Sen: 0.85, Spe: 0.97
		Hasasneh et al. [177]	MEG	CNN + MLP	Local	0.944
	Depression	Acharya et al. [124]	MD EEG	CNN	Local	0.935 ∼ 0.9596
		Al et al. [132]	MD EEG	DBN-RBM + MLP	Local	0.695
		Antoniades et al. [178]	EEG	AE + CNN	Local	0.68
	Schizophrenia	Chu et al. [127]		CNN + RF + Voting	Local	0.816, 0.967, 0.992
	Creutzfeldt-Jakob Disease (CJD)	Morabito et al. [138]	MD EEG	D-AE	Local	0.81 ∼ 0.83
Smart Environment	Robot Control	Behncke et al. [171]	EEG	CNN	Local	0.75
	Exoskeleton	Kwak et al. [179]	SSVEP	CNN	Local	0.9403
	Smart Home	Zhang et al. [2]	MI EEG	RNN	EEGMMI	0.9553
Brain Communication		Kawasaki et al. [180]	VEP	MLP	Local	0.908
		Cecotti et al. [181]	VEP	CNN + Voting	The third BCI competition, Dataset II	0.955
		Zhang et al. [9]	MI EEG	LSTM+CNN +AE	Local	0.9452
		Cecotti et al. [181]	VEP	CNN	The third BCI competition, Dataset II	0.945
		Maddula et al. [182]	VEP	RCNN	Local	0.65∼0.76
		Liu et al. [183]	VEP	CNN	The third BCI competition, Dataset II	0.92 ∼ 0.96
Security	Identification	Zhang et al. [8]	MI-EEG	Attention-based RNN	EEGMMI + local	0.9882
		Koike et al. [184]	VEP	MLP	Local	0.976
		Mao et al. [185]	RSVP	CNN	Local	0.97
	Authentication	Zhang et al. [74]	MI EEG	Hybrid	EEGMMI + local	0.984
Affective Computing		Mioranda et al. [168]	E-EEG	RNN + CNN	AMIGOS	<0.7
		Jia et al. [111]	E-EEG	DBN-RBM	DEAP	0.8 ∼ 0.85 (AUC)
		Li et al. [95]	E-EEG	Hierarchical CNN	SEED	0.882
		Xu et al. [105]	E-EEG	DBN-AE, DBN-RBM	DEAP	>0.86 (F1)
		Liu et al. [96]	E-EEG	CNN	Local	0.82
		Frydenlund et al. [94]	E-EEG	MLP	DEAP	N/A
		Yin et al. [116]	E-EEG	Multi-view D-AE + MLP	DEAP	Aro: 0.7719; Val: 0.7617
		Chai et al. [114]	E-EEG	AE	SEED	0.818
		Kawde et al. [117]	EEG	DBN-RBM	DEAP	Aro: 0.7033; Val: 0.7828; Dom: 0.7016
		Li et al. [102]	E-EEG	DBN-RBM	DEAP	Aro:0.642, Val:0.584, Dom 0.658

Table 2.3: Summary of deep learning based BCI applications (Continued).

BCI Applications		Reference	Signals	Deep Learning Models	Dataset	Performance
Affective Computing		Xu et al. [106]	E-EEG	DBN-RBM	DEAP	Aro:0.6984, Val:0.6688, Lik: 0.7539
		Zheng et al. [108]	E-EEG	DBN-RBM + HMM	Local	0.8762
		Alhagry et al. [115]	E-EEG	LSTM + MLP	DEAP	Aro:0.8565, Val:0.8545, Lik: 0.8799
		Li et al. [186]	E-EEG	CNN	SEED	0.882
		Zhang et al. [109, 110]	E-EEG	DBN-RBM + MLP	SEED	0.8608
		Liu et al. [113]	EEG	AE	SEED, DEAP	0.9101, 0.8325
		Gao et al. [103]	E-EEG	DBN-RBM + MLP	Local	0.684
		Zhang et al. [101]	E-EEG	RNN	SEED	0.895
Drive Fatigue Detection		Hung et al. [159, 159]	EEG	CNN	Local	0.572 (RMSE)
		Hajinoroozi et al. [160]	EEG	DBN-RBM	Local	0.85
		Hung et al. [159]	EEG	CNN	Local	
		Du et al. [156]	EEG	D-AE + SVM	Local	0.094 (RMSE)
		San et al. [161]	EEG	DBN-RBM + SVM	Local	0.7392
		Almogbel et al. [162]	EEG	CNN	Local	0.9531
		Hachem et al. [187]	SSVEP	MLP	Local	0.75
		Chai et al. [155]	EEG	DBN + MLP	Local	0.931
		Hajinoroozi et al. [157, 157]	EEG	CNN	Local	0.8294
Mental Load Measurement		Yin et al. [153]	EEG	D-AE	Local	0.9584
		Bashivan et al. [152]	EEG	R-CNN	Local	0.9111
		Bashivan et al. [148]	EEG	DBN + MLP	Local	N/A
		Bashivan et al. [151]	EEG	DBN-RBM	Local	0.92
		Li et al. [154]	EEG	DBN-RBM	Local	0.9886
Other Appli--cations	**School Bullying**	Baltatzis et al. [165]	EEG	CNN	Local	0.937
		Stober et al. [144]	EEG	CNN	Local	0.776
	Music Detection	Stober et al. [145]	EEG	AE + CNN	Open MIIR	0.27 for 12-class
		Stober et al. [143]	EEG	CNN	Local	0.244
	Number Choosing	Waytowich et al. [188]	SSVEP	CNN	Local	0.8
	Visual Object Recognition	Manor et al. [189]	RSVP	CNN	Local	0.75
		Cecotti et al. [190]	RSVP	CNN	Local	0.897 (AUC)
		Hajinoroozi et al. [191]	RSVP	CNN	Local	0.7242 (AUC)
		Perez et al. [192]	SSVEP	AE	Local	0.9778
		Shamwell et al. [193]	RSVP	CNN	Local	0.7252 (AUC)
	Guilty Knowledge Test	Kulasingham et al. [194]	SSVEP	DBN-RBM; DBN-AE	Local	0.869; 0.8601
	Concealed Information Test	Liu et al. [195]	EEG	DBN-RBM	Local	0.973
	Flanker Task	Volker et al. [167]	EEG	CNN	Local	0.841
	Eye State	Narejo et al. [163]	EEG	DBN-RBM	UCI	0.989
		Reddy et al. [164]	EEG	MLP	Local	0.975
	User Preference	Teo et al. [172]	EEG	MLP	Local	0.6399
	Emergency Braking	Hernandez et al. [170]	EEG	CNN	Local	0.718
	Gender Detection	Putten et al. [169]	EEG	CNN	Local	0.81

2.3.1 Health Care

In the health care area, deep learning based BCI systems mainly work on the detection and diagnosis of mental diseases such as sleeping disorders, Alzheimer's Disease, epileptic seizure, and other disorders. In the first place, for the sleeping disorder detection, most studies are focused on sleep-stage detection based on sleeping spontaneous EEG. In this situation, the researchers do not need to recruit patients with sleeping disorders because the sleeping EEG signals can be easily collected from healthy individuals. In terms of the algorithm, it can be observed from Table 2.3 that the DBN-RBM and CNN are widely adopted for feature engineering and classification. Ruffini et al. [128] went one step further by detecting REM Behavior Disorder (RBD) which may cause neuro-degenerative diseases such as Parkinson's disease. They achieved an average accuracy of 85% in recognition of the RBD from healthy controls.

Moreover, fMRI is widely used in the diagnosis of Alzheimer's Disease. By taking advantage of the high spatial resolution of fMRI, the diagnosis achieved an accuracy of above 90% in several studies. Another reason that contributes to competitive performance is the binary classification paradigm. Additionally, several publications aim to diagnose AD based on spontaneous EEG [125, 107].

Another area that has attracted much attention is the diagnosis of epileptic seizure. Seizure detection is mainly based on mental disease spontaneous EEG and occasionally on ECoG signals. The popular deep learning models in this scenario are independent CNN and RNN, along with hybrid models combining RNN and CNN. Some models integrated deep learning models for feature extraction and traditional classifiers for detection [130, 131]. For example, Yuan et al. [133] applied a D-AE in feature engineering followed by SVM for seizure diagnosis. Ullah et al. [119] adopted voting for post-processing, which proposed several different CNN classifiers and predicted the final result by voting.

Furthermore, there are a lot of other healthcare issues which can potentially be

solved by BCI systems. Cardiac artifacts in MEG signals can be automatically detected by deep learning models[176, 177]. Several modified CNN structures are proposed to detect brain tumors based on fMRI from the public BRATS dataset [196, 197, 198]. The literature demonstrates the effectiveness of deep learning models in the detection of a number of mental disorders such as depression [124], Interictal Epileptic Discharge (IED) [199], schizophrenia [200], Creutzfeldt-Jakob Disease (CJD) [138], and Mild Cognitive Impairment (MCI) [201].

2.3.2 Smart Environment

The smart environment is a promising application scenario for BCI in the future. With the development of Internet of Things (IoT), an increasing number of smart environments can be connected to BCI. For example, an assisting robot can be used in smart home [2, 88], in which the robot can be controlled by brain signals of the individuals. Moreover, Behncke et al. [171] and Huve et al. [202] investigated how to control a robot based on the visual stimulated spontaneous EEG and fNIRS signals. BCI controlled exoskeletons could help people with damaged to the motor control in the lower limbs in walking and daily activities [179]. In the future, research on brain-controlled appliances may be beneficial to the elderly people and the disabled in creating smart homes and smart hospitals.

2.3.3 Brain Communication

The biggest advantage of BCI, compared to other human-machine interface techniques, is that BCI enables patients who have lost most motor abilities, like speaking, to communicate with the outer world. Deep learning technology has substantially improved the efficiency of brain signal based communications. One typical paradigm which enables individual to type without any motor system is the P300 speller which can convert the user's intent into text [180]. Powerful deep learning models allow

the BCI systems to recognize P300 segments from non-P300 segments while the former contains the communication information of the user [181]. At a higher level, representative deep learning models can help to detect what character the user is focusing on and print it on the screen to chat with others [181, 182, 183].

Additionally, Zhang et al. [9] proposed a hybrid model combined RNN, CNN, and AE to extract informative features from MI EEG to recognize what letter the user wants to type. The proposed interface including 27 characters (26 English alphabets and the space bar) and all of them are separated by 3 character blocks (each block contains 9 characters) in the initial interface. Overall, there are three alternative selections, and each selection will lead to a specific sub-interface which includes 9 characters. Again, the $9 = 3 \times 3$ characters are divided into three character blocks, and each of them contains nine characters. Again, the $9 = 3 \times 3$ characters are divided into three character blocks, and each of them is connected to a lower level interface. In the bottom level, each block represents only one character. However, compared to P300 speller, the MI-based protocols have lower information transform rate because it requires three operations to find the specific letter at the bottom level.

2.3.4 Security

The security field is a common area of interest for BCI researchers. The security problem can be divided into identification (also called recognition) and authentication (also called verification) aspects. The former generally is a multi-class classification problem, and its aim is to recognize the identity of the test-person [8]. The latter usually is a binary classification problem, which only cares whether the test-person is authorized or unauthorized [74].

The existing biometric identification/authentication systems are mainly based on individuals' unique intrinsic physiological features (e.g., face, iris, retina, voice, and fingerprint). However, the state-of-the-art person identification systems are vulner-

able, e.g., anti-surveillance prosthetic masks can thwart face recognition, contact lenses can trick iris recognition, vocoders can compromise voice identification, and fingerprint films can deceive fingerprint sensors. In this perspective, the EEG (Electroencephalography) based biometric person identification systems are emerging as promising alternatives due to their high attack-resilience. An individualâĂŹs EEG signals are virtually impossible to mimic for an imposter, thus making this approach highly resilient to spoofing attacks encountered by other identification techniques. Koike et al. [184] have adopted deep neural networks to identify the user's ID based on VEP signals while Mao et al. [185] applied CNN for person identification based on RSVP signals. Zhang et al. [8] proposed an attention-based LSTM model and evaluated it over both public and local datasets. The authors [74] then combined EEG signals with gait information to introduce a dual-authentication system with a hybrid deep learning model.

2.3.5 Affective Computing

The affective states of a user provide critical information for many applications such as personalized information (e.g., multimedia content) retrieval or intelligent human-computer interface design [105]. Recent research illustrated that deep learning models can enhance the performance of affective computing. Emotion can be defined according to several dimensions. Dimensional models of emotion attempt to conceptualize human emotions by defining where they lie in two or three dimensions. The most widely used circumflex model states the emotions are distributed in two dimensions: arousal and valence. The arousal refers to the intensity of the emotional stimuli or how strong the emotion is. The valence refers to the relationship within the person who experiences the emotion (positive to negative). In some other models, the dominance and liking dimensions are used instead.

Some papers only attempt to classify the user's emotional state into a binary (positive/negative) or three-category (positive, neutral, and negative) problem and seek

to identify them using deep learning algorithms [94]. A range of publications adopted CNN and its variants to classify emotional EEG signals [95, 96, 98]. The DBN-RBM is the most representative deep learning model used to discover concealed features from emotional spontaneous EEG [105, 109]. Xu et al. [105] applied a DBN-RBM as specific feature extractors for the affective state classification problem using EEG signals.

Furthermore, at a more fundamental level, some researchers aim for the recognition of a positive/negative state for each specific emotional dimension. For example, Yin et al. [116] proposed a multiple-fusion-layer based ensemble classifier of AE for recognizing emotions. Each AE consists of three hidden layers to filter the unwanted noise in the physiological features and derives the stable feature representations. The proposed model was evaluated over the benchmark DEAP and achieved the arousal of 77.19% and valence of 76.17%. Mioranda et al. [168] presented a multi-task cascaded deep neural network which jointly predicts people's affective levels (valence and arousal) and personal factors using EEG signals recorded in response to the presentation of affective multimedia content.

2.3.6 Driver Fatigue Detection

Vehicle driver's ability to maintain optimal performance and attention is essential to ensure the safety of the traffic. EEG signals have been proven to be useful in evaluating peoples cognitive state during specific tasks [162]. Generally, the driver is regarded as being in an alert state if the reaction time is below or equal to 0.7 seconds and in a fatigued state if the reaction time is higher or equal to 2.1 seconds. Hajinoroozi et al. [160] considered the prediction of driver's fatigue from EEG signals by extracting the distinct features. They explored an approach based on DBN for dimensionality reduction.

The detection of driver fatigue is crucial because the drowsiness of the driver may lead to accidents. Additionally, driver fatigue detection is feasible in the real world.

In terms of the hardware, the equipment used to collect EEG signals is off-the-shelf and portable enough to be used in a car. Moreover, the price of an EEG headset is affordable for most people. In terms of the algorithms, deep learning models have greatly enhanced the performance of fatigue detection. As we summarized, the EEG based driving drowsiness can be recognized with high accuracy ($82\% \sim 95\%$).

The future scope of driver-fatigue detection is in the self-driving scenario. As we know, in most self-driving situations (e.g., Automation level 3[8]), the human driver is expected to respond appropriately to a request to intervene, which necessitates that the driver should maintain an alert state. Therefore, we believe the application of BCI based drive fatigue detection will benefit the development of the self-driving car.

2.3.7 Mental Load Measurement

Evaluation of operator mental workload levels via ongoing EEG is quite promising in Human-Machine collaborative task environments to alert when the operator performance is degraded[153]. The human operator works as a vital component in automation systems for decision making and strategy development. However, unlike machines or computers, the human functional states cannot always fit the task requirements due to limited working memory and time-dependent psychophysiological experience. Therefore, In such a case, operator performance degradation caused by abnormal cognitive states, e.g., high working stress or distraction, is considered to be a crucial factor for catastrophic accidents [203].

A number of researchers have focused on this topic. The mental workload can be measured from fNIRS signals or spontaneous EEG. Naseer et al. [204] analyzed and compared the classification accuracies of six different classifiers, including five traditional classifiers and a MLP classifier for a two-class mental task (mental arith-

[8]https://en.wikipedia.org/wiki/Self-driving_car

Table 2.6: The summary of public dataset for BCI systems. '# Sub', '# Cla', andS-Rate denote the number of subject, the number of class, and the sampling rate, respectively. FM denote finger movement while BCI-C denote the BCI competition. The datasets may contain more biometric signals (e.g., ECG) but we only list the channels related to BCI.

BCI Signals		Name Link	# Sub	# Cla	S-Rate	# Channel
EEG	Sleeping EEG	Sleep-EDF[9]: Telemetry	22	6	100	2 EEG, 1 EOG, 1 EMG
		Sleep-EDF: Cassette	78	6	100, 1	2 EEG (100Hz), 1 EOG (100Hz), 1 EMG (1Hz)
		MASS-1[10]	53	5	256	17/19 EEG, 2 EOG, 5 EMG
		MASS-2	19	6	256	19 EEG, 4 EOG, 1EMG
		MASS-3	62	5	256	20 EEG, 2 EOG, 3 EMG
		MASS-4	40	6	256	4 EEG, 4 EOG, 1 EMG
		MASS-5	26	6	256	20 EEG, 2 EOG, 3 EMG
		SHHS[11]	5804	N/A	125, 50	2 EEG (125Hz), 1 EOG (50Hz), 1 EMG (125Hz)
	Seizure EEG	CHB-MIT[12]	22	2	256	18
		TUH[13]	315	2	200	19
	MI EEG	EEGMMI[14]	109	4	160	64
		BCI-C II[15], Dataset III	1	2	128	3
		BCI-C III, Dataset III a	3	4	250	60
		BCI-C III, Dataset III b	3	2	125	2
		BCI-C III, Dataset IV a	5	2	1000	118
		BCI-C III, Dataset IV b	1	2	1001	119
		BCI-C III, Dataset IV c	1	2	1002	120
		BCI-C IV, Dataset I	7	2	1000	64
		BCI-C IV, Dataset II a	9	4	250	22 EEG, 3 EOG
		BCI-C IV, Dataset II b	9	2	250	3 EEG, 3 EOG
	Emotional EEG	AMIGOS[16]	40	4	128	14
		SEED[17]	15	3	200	62
		MAHNOB-HCI[18]	30	-	-	32
		DEAP[19]	32	4	512	32
	Others EEG	Open MIIR[20]	10	12	512	64
	VEP	BCI-C II, Dataset II b	1	36	240	64
		BCI-C III, Dataset II	2	26	240	64

metic and rest) using fNIRS signals. The experiment results showed that the MLP outperformed the traditional classifiers like SVM, kNN and achieved the highest accuracy of 96.3%. Bashivan et al. [151] presented a statistical approach, a DBN model, to predict cognitive load from single trial EEG. Before the DBN, the authors manually extracted the wavelet entropy and band-specific power from theta, alpha and beta bands. Finally, the experiments demonstrated the recognition of cognitive load across four different levels with an overall accuracy of 92% during execution of a memory task.

2.3.8 Auditory Assistance

One application of auditory steady state responses in BCI is to determine the signal of interest for assistive hearing devices such as hearing aids and cochlear implants. Deep learning approaches have been shown to be very effective for suppressing interfering background noise and thus improving speech understanding for hearing impaired listeners [205] and cochlear implant users [206] in the case of a known speaker. However, for real-world applications, the source of interest needs to be identified: for example, which of two talkers is the listener trying to understand and which should be suppressed by the system? When a listener is paying attention to one particular talker in a mixture their neural signals will become more correlated with the changes in energy of that talkers speech over time (the speech envelope) than with that of the other speech in the mixture [207]. This allows the focus of the listeners attention to be decoded from the neural signal.

Performance of neural decoders is greatest when ECoG signals are used [208], but approaches using EEG have also shown high performance [209], and above chance performance when just behind the ear electrodes are used [210]. One approach that has been employed is to use deep learning to separate the components of the audio signal and then use linear approaches to do the neural decoding - determining which

[9]https://physionet.org/physiobank/database/sleep-edfx/

[10]https://massdb.herokuapp.com/en/

[11]https://physionet.org/pn3/shhpsgdb/

[12]https://physionet.org/pn6/chbmit/

[13]https://www.isip.piconepress.com/projects/tuh_eeg/html/downloads.shtml

[14]https://physionet.org/pn4/eegmmidb/

[15]http://www.bbci.de/competition/ii/

[16]http://www.eecs.qmul.ac.uk/mmv/datasets/amigos/readme.html

[17]http://bcmi.sjtu.edu.cn/ seed/download.html

[18]https://mahnob-db.eu/hci-tagging/

[19]https://www.eecs.qmul.ac.uk/mmv/datasets/deap/

[20]https://owenlab.uwo.ca/research/the_openmiir_dataset.html

of the unmixed speech signals was being attended to by the listener [211]. More recent approaches have used convolutional neural networks to determine directly the signal of interest, either by taking speech envelopes and the EEG signals as its input [212] or, without access to the speech signals, using the EEG signal alone to determine whether the listener is attending to a talker on their left or right [213].

2.3.9 Other Applications

Apart from the aforementioned key applications, there are still some interesting scenarios, such as recommender system [172] and emergency braking [170] to which deep learning based BCI can be applied. One possible topic is the recognition of a visual object, which may be used in guilty knowledge test [194] and concealed information test [195]. The neurons of the participant will produce a pulse when he/she suddenly perceives a familiar object. Based on the theory, visual target recognition main uses RSVP signals. Cecotti et al. [190] investigated the performance of CNNs in terms of their architecture and how they are evaluated. Specifically, the authors aimed to build a common model target recognition which can work for various subjects instead of a specific subject. They addressed the change of performance that can be observed between specifying a neural network for a single subject, or by considering a neural network for a group of subjects, taking advantage of a larger number of trials from different subjects.

Other researchers have investigated whether it is possible to distinguish the subject's gender using fNIRS [186] or spontaneous EEG [169]. Hiriyasu et al. [186] adopted deep learning to recognize the gender of the subject based on the cerebral blood flow. The experiment results suggested that there exists a relation between cerebral blood flow changes and biological information. Putten et al. [169] tried to discover the sex-specific information from the brain rhythms and adopted a CNN model to recognize the participant's gender. This paper illustrated that fast beta activity (20 $\sim$25 Hz), and its spatial distribution is one of the main distinctive attributes.

2.4 Benchmark Datasets

BCI related experiments is hard to be conducted due to the lack of equipment, domain knowledge, and volunteered subjects. In this section, we extensively explore the benchmark datasets which can be used in deep learning based BCI. As listed in Table 2.6, we provide 27 reusable public datasets with download links, which cover most BCI signals. The BCI competition IV (BCI-C IV) contains five datasets. We give the access link at the first dataset. For better understanding, we present the number of subjects, the number of classes (how many categories), sampling rate and the number of channels of each dataset. In the '# Channel' column, the default channel is EEG signals.

2.5 Conclusion

In this chapter, we systematically summarize the recent advances in deep learning models for Brain-Computer Interface. Compared with traditional methods, deep learning not only enables to learn high-level features automatically from BCI signals but also depends less on manual-crafted features and domain knowledge. We overview dominant deep learning models, followed by discussing state-of-the-art deep learning techniques for BCI. Finally, we provide the common used BCI applications and present a batch of public available BCI datasets.

Part II

Effective Feature Learning

This part contains works published in:

[2] **X. Zhang**, L. Yao, C. Huang, Q. Z. Sheng, and X. Wang. Intent recognition in smart living through deep recurrent neural networks. In *Proceedings of the 24th International Conference on Neural Information Processing (**ICONIP 2017**)* (pp. 748—758). Springer, 2017. (CORE Rank A)

[3] **X. Zhang**, L. Yao, D. Zhang, X. Wang, Q. Z. Sheng, and T. Gu. Multi-person brain activity recognition via comprehensive EEG signal analysis. In *Proceedings of the 14th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services (**MobiQuitous 2017**)* (pp. 28-37). ACM, 2017. (CORE Rank A)

Chapter 3

Intent recognition through deep recurrent neural networks

As we mentioned before, EEG signals reflecting activities on certain brain areas not requiring any initiative actions is an effective method to connect the individuals and the outer world. The key challenge is how to learn the distinctive features from the noisy EEG signals in order to recognize the user's intent. In this chapter, taking smart living as an example, we introduce an effective feature learning method, i.e., recurrent neural network, for EEG-based intent recognition.

Intent recognition can be used to assist individuals who have troubles in motor abilities (such as aged individuals having motor neuron disease like Parkinson disease). EEG-based intent control has shown promising performance in various applied fields. For instance, if a person suffers from Amyotrophic Lateral Sclerosis (ALS) would have only very limited physical capacities, she would be unable to communicate with the outer world, such as performing most daily activities (e.g., turn on/off the light). It can be very difficult for her to use normal platforms to control smart home ap-

pliances. EEG signals[1] are generated when a subject imagines performing a certain action such as close hands. Therefore, EEG signal are widely captured to recognize one's intent, with the intent of using it as input to communicate or interact with external smart devices such as wheelchairs or service robots a real-time BCI systems [217].

So far, existing EEG-based intent recognition approaches face several challenges. First, the data pre-processing, parameters selection and feature engineering (e.g., feature selection and extraction both in time domain and frequency domain) are time-consuming and highly dependent on human expertise. Second, current accuracies mostly center around $60 \sim 85\%$ [218, 219, 220], which are too low for real-world deployment. Finally, existing research mainly focus on binary intents recognition while multi-intent scenario dominates the practical applications. The more scenarios can be distinguished, the more capabilities of EEG-based control systems can be expanded in real-world applications.

On the other hand, deep learning based approaches are capable of modeling high level representations as well as capturing complex relationships, which are often hidden in raw data, via stacking multiple layers of information processing modules in hierarchical architectures [221]. RNN is one typical model making use of sequential information. In particular, Long Short-Term Memory (LSTM) is one RNN architecture designed to model temporal sequences and their long-range dependencies, and often results in higher accurate compared to conventional RNNs [222]. In this work, we propose a deep recurrent neural network model for intent recognition , to help individuals with motor impairments. Reusable source code and dataset are provided to reproduce the results[2]. Our main contributions of this chapter are highlighted as below:

- We propose a LSTM recurrent neural network for intent recognition, which

[1]In this chapter, the mentioned EEG refers to MI EEG.

[2]https://github.com/xiangzhang1015/EEG-based-Control

directly processes raw EEG data under multi-class scenario.

- We apply Orthogonal Array experiment method for hyper-parameters tuning, which saves 98.4% of time compared to exhausting tuning.
- We evaluate our approach over an open EEG dataset and achieves 0.9325 of accuracy.

3.1 Related Work

The current application of EEG signals is mainly in medicine and neurology. [223] proposes a Logistic Regression (LR) approach to analyze EEG signals to detect seizure patient and achieves as high as 91% of accuracy. Wavelet analysis [224] is employed to carry on a diagnosis of Traumatic Brain Injury (TBI) by quantitative EEG (qEEG) data and reaches 87.85% of accuracy. Power spectral density [225] are extracted as EEG data features to input into SVM, extreme learning machine and linear discriminant analysis to predict the outcome of Transcranial direct current stimulation treatment. The work achieves 76% accuracy with the data from FC4 $\sim$ AF8 channels and 92% with the data from CPz $\sim$ CP2 channels.

All the aforementioned literature uses binary classification and extracts features in different areas manually. Recent research focuses more on the performance comparison of different classifiers. [226] builds one deep belief net classifier for each channel and combines them through Ada-boost algorithm and classifies the left and right hand motor imagery.The work achieves average 83% accuracy. [219] adopts SVM as the classifier and achieves an average accuracy of 65% with the input data being denoised by a wavelet denoising algorithm before power spectral density feature selection. [227] yields an accuracy of 80% with the foundational universal background models (UBMs) classifier after the data is processed by I-vectors and Joint Factor Analysis (JFA). [228] combined convolutional neural networks and stacked autoencoders to classify EEG Motor Imagery signals and results 90% accuracy. The

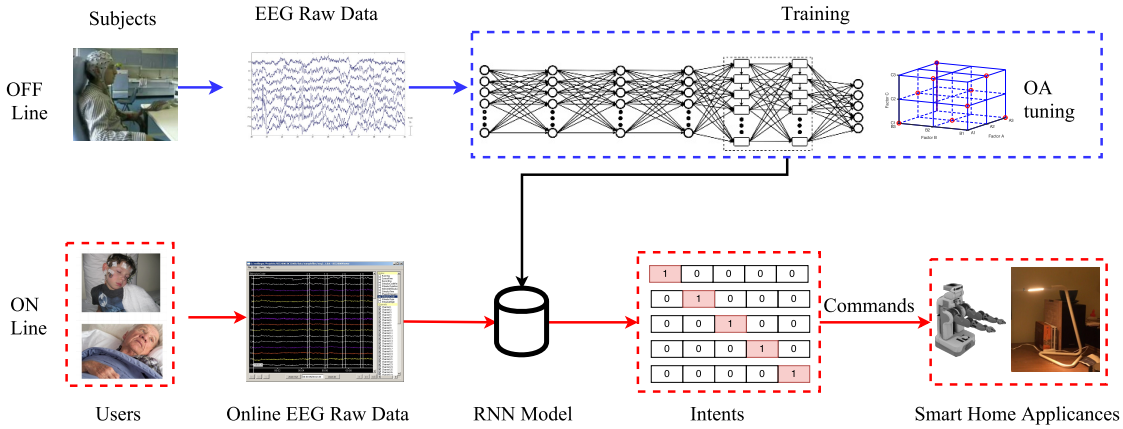


Figure 3.1: Workflow of the Proposed Approach

application of related methods in smart living in relatively limited. As an example, [229] uses high pass and low pass filter to reduce the noise signal interference and extracts EEG features by fisher distance. The switch control experiment results show that their approach achieves an accuracy of 86%.

3.2 The Proposed Approach

In this section we introduce the flow chart of the proposed approach at first and then involve to more details. The architecture of our approach is shown in Figure 3.1. The system consists of two components: the online component and the offline component.

In the online component, raw EEG data, collected from subjects, are used to train a deep recurrent neural network model (Section 3.2.1). The model directly works on raw EEG data without any pre-processing, smoothing, filtering or feature extraction. The parameters in the deep learning model are optimized by the Orthogonal Array experiment (Section 3.2.2). In the offline component, the user's willing (EEG signal) is sent to above pre-trained RNN model and then recognized as specific intent. The intent is subsequently used to command devices, such as turning lights on/off or driving a robot to serve a cup of water.

3.2.1 LSTM Recurrent Neural Network

RNN, as a class of deep neural networks, can help to explore the feature dependencies over time through an internal state of the network, which allows us to exhibit dynamic temporal behavior. In order to precisely recognize the user's intent, we propose a 7-layer LSTM Recurrent Neural Network model including three components: 1 input layer, 5 hidden layers, and 1 output layer. In hidden layers, two of them are consisted of LSTM cells [230] (shown as the rectangles in Figure 3.1).

Assume one collection of EEG signals is $E = \{E_1, E_2, ..., E_j, ..., E_{bs}\}, E_j \in \mathbb{R}^K$ with n_{bs} denotes the batch size, j denotes the j-th EEG sample, and K denotes the number of dimensions in each EEG raw signal ($K = 64$). And in the RNN model, we denote the i-th layer ($i = 1, 2, \cdots, I$, $I = 7$) $X_i^r = \{X_{ijk}^r | k = 1, 2, \cdots, K_i\}, X_i^r \in \mathbb{R}^{[n_{bs}, 1, K_i]}$ ($K_1 = K = 64$), where K_i denotes the dimension of the layer. Note that the number of dimension equals to the amount of neurons accordingly in each layer. When the input only contains one EEG sample, the first layer can be $X_1^r = E_j$.

Weights between layer i and layer $i+1$ can be denoted as $W_{i,(i+1)}^r \in \mathbb{R}^{[K_i, K_{i+1}]}$, for instance, $W_{1,2}^r$ describes the weight between layer 1 and layer 2. $b_i^r \in \mathbb{R}^{K_i}$ denotes the biases of i-th layer. The connection between the i-th and $(i+1)$-th layer will be $X_{i+1}^r = X_i^r * W_{i,i+1}^r + b_i^r$.

Please note the sizes of X_i^r, $W_{i,i+1}^r$ and b_i^r must match. For example, in Figure 3.1, the transformation between H1 layer and H2 layer, the sizes of X_3^r, X_2^r, $W_{[2,3]}$, and b_2^r are correspondingly $[n_{bs}, 1, K_3]$, $[n_{bs}, 1, K_2]$, $[K_2, K_3]$, and $[n_{bs}, 1]$. The 5-th and 6-th layers here are LSTM layers, and they can be connected by:

$$f_i = sigmoid(T(X_{(i-1)j}^r, X_{(i)(j-1)}^r)) \tag{3.1}$$

$$f_f = sigmoid(T(X_{(i-1)j}^r, X_{(i)(j-1)}^r)) \tag{3.2}$$

$$f_o = sigmoid(T(X_{(i-1)j}^r, X_{(i)(j-1)}^r)) \tag{3.3}$$

$$f_m = tanh(T(X_{(i-1)j}^r, X_{(i)(j-1)}^r)) \tag{3.4}$$

$$c_{ij} = f_f \odot c_{i(j-1)} + f_i \odot f_m \tag{3.5}$$

$$X_{ij}^r = f_o \odot tanh(c_{ij}) \tag{3.6}$$

where f_i, f_f, f_o and f_m represent the input gate, forget gate, output gate and input modulation gate accordingly, and $\odot$ denotes the element-wise multiplication. The c_{ij} denotes the state (memory) in the j-th LSTM cell in the i-th layer, which is the most significant part to explore the time-series relevance between samples. The $T(X_{(i-1)j}^r, X_{(i)(j-1)}^r)$ denotes the operation as follows:

$$X_{(i-1)j}^r * W + X_{(i)(j-1)}^r * W' + b \tag{3.7}$$

where W, W' and b denote the corresponding weights and biases. At last, we obtain the RNN predict results X_7^r and employ the cross-entropy as the cost function. The ℓ_2 norm is selected as the regularization function and the cost is optimized by the AdamOptimizer algorithm [231].

3.2.2 Orthogonal Array Tuning Method

Although deep learning algorithms can generally achieve good performance in many areas, tuning the hyper-parameters (e.g., the number of layers, the number of nodes in each layer and the learning rate) is time-consuming and dependent on one's experience. This chapter employs the Orthogonal Array Tuning Method (OATM) [232] to select the hyper-parameters, which works *much faster* than traditional hyper-parameters tuning methods. OATM[3] is widely used in *design of experiments, coding theory, and cryptography*, however, to our best knowledge, this chapter is the very first work to apply OATM of the parameter tuning in machine learning and data mining areas. More details are shown in Chapter 8.

OATM is a systematic and statistical method and its principle is to compare the dependent variable which is resulted from a different combination of independent

[3] `https://www.york.ac.uk/depts/maths/tables/taguchi_table.htm`

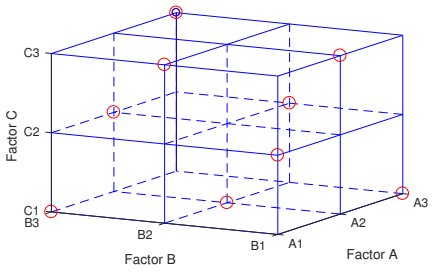


Figure 3.2: OATM selection

	Level 1	**Level 2**	**Level 3**	**Level 4**
λ	0.002	0.004	0.006	0.008
lr	0.005	0.01	0.015	0.02
K_i	16	32	48	64
I	5	6	7	8
n_b	1	3	6	13

Table 3.1: Factors and levels

variables. It chooses certain representative combinations instead of all combinations for testing. In this method, independent variable is called “factor" and different values of factor are called “levels". For instance, if the program has three factors and each of them has three levels, which are represented by a cube with 27 nodes (each node represents one combination of hyper-parameters), OATM only chooses 9 representative groups of parameters to optimize the selection. As shown in Figure 3.2, A_1, A_2, A_3 represent 3 levels of factor A, while factors B, C are by the same token (the factor is supposed to be statistically independent with the others). The 9 circled nodes are the nine groups selected by OATM. Each edge (totally 27 edges) in the cube has one circled node and each face (totally 9 faces) has three circled nodes.

For different number of factors and levels, corresponding OATM table is provided. Generally, an OATM table can be written as $Ln_a(n_b^{n_c})$, where n_a denotes the number of hyper-parameter combination, n_b denotes the number of levels of each factor and n_c denotes the number of factors.

3.3 Experiments

3.3.1 Dataset

We use the EEG data from PhysioNet eegmmidb (*EEG motor movement/imagery database*) database, a widely used EEG database collected by the BCI2000 (Brain Computer Interface) instrumentation system [233, 234], to evaluate the proposed method. In particular, the data is collected by the BCI 2000 system, which owns 64 channels and an EEG data sampling rate of 160 Hz. During the collection of this database, the subject sits in front of one screen and performs the corresponding action as one target appears in different edges of the screen. According to the tasks, different annotations are labeled and can be downloaded from PhysioBank ATM. The actions in different tasks are as follows:

Task 1: The subject closes his or her eyes and keeps relax.

Task 2: A target appears at the left side of the screen and then the subject focuses on the *left hand* and *imagines* he/she is opening and closing the left hand until the target disappears.

Task 3: A target appears at the right side of the screen and then the subject focuses on the *right hand* and *imagines* he/she is opening and closing the right hand until the target disappears.

Task 4: A target appears on the top of the screen, and the subject focuses on *both hands* and *imagines* he/she is opening and closing both hands until the target disappears.

Task 5: A target appears on the bottom of the screen, and the subject focuses on *both feet* and *imagines* he/she is opening and closing both feet until the target disappears.

Specifically, we select 560,000 EEG samples from 20 subjects (28,000 samples each

subject) for our experiments. Every sample is one vector which includes 64 elements corresponding to 64 channels. Each sample corresponding to one intent (from intent 1 to intent 5 separately is *eye closed*, *focus on left hand*, *focus on right hand*, *focus on both hands* and *focus on both feet*).

3.3.2 Overall Comparison

This section is aimed to demonstrate the efficiency of the proposed approach, for which we compare our approach with the state-of-the-art methods. Our model is composed of 7 layers RNN with 2 LSTM layers, the learning rate and the λ are set as 0.004 and 0.005, the number of the nodes in each hidden layer is 64 and the number of batches n_b is 3 (detailed in Section 3.3.3).

Our intent recognition result, the confusion matrix is presented in Table 3.4. It can be read that our approach produces a mean accuracy of **0.9325**, in tests of five intents recognition on 10 subjects. The ROC (Receiver Operating Characteristic) curves of five intents are displayed in Figure 3.3. Additionally, comparison with the state-of-the-art methods is shown in Table 3.3 (the Binary/Multi column refers binary intents recognition or multi-intents recognition). The KNN sets the number of neighbors as 3; the SVM adopts One-vs-the-rest (OvR) multi-class strategy and the estimator is LinearSVC; the RF sets the number of estimators as 300; the AdaBoost adopts the number of estimators as 50 and the learning rate as 0.3; all the not mentioned parameters are set as default values. We can perceive that the proposed approach significantly outperforms all the state-of-the-art methods, by a large margin of 10%.

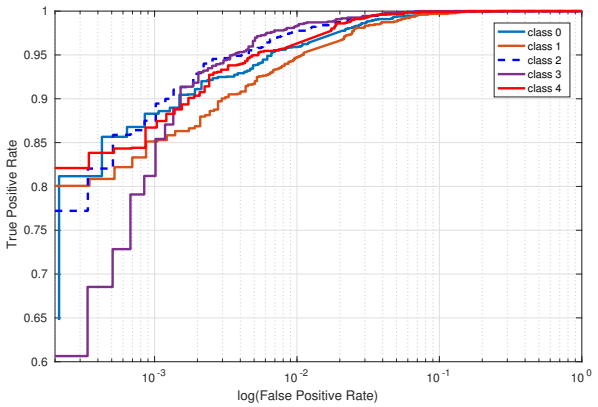


Figure 3.3: ROC curves

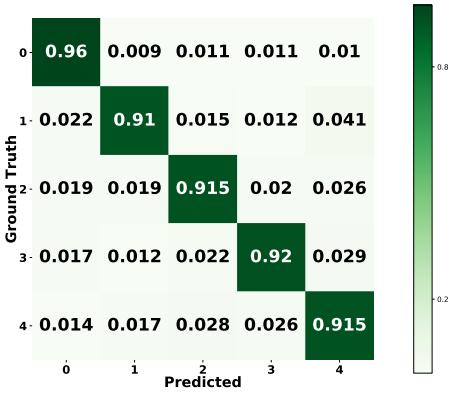


Figure 3.4: Confusion Matrix

3.3.3 Hyper-parameter Tuning

The intent recognition results rely on hyper-parameters since we adopt deep learning model. To achieve optimal recognition accuracy, we employ OATM to optimize the hyper-parameters. We select five most common hyper-parameters including λ (the coefficient of ℓ_2 norm), lr (learning rate), K_i(the hidden layer nodes size), I (the number of layers), and n_b (the number of batches[4]), and they are shown in Table 3.1. Since this OATM experiment contains 5 factors and 4 levels, the total number of factor combinations can be found in *the standard orthogonal experiment table*[5]. As shown in the standard orthogonal experiment table, 5 factors with 4 levels OATM experiment has 16 different combine ways, which means 16 experiments should be conducted to optimize the hyper-parameters. The combination of hyper-parameters and the range analysis of results of the experiment, are shown in Table 3.2. The optical λ, lr, K_i, I, and n_b tuned by OATM are 0.004, 0.005, 64, 7, and 3, respectively. The parameter selection of 5 factors and 4 levels needs $1024 = 4^5$ combinations in an exhaustive method, while with OATM only 16 combinations are needed. This

[4]The size of training dataset and testing dataset depends on n_b since the total dataset is fixed, e.g., if n_b equals 1, there will be 14,000 training dataset and 14,000 testing dataset. If n_b equals 3, we will have 21,000 training dataset and 7,000 testing dataset

[5]`https://www.york.ac.uk/depts/maths/tables/l16b.htm`

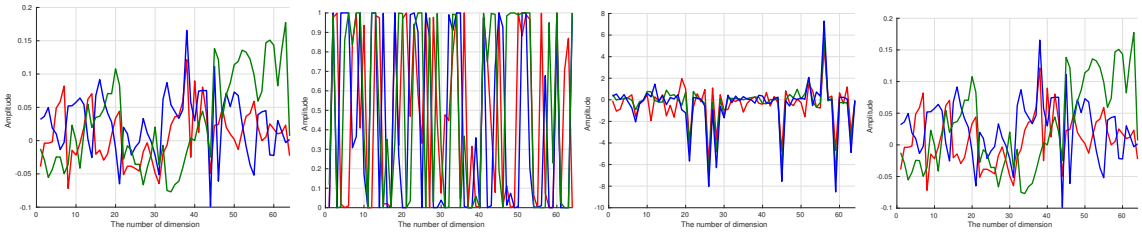
No.	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
λ	0.002	0.002	0.002	0.002	0.004	0.004	0.004	0.004	0.006	0.006
lr	0.005	0.01	0.015	0.02	0.005	0.01	0.015	0.02	0.005	0.01
K_i	16	32	48	64	32	16	64	48	48	64
I	5	6	7	8	7	8	5	6	8	7
n_b	1	3	6	13	13	6	3	1	3	1
Acc	0.689	0.91	0.893	0.667	0.925	0.717	0.848	0.77	0.926	0.826
11	**12**	**13**	**14**	**15**	**16**	R_{level1}	R_{level2}	R_{level3}	R_{level4}	**Best level**
0.006	0.006	0.008	0.008	0.008	0.008	3.159	**3.26**	2.441	2.44	**0.004**
0.015	0.02	0.005	0.01	0.015	0.02	**3.47**	2.875	2.747	2.208	**0.005**
16	32	64	48	32	16	2.132	2.886	3.011	**3.271**	**64**
6	5	6	5	8	7	2.326	2.932	**3.048**	2.894	**7**
13	6	6	13	1	3	2.969	**3.088**	2.907	2.336	**3**
0.322	0.367	0.93	0.422	0.684	0.404					

Table 3.2: OATM experiment factor analysis

means $(1 - 16/1024) = \mathbf{98.4\%}$ of time are saved. In Table 3.2, R_{leveli} is the sum of accuracy of all the combinations contains $level_i$. We selected the best levels listed in Table 3.2 for training the model and obtain an accuracy of **0.9325**.

3.3.4 Feature Evolution

To better understand the essence of the proposed model, we graphically describe the feature evolution procedures. Figure 3.5 shows the revolution of variations between samples from different classes. In the input layer, the samples are chaotic entangled; and they become clear and observable in the last LSTM layer after the training through several hidden layers. Particularly, in Figure 3.5d, the black rectangles display parts of the dimensions which can clearly show the difference between the intents. Conclusively, the proposed approach is enabled to automatically extract distinguishable features (Figure 3.5d) from the chaotic raw EEG data (Figure 3.5a).



(a) input layer (b) hidden layer 1 (c) hidden layer 3 (d) LSTM layer 2

Figure 3.5: Feature evolution. The black rectangles in Figure 3.5d indicate the features which can clearly show the difference between the various intents.

3.4 Conclusion

In this chapter, we present an LSTM approach to recognize the smart living user intents in EEG raw signals. By experimenting on large scale EEG dataset, we can claim that our proposed approach significantly outperforms a series of the state-of-the-art methods by achieving 0.9325 of accuracy. It provides insight into feature revolution by visualizing the data shape, waveform fluctuation flowing through each layer of our proposed model.

	Index	Methods	Binary/Multi	Accuracy
State of the art	1	Almoari [217]	Binary	0.7497
	2	Sun [219]		0.65
	3	Major [218]		0.68
	4	Shenoy [220]		0.8206
	5	Tolic [235]		0.6821
	6	Ward [227]	Multi (3)	0.8
	7	Pinheiro [236]	Multi (4)	0.8505
Baselines	8	KNN (k=3)	Multi (5)	0.8369
	9	SVM		0.5082
	10	RF		0.7739
	11	LDA		0.5127
	12	AdaBoost		0.3431
	13	CNN		0.8409
	14	Ours		**0.9325**

Table 3.3: Performance comparison with the state of the art methods. RF: Random Forest, LDA: Linear Discriminant Analysis. All the methods are evaluated using the same database.

Chapter 4

Multi-Person Intent Recognition via Comprehensive EEG Signal Analysis

Apart from person-independent EEG feature learning introduced in the previous chapter, in this chapter, we propose one novel method for multi-person effective EEG feature learning. Person-independent refers to the training and testing dataset are gathered from the same subject, while multi-person refers to the testing data are randomly selected from different subjects.

Although intent recognition has been widely investigated over the last several years, it still faces several challenges such as multi-person and multi-class classification. First, despite several studies on multi-person EEG classification, e.g., [237] employed a LDA classifier to classify two datasets with nine and three subjects, there still has space for improvement over the existing methods in terms of the classification accuracy (86.06% and 93% over the two datasets in [237]). Second, to the best of our knowledge, most existing applications that adopt EEG classification are for diseases diagnosis (such as epilepsy and Alzheimer's diseases), which requires only binary classification (normal or abnormal). However, there exist various other deployment occasions (e.g., smart home and assisted living) that demand multi-class

EEG classification. For instance, EEG-based assisting robots require more than two commands (such as walking straight, turning left/right, and raising/lowering hands) to complete assisted living tasks. Regarding this, only some preliminary research exists, such as [238], which adopted SVM to classify a four-class EEG dataset and achieved the accuracy of 70%.

In this chapter, we propose a novel intent recognition approach to classifying the multi-person and multi-class EEG data. We analyze the similarity of EEG signals and calculates the correlation coefficients matrix in both inter-class and inter-person conditions. Then, on top of data similarity analysis, we extract EEG signal features by the Autoencoder algorithm, and finally feed the features into the XGBoost classifier to recognize categories of EEG data, with each category corresponding to one specific intent. The main contributions of this chapter are summarized as follows:

- We present a novel intent recognition approach based on comprehensive EEG analysis. The proposed approach directly works on the raw EEG data, which enhances the ductility, relieves from EEG signal pre/post-processing, and decreases the need of human expertise.
- We calculate the correlation coefficients matrix and measure the self-similarity and cross-similarity under both inter-class and inter-person conditions. Based on the similarity investigation, we propose three favorable conditions of multi-person and multi-class EEG classification.
- We adopt the Autoencoder, an unsupervised neuron network algorithm, to refine EEG features. Moreover, we investigate the size of hidden-layer neurons to optimize the neurons size to optimize the classification accuracy.
- We conduct an experiment to evaluate the proposed approach on a public EEG dataset (containing 560,000 samples from 20 subjects and 5 classes) and obtain the accuracy of 79.4%. Our approach achieves around 10% accuracy improvement compared with other popular EEG classification methods.

- We design a case study to evaluate the proposed approach on a local dataset which consists of 172,800 samples collected from 5 subjects and 6 classes. Our approach obtains the accuracy of 74.85% and outperforms the result of the state-of-the-art methods.

4.1 Related Work

Over the last decade, much attention has been drawn to brain data modeling, a crucial pathway to translating human intent into computer commands to realize Brain-Computer Interaction (BCI). BCI systems are an alternative way to allow paralyzed or severely muscular disordered patients to recover communication and control abilities, as well as to save scarce medical care resources. Recent research has also found its application in virtual reality [239] and space applications [240]. As EEG signals are the most commonly used brain data for BCI system [241, 242, 243], significant efforts have been devoted to build accurate and effective models for EEG-based intent analysis [244, 245, 246, 247].

EEG Feature Representation Method. Feature representation of EEG raw data has great impact on classification accuracy due to the complexity and high dimensionality of EEG signals. VÃľzard et al. [248] employed common spatial pattern (CSP) along with LDA to pre-process EEG data and obtained an accuracy of 71.59% on binary alertness states. Meisheri et al. [249] exploited multi-class CSP (mCSP) combined with Self-Regulated Interval Type-2 Neuro-Fuzzy Inference System (SRIT2NFIS) classifier for four EEG-based motor imagery classes (movement imagination of left hand, right hand, both feet, and tongue) classification and achieved the accuracy of 54.63%, which is significantly lower than the accuracy of binary classification. Shiratori et al. [250] achieved a similar accuracy of 56.7% using mCSP coupled to the random forests for a three-class EEG-based motor imagery task. The autoregressive (AR) modeling approach, a widely used algorithm for EEG feature extraction, is also broadly combined with other feature extraction techniques

to gain a better performance [251]. For example, [252] investigated two methods EEG with AR and feature extraction combination: 1) AR model and approximate entropy, 2) AR model and wavelet packet decomposition. They employed SVM as the classifier and showed that AR can effectively improve classification performance. Duan et al. [253] introduced the Autoencoder method for feature extraction and obtained an accuracy of 86.69%.

EEG Multi-person Classification. Multi-person EEG classification investigates mental signals from multiple participants, each of whom undergoing the same brain activities. It is the requirement of future ubiquitous application of EEG instruments to capture the underlying consistency and inter-subject variations among EEG patterns of different subjects. Kang et al. [254] presented a Bayesian CSP model with Indian Buffet process (IBP) to investigate the shared latent subspace across subjects for EEG classification. Their experiments on two EEG datasets containing five and nine subjects showed the superior performance of approximate 70% accuracy. Djemal et al. [237] utilized two multi-person multi-class EEG datasets to validate sequential forward floating selection (SFFS) and a multi-class LDA algorithm. Eugster et al. [255] involved forty participants in their experiments to perform relevance judgment tasks. They also recorded the EEG signals for further classification research. Ji et al. [256] investigated a dataset containing nine subjects for analyzing and evaluating a hybrid brain-computer interface.

EEG Multi-class Classification. Multi-class classification is a major challenge in EEG signal analysis, given that current EEG classification research is mostly focused on binary classification. Usually, an algorithm achieves only inferior performance when handling multi-classification than in handling binary classification. Anh et al. [257] used Artificial Neural Network trained with output weight optimization back-propagation (OWO-BP) training scheme for dual- and triple-mental state classification problems. They got a classification accuracy of 95.36% on dual mental state for triple classification problems, the algorithm performance fell off to 76.84%. With the four-class problem, Olivier et al. [258] got an accuracy of around

Table 4.1: Inter-class correlation coefficients matrix. The correlation coefficients matrix (upper left section) is the average of 20 correlation coefficients matrix separately from 20 subjects. Self-S, Cross-S, and PD denote self-similarity, cross-similarity, and percentage difference, respectively.

Class	**0**	**1**	**2**	**3**	**4**	**Self-S**	**Cross-S**	**PD**
0	0.4010	0.2855	0.4146	0.4787	0.3700	0.401	0.3872	3.44%
1	0.2855	0.5100	0.0689	0.0162	0.0546	0.51	0.1063	79.16%
2	0.4146	0.0689	0.4126	0.2632	0.3950	0.4126	0.2854	30.83%
3	0.4787	0.0162	0.2632	0.3062	0.2247	0.3062	0.2457	19.76%
4	0.3700	0.0546	0.3950	0.2247	0.3395	0.3395	0.3156	7.04%
Range	0.1932	0.4938	0.3458	0.4625	0.3404	0.2038	0.2809	75.72%
Average	0.3900	0.1870	0.3109	0.2578	0.2768	0.3939	0.2680	28.05%
STD	0.0631	0.1869	0.1334	0.1487	0.1255	0.0700	0.0932	27.33%

50% when using a voting ensemble neural network classifier. Aiming at four-class EEG classification, Wang et al. [238] employed four preprocessing steps and a simple SVM classifier and got an average classification accuracy of 70%.

In summary, differing from previous work, this chapter proposes an *Autoencoder+XGBoost* algorithm to address the multi-class multi-person EEG signal classification problem, which is a core challenge in applying intent recognition technologies to many important domains. The proposed algorithm engages the Autoencoder for EEG feature representation to explore the relevant EEG features. Also, it emphasizes on the generalization over participants by solving an EEG classification problem with as much as five classes and taking twenty subjects. The present approach is supposed to improve the accuracy and practical feasibility of EEG classification.

4.2 EEG Characteristic Analysis

To gain knowledge about EEG data characteristics and prepare for the further EEG classification, we quantify the similarity between EEG samples by calculating their Pearson correlation coefficients, using the following equation:

Table 4.2: Inter-person correlation coefficients matrix. STD denotes Standard Deviation, SS denotes Self-similarity, CS denotes Cross-similarity, and PD denotes Percentage Difference.

	Class 0			Class 1			Class 2			Class 3			Class 4		
subjects	**SS**	**CS**	**PD**	**SS**	**CS**	**PD**	**SS**	**CS**	**PD**	**SS**	**CS**	**PD**	**SS**	**CS**	**PD**
subject1	0.451	0.3934	12.77%	0.2936	0.1998	31.95%	0.3962	0.3449	12.95%	0.4023	0.1911	52.50%	0.5986	0.4375	26.91%
subject2	0.3596	0.2064	42.60%	0.3591	0.1876	47.76%	0.5936	0.3927	33.84%	0.2354	0.2324	1.27%	0.3265	0.2225	31.85%
subject3	0.51	0.3464	32.08%	0.3695	0.2949	20.19%	0.3979	0.3418	14.10%	0.4226	0.3702	12.40%	0.4931	0.4635	6.00%
subject4	0.3196	0.1781	44.27%	0.4022	0.1604	60.12%	0.3362	0.2682	20.23%	0.4639	0.3905	15.82%	0.3695	0.2401	35.02%
subject5	0.4127	0.2588	37.29%	0.3961	0.2904	26.69%	0.3128	0.2393	23.50%	0.4256	0.1889	55.62%	0.3958	0.3797	4.07%
subject6	0.33	0.2924	11.39%	0.3869	0.3196	17.39%	0.3369	0.3281	2.61%	0.4523	0.1905	57.88%	0.4526	0.3321	26.62%
subject7	0.4142	0.3613	12.77%	0.3559	0.342	3.91%	0.3959	0.3867	2.32%	0.4032	0.3874	3.92%	0.4862	0.2723	43.99%
subject8	0.362	0.1784	50.72%	0.4281	0.2121	50.46%	0.4126	0.2368	42.61%	0.3523	0.1658	52.94%	0.4953	0.2438	50.78%
subject9	0.324	0.2568	20.74%	0.3462	0.2987	13.72%	0.3399	0.3079	9.41%	0.3516	0.1984	43.57%	0.3986	0.177	55.59%
subject10	0.335	0.1889	43.61%	0.3654	0.2089	42.83%	0.2654	0.2158	18.69%	0.3326	0.2102	36.80%	0.3395	0.2921	13.96%
subject11	0.403	0.1969	51.14%	0.3326	0.2066	37.88%	0.3561	0.3173	10.90%	0.4133	0.1697	58.94%	0.5054	0.44	12.94%
subject12	0.4596	0.2893	37.05%	0.4966	0.3702	25.45%	0.3326	0.2506	24.65%	0.4836	0.3545	26.70%	0.3968	0.3142	20.82%
subject13	0.3956	0.2581	34.76%	0.4061	0.3795	6.55%	0.3965	0.3588	9.51%	0.3326	0.1776	46.60%	0.3598	0.3035	15.65%
subject14	0.3001	0.299	0.37%	0.3164	0.2374	24.97%	0.4269	0.3763	11.85%	0.3856	0.1731	55.11%	0.4629	0.3281	29.12%
subject15	0.3629	0.3423	5.68%	0.3901	0.2278	41.60%	0.7203	0.2428	66.29%	0.3623	0.3274	9.63%	0.3862	0.3303	14.47%
subject16	0.3042	0.1403	53.88%	0.3901	0.3595	7.84%	0.4236	0.331	21.86%	0.4203	0.1634	61.12%	0.4206	0.3137	25.42%
subject17	0.396	0.1761	55.53%	0.3001	0.2232	25.62%	0.6235	0.3579	42.60%	0.5109	0.198	61.24%	0.3339	0.2608	21.89%
subject18	0.4253	0.3194	24.90%	0.3645	0.2286	37.28%	0.6825	0.222	67.47%	0.4236	0.3886	8.26%	0.4936	0.3017	38.88%
subject19	0.5431	0.3059	43.68%	0.3526	0.2547	27.77%	0.4326	0.3394	21.54%	0.5632	0.3729	33.79%	0.4625	0.219	52.65%
subject20	0.3964	0.3459	12.74%	0.3265	0.2849	12.74%	0.4025	0.3938	2.16%	0.3265	0.1873	42.63%	0.3976	0.2338	41.20%
Min	0.3001	0.1403	0.37%	0.2936	0.1604	3.91%	0.2654	0.2158	2.16%	0.2354	0.1634	1.27%	0.3265	0.177	4.07%
Max	0.5431	0.3934	55.53%	0.4966	0.3795	60.12%	0.7203	0.3938	67.47%	0.5632	0.3905	61.24%	0.5986	0.4635	55.59%
Range	0.2430	0.2531	55.16%	0.2030	0.2191	56.21%	0.4549	0.1780	65.31%	0.3278	0.2271	59.97%	0.2721	0.2865	51.53%
Average	0.3902	0.2667	31.40%	0.3689	0.2643	28.14%	0.4292	0.3126	22.96%	0.4032	0.2519	36.84%	0.4288	0.3053	28.39%
STD	0.0644	0.0723	0.1695	0.0456	0.0636	0.1518	0.1223	0.0589	0.1853	0.0717	0.0890	0.2066	0.0690	0.0759	0.1485

$$\rho(A,B) = \frac{1}{\bar{N}-1}\sum_{\bar{i}=1}^{\bar{N}}(\frac{A_{\bar{i}}-\bar{\mu_A}}{\bar{\sigma_A}})(\frac{B_{\bar{i}}-\bar{\mu_B}}{\bar{\sigma_B}}), \bar{i} = 1, 2, \ldots, \bar{N} \tag{4.1}$$

where A and B denote two EEG vector samples, each containing $\bar{N}$ elements. μ_A and σ_A denote the mean and standard deviation of A. μ_B and σ_B denote the mean and standard deviation of B. The Person correlation coefficient is positively correlated with the similarity, and both are in the range of $[0, 1]$.

We introduce two similarity concepts used in our measurement: self-similarity and cross-similarity. The self-similarity is defined by the similarity of EEG signals within the same EEG category while the cross-similarity is defined by the similarity of EEG signals of two different EEG categories. Both the self-similarity and cross-similarity are measured under two conditions: inter-class and inter-person, respectively.

Inter-class measurement. Under the inter-class situation, we measure the correlation coefficient matrix for *every specific subject* and calculate the average matrix by calculating the mean value of all the matrix. For example, there are 5 classes

for the specific subject, we calculate a $5 * 5$ correlation coefficient matrix. In this matrix, $\rho_{\breve{i},\breve{j}}$ denotes the correlation coefficient between the samples of the class $\breve{i}$ and the samples of the class $\breve{j}$. The self-similarity indicates the similarity between two different samples from the same class. The cross-similarity indicates the average of similarity of each possible *class pair* of samples belonging to the *specific subject*.

Inter-person measurement. Under the inter-person situation, we measure the correlation coefficients matrix for *every specific* class and then calculate the average matrix. The self-similarity indicates the similarity between two different samples from the same class of the same subject. The cross-similarity denotes the average of similarity of each possible *subject pair* of samples belonging to the *specific class*.

Table 4.1 shows the inter-class correlation coefficient matrix and the corresponding statistical self- and cross-similarity. The last column (PD) denotes the Percentage Difference between the self-similarity and cross-similarity. We can observe from the results that the self-similarity is always higher than the cross-similarity for all classes, meaning that the samples' intra-class cohesion is stronger than the inter-class cohesion. The percentage difference has a noticeable fluctuation, indicating the varying intra-class cohesion over different class pairs. *Class 1* is easier to be distinguished due to its highest percentage difference, while in contrast, *class 0* and *class 4* are difficult to be accurately classified.

Similarly, Table 4.2 shows the inter-person correlation coefficient matrix and gives an alternative visualization of the results. Again, we find that, for each class, the self-similarity is higher than cross-similarity with varying percentage difference. The standard deviations of cross-similarity in the five classes are similar. This indicates the steady and even distribution of the dataset between different subjects and different classes.

The above analysis results basically satisfy our following hypothesis for multi-person multi-class classification: 1) the self-similarity is consistently higher than cross-similarity both under inter-class and inter-person conditions; 2) the higher inter-class

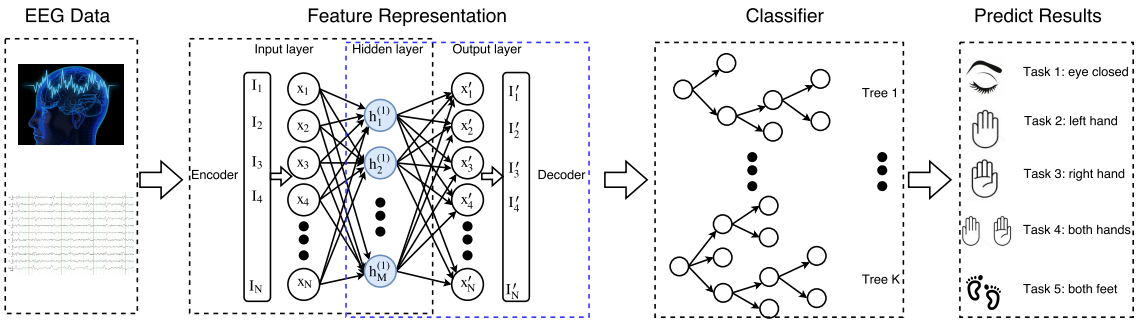


Figure 4.1: The methodology flowchart. The collected EEG data flow into the Feature Representation component to seek for the appropriately representation and interpretation. I_i and I'_i separately indicate the input and output EEG data. x_i, h_i, and x'_i indicate the neurons in the input layer, the hidden layer and the output layer, respectively. The learned feature representation h will be sent to an XGBoost classifier with K trees. The classifier's predict result is corresponding to the user's intent, which indicates the user's intention such as closing eye, moving left hand or moving right hand.

percentage difference, the better classification results; 3) lower average percentage differences and standard deviations of the subjects result in the better classification performance under the inter-person condition.

4.3 Methodology

In this section, we review the algorithm by first normalizing the input EEG data and then automatically explore the feature representation of the normalized data. At last, we adopt the XGBoost classifier to classify the trained features. The methodology flowchart is shown in Figure 4.1.

4.3.1 Normalization

Normalization plays a crucial role in a knowledge discovery process for handling different units and scales of features. For instance, given one input feature ranges

from 0 to 1 while another ranges from 0 to 100, the analysis results will be dominated by the latter feature. Generally, there are three widely used normalization methods: Min-Max Normalization, Unity Normalization, and Z-score Scaling (also called standardization).

Min-Max Normalization. Min-Max Normalization projects all the elements in an vector to the range of $[0, 1]$. This method maps features to the same range despite of their original means and standard deviations. The formula of Min-Max normalization is given below:

$$x_{new} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{4.2}$$

where x_{min} and x_{max} separately denotes the minimum and maximum in the feature x.

Unity Normalization. Unity Normalization re-scales the features by the percentage or the weight of each single element. It calculates the sum of all the elements and then divides each element by the sum. The equation is:

$$x_{new} = \frac{x}{\sum x} \tag{4.3}$$

where $\sum x$ denotes the sum of feature x. Similar to Min-Max Normalization, the results of this method also belong to the range of $[0, 1]$.

Z-score Scaling. Z-score Scaling forces features under normal Gaussian distribution (zero mean and unit variance), using the equation below:

$$x_{new} = \frac{x - \mu}{\sigma} \tag{4.4}$$

where μ denotes the expectation of feature x and σ denotes the standard deviation.

Depending on the feature characteristics of datasets, these 3 categories of normalization methods may lead to differed analysis results.

4.3.2 Feature Representation

To exploit the deeper correlationship between EEG signals, we adopt Autoencoder to have a better representation of EEG. The Autoencoder [259] is an unsupervised machine learning algorithm that aims to explore a lower-dimensional representation of high-dimensional input data for *dimensionality reduction*. In structure, Autoencoder is a multi-layer back propagation neural network that contains three types of layers: the input layer, the hidden layer, and the output layer. The procedure from the input layer to the hidden layer is called *encoder* while the procedure from the hidden layer to the output layer is called *decoder*. Both the encoder and the decoder yield a set of weights W and biases b. Autoencoder is called either *Basic Autoencoder* when there is only one hidden layer or *Stacked Autoencoder* when there are more than one hidden layers. Based on our prior experiment experience, basic Autoencoder works better than stacked Autoencoder when dealing with EEG signals. Therefore, we adopt the basic Autoencoder structure.

Let $X = \{X_i | i = 1, 2 \cdots, N\}, X \in \mathbb{R}^N, X_i \in \mathbb{R}^d$ be the entire training data (unlabeled), where X_i denotes the i-th sample, N denotes the number of training samples, and d denotes the number of elements in each sample. $h_i = \{h_{ij} | j = 1, 2, \cdots, M\}, h_i \in \mathbb{R}^M$ represents the learned feature in the hidden layer for the i-th sample, where M denotes the number of neural units in current layer (the number of elements in h_i). For simplicity, we use x and h to represent the input data and the data in the hidden layer, respectively. First, the encoder transforms the input data x to the corresponding representation h by the encoder weights W_{en} and the encoder biases b_{en}:

$$h = W_{en}x + b_{en} \tag{4.5}$$

Then, the decoder transforms the hidden layer data h to the output layer data x' by the decoder weights W_{de} and the decoder biases b_{de}:

$$x' = W_{de}h + b_{de} \tag{4.6}$$

The function of the decoder is to reconstruct the encoded feature h and make the

reconstructed data x' as similar to the input data x as possible. The discrepancy between x and x' is calculated by the MSE (mean squared error) cost function which is optimized by the RMSPropOptimizer.

In summary, training Autoencoder is the task of optimizing the parameters to achieve the minimum cost between the input x and the reconstructed data x'. At last, the hidden layer data h would contain the refined information. Such information can be regarded as representation of the input data, which is also the final outcome of Autoencoder. In above formulation, the dimension of the input data x and the refined feature (the hidden layer data h) are d and M, respectively. The function of the Autoencoder is either dimensionality reduction if $d > M$ or dimensionality ascent if $d < M$.

4.3.3 Intent Recognition

To recognize the intent based on the represented feature, in this section, we employ the XGBoost [260] classifier. XGBoost, also known as *Extreme Gradient Boosting*, is a supervised scalable tree boosting algorithm derived from the concept of *Gradient Boosting Machine* [261]. Compared with gradient boosting algorithm, XGBoost proposes a more regularized model formalization to prevent over-fitting, with the engineering goal of pushing the limit of computation resources for boosted tree algorithms to achieve better performance.

Consider n sample pairs $D = \{(x_{i'}, y_{i'})\}$, $(|D| = n, x_{i'} \in \mathbb{R}^m, y_{i'} \in \mathbb{R})$ where $x_{i'}$ denotes a m-dimensional sample and $y_{i'}$ denotes the corresponding label. XGBoost aims to predict the label $\tilde{y_{i'}}$ of every given sample $x_{i'}$.

The XGBoost model is the ensemble of a set of classification and regression trees (CART), each having its leaves and corresponding scores. The finial results of tree ensemble is the sum of all the individual trees. For a tree ensemble model of K'

trees, the predict output is:

$$\tilde{y_{i'}} = \sum_{k'=1}^{K} f_{k'}(x_{i'}), f_{k'} \in F \tag{4.7}$$

where F is the space of all trees and $f_{k'}$ denotes a single tree.

The objective function of XGBoost includes loss function and regularization. The loss function evaluates the difference between each ground truth label $y_{i'}$ and the predict result $\tilde{y_{i'}}$. It can be chosen based on various conditions such as cross-entropy, logistic, and mean square error. The regularization part is the most outstanding contribution of XGBoost. It calculates the complexity of the model and a more complex structure brings larger penalty.

The objective function is defined as:

$$\Psi = \sum_{i}^{n} l(\tilde{y_{i'}}, y_{i'}) + \sum_{k'}^{K} \Omega(f_{k'}) \tag{4.8}$$

where $l(\tilde{y_{i'}}, y_{i'})$ is the loss function and $\sum_{k'} \Omega(f_{k'})$ is the regularization term. The complexity of a single tree is calculated as

$$\Omega(f_{k'}) = \gamma T + \frac{1}{2}\lambda \|\omega\|^2 \tag{4.9}$$

where T is the number of leaves in the tree, $\|\omega\|^2$ denotes the square of the L2-norm of the weights in the tree, γ and λ are the coefficients. The regularized objective helps deliver a model of simple structure and predictive functions. More specifically, the first term, Ω, penalizes complex structures of the tree (fewer leaves lead to a smaller Ω), while the second term penalizes the overweights of individual trees in case of the overbalanced trees dominating the model. Moreover, the second term helps smooth the learned weights to avoid overfitting.

4.4 Experiment

In this section, we evaluate the proposed approach on a public EEG dataset and report the results of our experimental studies. Firstly, we introduce the experimental

setting and evaluation criterion. Then, we provide the classification results, followed by the analysis of influencing factors (e.g., normalization method, training data size, and neuron size in the Autoencoder hidden layer). Additional experiments are conducted to study the efficiency and robustness by comparing our approach with the state-of-the-art methods.

4.4.1 Experimental Setting

We adopt the EEGMMIDB dataset as described in Section 3.3.1. There are totally five tasks and each task is labeled as one class (from 0 to 4).

4.4.2 Evaluation

Basic definitions related to classification problems include:

- True Positive (TP): the ground truth is positive and the prediction is positive;
- False Negative (FN): the ground truth is positive but the prediction is negative;
- True Negative (TN): the ground truth is negative and the prediction is negative;
- False Positive (FP): the ground truth is negative but the prediction is positive;

Based on these concepts, we define criteria to evaluate the performance of the classification results as follows:

Accuracy. The proportion of all correctly predicted samples. Accuracy is a measure of how good a model is.

$$accuracy = \frac{TP + FN}{FP + FN + TP + TN} \tag{4.10}$$

The *test error* refers to the incorrectly predicted samples' proportion, which equals to 1 minus accuracy.

Precision. The proportion of all positive predictions that are correctly predicted.

$$Precision = \frac{TP}{TP + FP} \tag{4.11}$$

Recall. The proportion of all real positive observations that are correctly predicted.

$$Recall = \frac{TP}{TP + FN} \tag{4.12}$$

F1 Score. A 'weighted average' of precision and recall. The higher F1 score, the better the classification performance.

$$F1\ Score = 2\frac{precision * recall}{precision + recall} \tag{4.13}$$

ROC. The ROC (Receiver Operating Characteristic) curve describes the relationship between TPR (True Positive Rate) and FPR (False Positive Rate) at various threshold settings.

AUC. The AUC (Area Under the Curve) represents the area under the ROC curve. The value of AUC drops in the range $[0.5, 1]$. The higher the AUC, the better the classifier.

4.4.3 Experiments and Results

In our experiments, the Autoencoder model is trained by the training dataset then the testing dataset is fed into the trained Autoencoder model for feature extraction. The extracted features of the training dataset are used by the XGBoost classifier, which will be evaluated by the features of the testing dataset. The number of neurons in the input and output layers in the Autoencoder model fixed at 64 (the input EEG data contains 64 dimensions), and the learning rate is set as 0.01. Parameter tuning

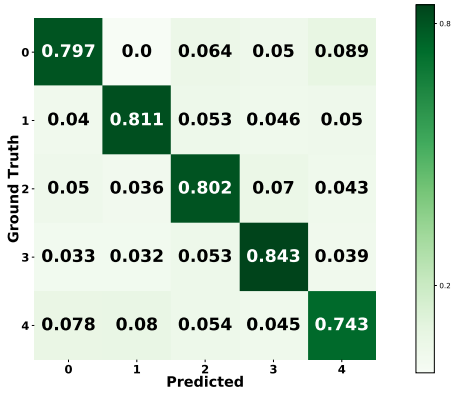


Figure 4.2: Confusion Matrix

Labels	Precision	Recall	F-1	AUC
0	0.7973	0.7703	0.7836	0.9454
1	0.8108	0.9219	0.8628	0.9572
2	0.8017	0.7556	0.7780	0.9492
3	0.8429	0.7294	0.7820	0.9506
4	0.7427	0.7279	0.7352	0.9258

Table 4.3: Evaluation

experience shows that Autoencoder performs better with more hidden layer neurons. For XGBoost, we set the objective function as *softmax* for multi-class classification through pre-experiment experience. The parameters of XGBoost are selected based on the parameters tuning document[1]. More specifically, we set the learning rate $\eta = 0.7$, the parameter related to the minimum loss reduction and the number of leaves $gamma = 0$, the maximum depth of a tree $max_{depth} = 6$ (too large max_{depth} may lead to overfitting), the subsampling ratio of training instance $subsample = 0.9$ (to prevent overfitting), and $num_{class} = 5$, since we have samples of 5 categories. All the other parameters are set as default value. Without specific explanation, all the Autoencoder and XGBoost classifiers are taking above parameters setting.

The hardware used in experiments is a GPU-accelerated machine with Nvidia Titan X pascal GPU, 768G memory, and 1.45TB PCIe based SSD. The training time is listed in related experiments, respectively.

Multi-person Multi-class EEG Classification. To evaluate the proposed approach, 560,000 EEG sample patches are utilized in this experiment. Each sample patch contains a feature vector of 64 dimensions and a ground truth label. The raw EEG data is normalized by the z-score scaling method and then randomly split into training dataset (532,000 samples) and testing dataset (28,000 samples). The rep-

[1] `https://github.com/dmlc/xgboost/blob/master/doc/parameter.md`

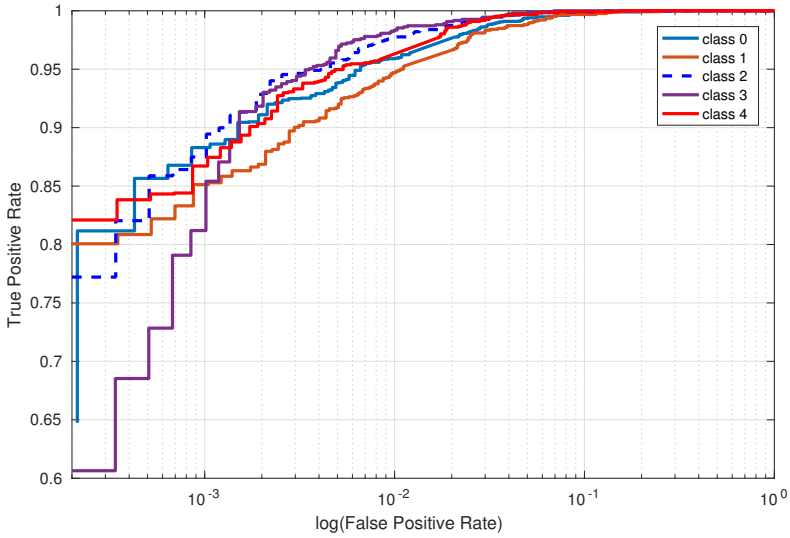


Figure 4.3: ROC curve for 5-class classification by XGBoost. Five curves separately indicate the ROC curve of five classes. The dotted diagonal line denotes the random classifier where TPR=FPR. The closer the ROC curve to the upper left corner, the better performance the classifier has. It is clear to notice that the class 1 has the best classification performance.

resentative features are extracted by Autoencoder with 121 hidden neurons and are input to the XGBoot classifier. The confusion matrix and evaluation of the results are listed in Figure 4.2 and Table 4.3. The classification accuracy of 28,000 testing samples (from 20 subjects and belong to 5 classes) is 0.794. The average precision, recall, F1 score, and AUC are 0.7991, 0.781, 0.7883, and 0.9456, respectively. Among the evaluation standards, the *class 1* obtains the highest precision, recall, F1 score, and AUC. This means that the *class 1* samples have the most obvious divergence and are most distinguishable. On the contrary, *class 4* is most confusing. This conclusion is highly consistent with our similarity analysis results in Section 4.2. From the ROC curve, shown as Figure 4.3, we can deduce to the same conclusion. All the classes achieved the AUC higher than 0.92, indicating that the classifier is steady and of high quality, according to the characteristics of AUC mentioned in Section 4.4.2.

Effect of Normalization Method. The Autoencoder component regards the input

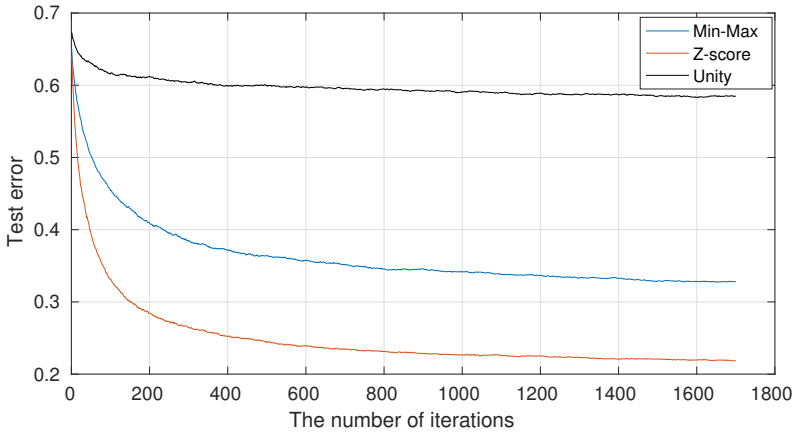


Figure 4.4: The effect of normalization method. The three test error curves denote Min-Max, Z-score, and Unity normalization method, respectively.

data as the training target and calculates the discrepancy between them for error back propagation. This character of Autoencoder determines that the feature extraction quality and training cost are affected by the amplitude of the input data. In data pre-processing stage, the data values are directly related to the normalization method.

To explore the impact of the normalization method, 560,000 EEG samples from 20 subjects are randomly split into a training dataset of 532,000 samples (95% proportion) and a testing dataset of 28,000 samples (5% proportion). By setting 121 neurons for the hidden layer of Autoencoder, the XGBoost test error under three kinds of normalization methods is shown in Figure 4.4. The figure shows that the z-score scaling normalization earns the lowest test error while the unity normalization obtains the highest test error. All the curves trend to convergence after 1,600 iterations. Without specific explanation, all the remaining experiments in this chapter use the z-score scaling method.

Effect of Training Data Size. We explore in this section the relationship between the classification performance and the training data size. We design five experiments with the training data proportion of 60%, 70%, 80%, 90%, and 95%, respectively. Each experiment is repeated 5 times and the test error's error bar is shown in Fig-

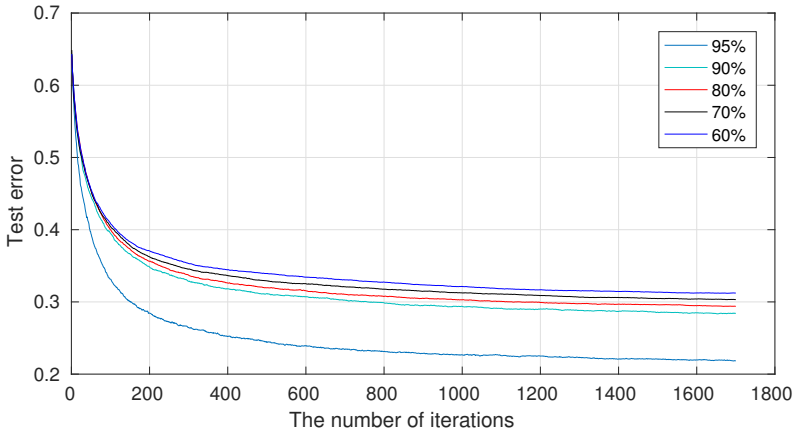


Figure 4.5: The relationships between test error and the iterations under various training data proportions

Table 4.4: Comparison of various classification methods. The first nine groups investigate the proper EEG data classifier and the last 7 groups illustrate the most efficient feature representation method.

No.	**1**	**2**	**3**	**4**
Classifier	SVM	RNN	LDA	RNN+SVM
Acc	0.3333	0.6104	0.3384	0.6134
No.	**5**	**6**	**7**	**8**
Classifier	CNN	DT	AdaBoost	RF
Acc	0.5729	0.3345	0.3533	0.6805
No.	**9**	**10**	**11**	**12**
Classifier	XGBoost	PCA+XGBoost	PCA+AE+XGBoost	EIG+AE+XGBoost
Acc	0.7453	0.7902	0.6717	0.5125
No.	**13**	**14**	**15**	**16**
Classifier	EIG+PCA+XGBoost	DWT+XGBoost	Stacked AE+XGBoost	AE+XGBoost
Acc	0.6937	0.7221	0.7048	**0.794**

ure 4.6. The training time is positively correlated with the training data proportion. The test error arrives at the lowest point 0.206, with an acceptable training time, while the proportion is 95%. All the following experiments will take 95% proportion.

The relationships between test error and the iterations under various training data proportions are shown in Figure 4.5. All the curves trend to convergence after 1,600 iterations and the higher proportion leads to lower test error.

Effect of Neuron Size in Autoencoder Hidden Layer. The neuron size in the hidden layer of Autoencoder indicates the number of dimensions of the extracted

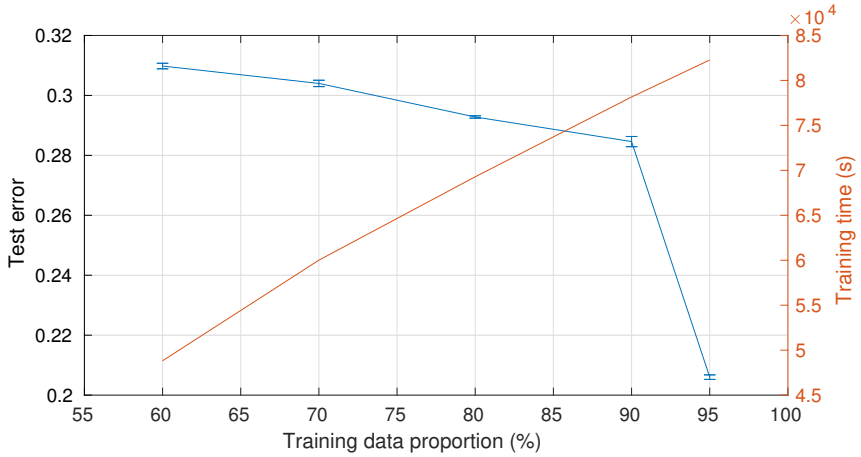


Figure 4.6: The relationship between the test error with error bars, the training time and the training data proportion

features. Thus it has great impact on the quality of feature extraction as well as the classification results. We design the experiment with the neuron size ranges from 30 to 200 and the experimental results (the test error and the training time) are shown in Figure 4.7.

In the first stage (0-120), the test error keeps decreasing with the increase of the neuron size; in the second stage (larger than 120), the test error stands at around 0.21 with slight fluctuation. The training time curve has a linear relationship with the neuron size on the whole. Although the gap between the test error curve and the training time curve arrives at the minimum around 100 neurons, the test error is still high. The test error reaches the bottom while the neuron size is 121, and the training time is acceptable at this point. Moreover, the test error curve keeps steady after 121. We set the hidden layer neuron size for all other experiments as 121.

4.4.4 Overall Comparison

In our approach, we employ XGBoost as the classifier to classify the refined EEG features yielded by Autoencoder. To demonstrate the efficiency of this method, in this

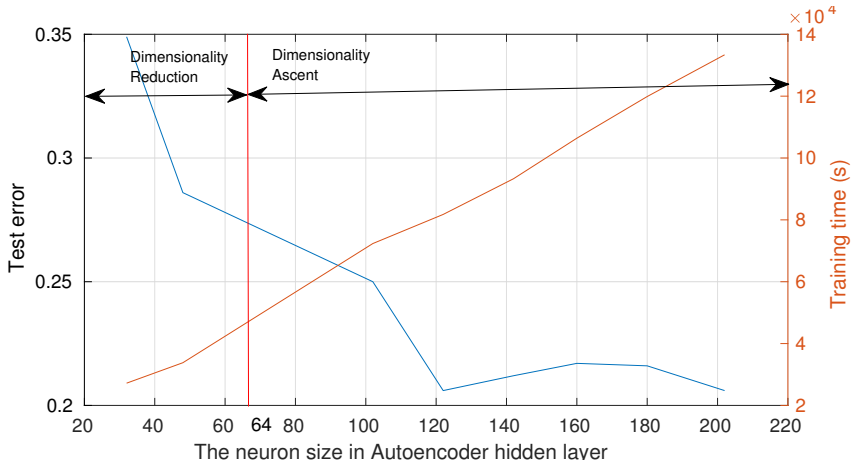


Figure 4.7: The effect of neuron size in Autoencoder hidden layer. Since the input data is 64-dimension (marked as red line), the left part (smaller than 64) is dimensionality reduction area while the right part (larger than 64) is dimensionality ascent area.

section, we compare the proposed approach with several widely used classification methods. All the classifiers work on the same EEG dataset and their corresponding performance is listed in Table 4.4.

In Table 4.4, EIG denotes the eigenvector-based dimensionality reduction method used in Eigenface recognition[2]; PCA denotes Principal Components Analysis which

[2]http://www.vision.jhu.edu/teaching/vision08/Handouts/case_study_pca1.pdf

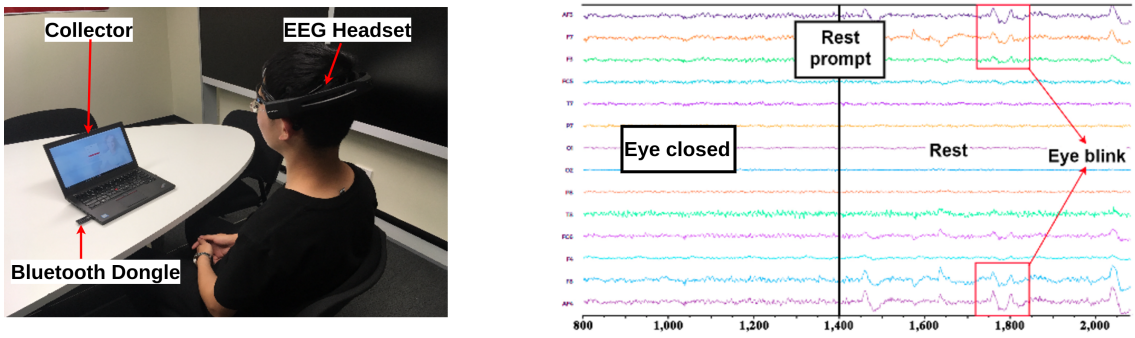


Figure 4.8: EEG collection and the raw data. The pure EEG data is selected for recognition and the data, which is contaminated by eye blink and other noise, is not included in the local dataset (dropped).

is a commonly used dimensionality reduction method; DWT denotes Discrete Wavelet Transform, which is the wavelet transformation with the wavelets discretely sampled. The stacked Autoencoder contains 3 hidden layers with 100, 121, 100 neurons, respectively.

The comparison is divided into two aspects: the classifier and the feature representation method. At first, we classify our dataset separately by 9 commonly used sensing data classifier (e.g., SVM, RF, RNN, and CNN) to investigate the most suitable classifier for raw EEG data. Then 7 categories feature extraction method (e.g., PCA, AE, and DWT) are conducted to investigate the most appropriately EEG feature representation approach. The comparison results show that the XGBoost classifier outperforms its counterpart (without pre-processing and feature extraction) and obtains the accuracy of 0.74, which means that XGBoost is more suitable to solve this problem. On the other hand, some feature extraction is positive to the classification whilst some are negative. Through the comparison, we find that Autoencoder (121 hidden neurons) achieves the highest multi-person classification accuracy as **0.794**.

4.5 Case study

In this section, to further demonstrate the feasibility of the proposed approach, we conduct a local experiment and present the classification result.

4.5.1 Experimental Setting

This experiment is carried on by 5 subjects (3 males and 2 females) aged from 24 to 30. During the experiment, the subject wearing the *Emotiv Epoc+*[3] EEG

[3]`https://www.emotiv.com/product/emotiv-epoc-14-channel-mobile-eeg/`

Table 4.5: Mark in experiment and corresponding intent and label in case study

Mark	**up arrow**	**down arrow**	**left arrow**	**right arrow**	**central cycle**	**close eye**
Intent	upward	downward	leftward	rightward	center	relax
Label	0	1	2	3	4	5

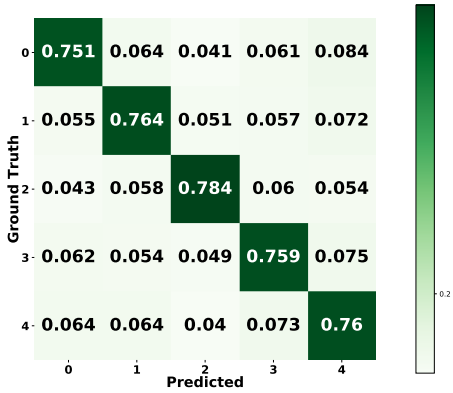


Figure 4.9: Confusion Matrix

Labels	**Precision**	**Recall**	**F-1**	**AUC**
0	0.7369	0.7555	0.7461	0.8552
1	0.7448	0.7407	0.7428	0.8395
2	0.7652	0.7930	0.7788	0.8931
3	0.7374	0.7230	0.7301	0.8695
4	0.7401	0.6986	0.7187	0.8759
5	0.7765	0.8062	0.7911	0.9125

Table 4.6: Evaluation

collection headset, facing the computer screen and focus on the corresponding *mark* which appears on the screen (shown in Figure 4.8). The Emotiv Epoc+ contains 14 channels and the sampling rate is set as 128 Hz. The marks are shown on the screen and the corresponding brain activities and labels used in this chapter are listed in Table 4.5. Summarily, this experiment contains 172,800 samples with 34,560 samples for each subject.

4.5.2 Recognition Results and Comparison

The dataset is divided into a training set (155,520 samples) and a testing set (17,280 samples). There are 9 mini-batches and the batch size is 17,280. All the other parameters are the same as listed in Section 4.4. The proposed approach achieves the 6-class classification accuracy of **0.7485**. The confusion matrix and evaluation are reported in Figure 4.9 Table 4.6. Clearly, the 5th class intent (eye closed and keep relax) has the highest precision and is the easiest activity to be recognized.

Subsequently, to demonstrate the efficiency of the proposed approach, we compare

Table 4.7: Comparison of various classification methods over the case study dataset

No.	**1**	**2**	**3**	**4**
Classifier	SVM	RNN	LDA	RNN+SVM
Acc	0 0.356	0.675	0.343	0.6312
No.	**5**	**6**	**7**	**8**
Classifier	CNN	DT	AdaBoost	RF
Acc	0.5291	0.305	0.336	0.6609
No.	**9**	**10**	**11**	**12**
Classifier	XGBoost	PCA+XGBoost	PCA+AE+XGBoost	EIG+AE+XGBoost
Acc	0.6913	0.7225	0.6045	0.4951
No.	**13**	**14**	**15**	**16**
Classifier	EIG+PCA+XGBoost	DWT+XGBoost	Stacked AE+XGBoost	AE+XGBoost
Acc	0.6249	0.6703	0.6593	**0.7485**

our method with the state-of-the-art methods and the results are shown in Table 4.7.

4.6 Conclusion

In this chapter, we have focused on multi-class EEG signal classification based on EEG data that come from different subjects (multi-person). To achieve this goal, we aim at discovering the patterns in the discrepancy between different EEG classes with robustness over the difference between various subjects. Firstly, we analyze three widely used normalization methods in pre-processing stage. Then, we feed the normalized EEG data into the Autoencoder and train the Autoencoder model. Autoencoder transforms the original 64-dimension features to 121-dimension features and essentially maps the data to a new feature space when meaningful features play a dominating role. Finally, we evaluate our approach over an EEG dataset of 560,000 samples belonging to 5 categories and achieve the accuracy of 0.794. Compared with the accuracy of around 0.34 achieved by traditional methods (e.g., SVM, AdaBoost, Decision Tree, and RNN), our results 0.794 show significant improvement. Furthermore, we explore the effect of two factors (the training data size and the neuron size in Autoencoder hidden layer) on the training results. At last, we conduct a case study to gather 6 categories of brain activities and obtain the classification accuracy of 0.7485.

Part III

Adaptive Representation Learning

4. Multi-Person Brain Activity Recognition

This part contains works published in:

[4] **X. Zhang**, L. Yao, C. Huang, S. Wang, M. Tan, G. Long, and C. Wang. Adaptive Sensor Data Classification with Selective Attention. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (**IJCAI 2018**)* (pp. 3111-3117), 2018. (CORE Rank A*)

[5] **X. Zhang**, L. Yao, X. Wang, W. Zhang, S. Zhang, and Y. Liu. Know Your Mind: Adaptive Brain Signal Classification with Reinforced Attentive Convolutional Neural Networks. In *Proceedings of the 19th IEEE International Conference on Data Mining (**ICDM 2019**)*. Beijing, China, November 8-11, 2019. (CORE Rank A*)

Chapter 5

Adaptive Sensory Data Classification with Selective Attention

In some cases, human intention could be better predicted by combing brain signals and other sensor signals (such as activity and location information). Nowadays, diverse categories of sensors can be found in various wearable devices. Such devices are now being widely applied in multiple fields, such as Internet of Things [262, 9]. As a result, massive multimodal sensor data are being produced continuously. The question that how we can deal with these data efficiently and effectively has become a major concern. Addressing this issue, in this thesis, we adopt adaptive representation learning which can adaptively capture the most informative features based on the characteristics of the input signals.

Compared to images and videos, sensory data are naturally formed as a 1-D signal, with each element representing one sensor channel accordingly. There are several challenges for such sensor data classification. First, most existing classification methods use domain-specific knowledge and thus may become ineffective or even fail in complex situations where multimodal sensory data are being collected [263]. For example, one approach that works well on IMU (Inertial Measurement Unit) signals

may not be able to deal with EEG (Electroencephalography) brain signals. Therefore, an effective and universal sensor data classification method is highly desirable for complex situations. This framework is expected to have both efficiency and robustness over various sensor signals.

Second, the wearable sensor data carries far less information than texts and images. For example, a sample signal gathered by a 64-channel EEG equipment only contains 64 numerical elements. Hence, a more effective classifier is required to extract discriminative information from such limited raw data. However, maximizing the utilization of the given scarce data demands cautious preprocessing and a rich fund of domain knowledge.

Inspired by attention mechanism [264], we propose to concentrate on an attention zone of the signal to automatically learn the informative attention patterns for different sensor combinations. Here, the attention zone is a selection block of the signal with a certain length, sliding over the feature dimensions. Note that reinforcement learning has been shown to be capable of learning human-control level policy on a variety of tasks [265]. Then we exploit the reinforcement learning to discover the attention zone. Moreover, considering that the signals in different categories may have different inter-dimension dependency [266], we propose to use the LSTM (Long Short-Term Memory [267, 268]) to exploit the latent correlation between signal dimensions. We propose a weighted average spatial LSTM (WAS-LSTM) classifier by exploring the dependency in sensor data.

The main contributions of this chapter are as follows:

- We propose a selective attention mechanism for sensor data classification using the spatial information only. The proposed method is insensitive to sensor types since it is capable of handling multimodal sensor data.
- We apply deep reinforcement learning to automatically select the most distinguishable features, called attention zone, for multimodal sensor data of different sensor types and combinations. We design a novel objective function as the

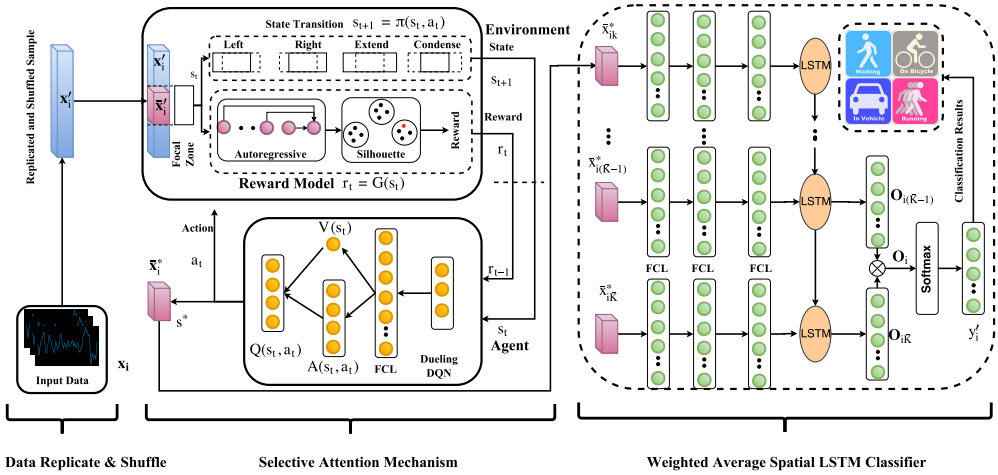


Figure 5.1: Flowchart of the proposed approach. The attention zone $\bar{\mathbf{x}}_i$ is a selected fragment from $\mathbf{x}'_i$ to feed in the state transition and the reward model. In each step t, one action is selected by the state transition to update s_t based on the agent's feedback. The reward model evaluates the quality of the attention zone to the reward r_t. The dueling DQN is employed to find the optimal attention zone $\bar{\mathbf{x}}^*_i$ which will be feed into the LSTM based classifier to explore the inter-dimension dependency and predict the sample's label y'_i. FCL denotes Fully Connected Layer. The State Transition contains four actions: left shifting, right shifting, extend, and condense.

award in reinforcement learning task to optimize the attention zone. The new reward model saves more than 98% training time of the deep reinforcement learning.

- We propose Weighted Average Spatial LSTM classifier to capture the cross-dimensional dependency in multimodal sensor data.

5.1 Methodology

Suppose the input sensor data can be denoted by $\mathbf{X} = \{(\mathbf{x}_i, y_i), i = 1, 2, \cdots I\}$ where $(\mathbf{x}_i, y_i)$ denotes the 1-D sensor signal, called one *sample*, and I denotes the number of samples. In each sample, the feature $\mathbf{x}_i \in \mathbb{R}^K$ contains K elements and the corresponding ground truth $y_i \in \mathbb{R}$ is an integer denotes the sample's category. $\mathbf{x}_i$ can be described as a vector with K elements, $\mathbf{x}_i = \{x_{ik}, k = 1, 2, \cdots, K\}$.

The proposed algorithm is shown in Figure 5.1. The main focus of the algorithm is to exploit the latent dependency between different signal dimensions. To this end, the proposed approach contains several components: 1) the replicate and shuffle processing; 2) the selective attention learning; 3) the sequential LSTM-based classification. In the following, we will first discuss the motivations of the proposed method and then introduce the aforementioned components in details.

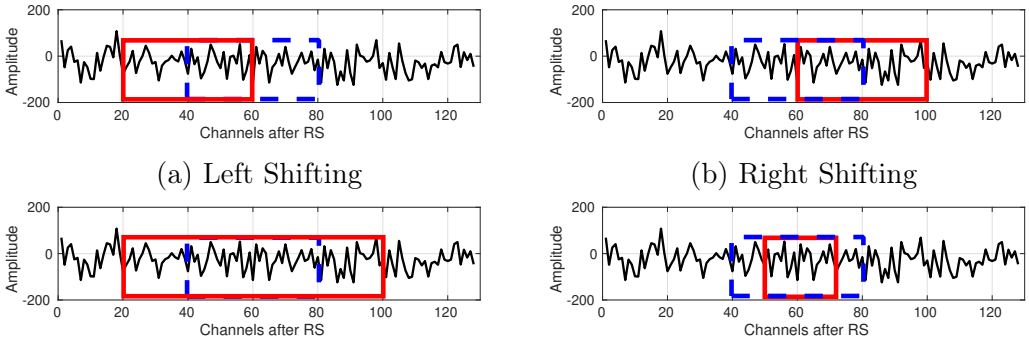
5.1.1 Motivation

How to exploit the latent relationship between sensor signal dimensions is the main focus of the proposed approach. The signals belonging to different categories are supposed to have different inter-dimension dependent relationships which contain rich and discriminative information. This information is critical to improve the distinctive signal pattern discovery.

In practice, the sensor signal is often arranged as 1-D vector, which is less informative for the limited and fixed element arrangement. The elements order and the number of elements in each signal vector can affect the element dependency. In many real-world scenarios, the multimodal sensor data are associated with the practical placement. For example, the EEG data are concatenated following the distribution of biomedical EEG channels. Unfortunately, the practical sensor sequence, with the fixed order and number, may not be suitable for inter-dimension dependency analysis. Meanwhile, the optimal dimension sequence [269] varies with the sensor types and combinations. Therefore, we propose the following three techniques to amend these drawbacks.

First, we replicate and shuffle the input sensor signal vector on dimension-wise in order to provide as much latent dependency as possible among feature dimensions (Section 5.1.2).

Second, we introduce an attention zone as a selective attention mechanism, where



(a) Left Shifting (b) Right Shifting

(c) Extend Shifting (d) Condense Shifting

Figure 5.2: Four actions in the state transition: left shifting, right shifting, extend, and condense. The dashed line indicates the attention zone before the action while the solid line indicates after the action.

the optimal inter-dimension dependency for each sample only depends on a small subset of features. Here, the attention zone is optimized by deep reinforcement learning which has been proved to be stable and well-performed in policy learning (Section 5.1.3).

Third, we propose the WAS-LSTM classifier by extracting the distinctive inter-dimension dependency (Section 5.1.4).

5.1.2 Data Replicate and Shuffle

To provide as much as possible information, we design an approach to exploit the spatial relationships among EEG signals. The signals belonging to different brain activities are supposed to have different spatial dependent relationships. We replicate and shuffle the input EEG signals on dimension-wise. Within this method, all the possible dimension arrangements have the equiprobable appearance.

Suppose the input raw EEG data are denoted by $\mathbf{X} = \{(\mathbf{x}_i, y_i), i = 1, 2, \cdots I\}$, where $(\mathbf{x}_i, y_i)$ denotes a single EEG sample and I denotes the number of samples. In each sample, the feature $\mathbf{x}_i = \{x_{ik}, k = 1, 2, \cdots, K\}, \mathbf{x}_i \in \mathbb{R}^K$ contains K elements corresponding to K EEG channels and $y_i \in \mathbb{R}$ denotes the corresponding label. x_{ik}

denotes the k-th dimension value in the i-th sample.

In real-world collection scenarios, the EEG data are generally concatenated following the distribution of biomedical EEG channels. However, the biomedical dimension order may not present the best spatial dependency. The exhausting method is too computationally expensive to exhaust all the possible dimension arrangements. For example, a 64-channel EEG sample has $A_{64}^{64} = 1.28 \times 10^{89}$ combinations, which is an astronomical figure.

To provide more potential dimension combinations, we propose a method called Replicate and Shuffle (RS). RS is a two-step mapping method which maps $\mathbf{x}_i$ to a higher dimensional space $\mathbf{x}'_i$ with complete element combinations:

$$\mathbf{x}_i \in \mathbb{R}^K \rightarrow \mathbf{x}'_i \in \mathbb{R}^{K'}, K' > K \tag{5.1}$$

In the first step (Replicate), replicating $\mathbf{x}_i$ for $h = K'/K + 1$ times. Then, we get a new vector with length as $h * K$ which is not less than K'; in the second step (Shuffle), we randomly shuffle the replicated vector in the first step and intercept the first K' elements to generate $\mathbf{x}'_i$. Theoretically, compared with $\mathbf{x}_i$, $\mathbf{x}'_i$ contains more diverse dimension combinations. Note, this RS operation only be performed once for a specific input dataset in order to provide a stable environment for the following reinforcement learning.

5.1.3 Selective Attention Mechanism

Inspired by the fact that the optimal spatial relationship only depends on a subset of feature dimensions, we introduce an attention zone to focus on a fragment of feature dimensions. Here, the attention zone is optimized by deep reinforcement learning, which has been proved to be stable and well-performed in policy learning.

In particular, we aim to detect the optimal dimension combination, which includes the most distinctive spatial dependency among EEG signals. Since K', the length of

$\mathbf{x}'_i$, is too large and computationally expensive, to balance the length and the information content, we introduce the attention mechanism [264] since its effectiveness has been demonstrated in recent research areas such as speech recognition [270]. We attempt to emphasize the informative fragment in $\mathbf{x}'_i$ and denote the fragment by $\bar{\mathbf{x}}_i$, which is called *attention zone*. Let $\bar{\mathbf{x}}_i \in \mathbb{R}^{\bar{K}}$ and $\bar{K}$ denote the length of the attention zone which is automatically learned by the proposed algorithm. We employ deep reinforcement learning to discover the best attention zone [265].

As shown in Figure 5.1, the detection of the best attention zone includes two key components: the environment (including state transition and reward model) and the agent. Three elements (the state s, the action a, and the reward r) are exchanged in the interaction between the environment and the agent. All of the three elements are customized based on our context in this study. Next, we introduce the design of the crucial components of our deep reinforcement learning structure:

- The **state** $\mathcal{S} = \{s_t, t = 0, 1, \cdots, T\}, s_t \in \mathbb{R}^2$ describes the position of the attention zone, where t denotes the time stamp. Since the attention zone is a shifting fragment on 1-D $\mathbf{x}'_i$, we design two parameters to define the state: $s_t = \{start^t_{idx}, end^t_{idx}\}$, where $start^t_{idx}$ and end^t_{idx} denote the start index and the end index of the attention zone[1], separately. In the training, s_0 is initialized as

$$s_0 = [(K' - \bar{K})/2, (K' + \bar{K})/2] \tag{5.2}$$

- The **action** $\mathcal{A} = \{a_t, t = 0, 1, \cdots, T\} \in \mathbb{R}^4$ describes which action the agent could choose to act on the environment. Here at time stamp t, the state transition chooses one action to implement following the agent's policy π:

$$s_{t+1} = \pi(s_t, a_t) \tag{5.3}$$

  In our case, we define four categories of actions (Figure 5.2) for the attention zone: left shifting, right shifting, extend, and condense. For each action, the

[1]For example, for a random $\mathbf{x}'_i = [3, 5, 8, 9, 2, 1, 6, 0]$, the state $\{start^t_{idx} = 2, end^t_{idx} = 5\}$ is sufficient to determine the attention zone as $[8, 9, 2, 1]$.

attention zone moves a random distance $d \in [1, d^u]$ where d^u is the upper boundary. For left shifting and right shifting actions, the attention zone shifts light-ward or right-ward with the step d; for the extend and condense actions, both $start^t_{idx}$ and end^t_{idx} are moving d. At last, if the state start index or end index is beyond the boundary, a clip operation is conducted. For example, if $start^t_{idx} = -5$ which is lower than the lower boundary 0, we clip the start index as $start^t_{idx} = 0$.

- The **reward** $\mathcal{R} = \{r_t, t = 0, 1, \cdots, T\} \in \mathbb{R}$ is calculated by the reward model, which will be detailed later. The reward model Φ:

$$r_t = \Phi(s_t) \tag{5.4}$$

receives the current state and returns an evaluation as the reward.

Reward Model. Next, we introduce in detail the design of the reward model. The purpose of the reward model is to evaluate how the current state impacts the classification performance. Intuitively, the state which leads to better classification performance should have a higher reward: $r_t = \mathcal{F}(s_t)$. We set the reward modal $\mathcal{F}$ as a combination of the convolutional mapping and classification (Section 6.2.1). Since in the practical approach optimization, the higher the accuracy is, the more difficult to increase the classification accuracy. For example, improving the accuracy on a higher level (e.g., from 90% to 100%) is much harder than on a lower level(e.g., from 50% to 60%). To encourage accuracy improvement at the higher level, we design a non-linear reward function:

$$r_t = \frac{e^{acc}}{e-1} - \beta \frac{\bar{K}}{K'} \tag{5.5}$$

where acc denotes the classification accuracy. The function contains two parts; the first part is a normalized exponential function with the exponent $acc \in [0, 1]$, this part encourages the reinforcement learning algorithm to search the better s_t which leads to a higher acc. The motivation of the exponential function is that: *the reward*

growth rate is increasing with the accuracy's increase[2]. The second part is a penalty factor for the attention zone length to keep the bar shorter and the β is the penalty coefficient.

In summary, the aim of the deep reinforcement learning is to learn the optimal attention zone $\bar{\mathbf{x}}_i^*$ which leads to the maximum reward. The selective mechanism totally iterates $N = n_e * n_s$ times where n_e and n_s denote the number of episodes and steps [271], respectively. ε-greedy method [272] is employed in the state transition, which chooses a random action with probability $1 - \varepsilon$ or an action according to the optimal Q function $argmax_{a_t \in \mathcal{A}} Q(s_t, a_t)$ with probability ε. In formula,

$$a_{t+1} = \begin{cases} argmax_{a_t \in \mathcal{A}} Q(s_t, a_t) & \varepsilon' < \varepsilon \\ \bar{a} \in \mathcal{A} & otherwise \end{cases} \tag{5.6}$$

where $\varepsilon' \in [0, 1]$ is random generated for each iteration while $\bar{a}$ is random selected in $\mathcal{A}$.

For better convergence and quicker training, the ε is gradually increasing with the iterating. The increment ε_0 follows:

$$\varepsilon_{t+1} = \varepsilon_t + \varepsilon_0 N \tag{5.7}$$

Agent Policy and Optimization. The Dueling DQN (Deep Q Networks [271]) is employed as the optimization policy $\pi(s_t, a_t)$, which is enabled to learn the state-value function efficiently. The primary reason we employ a dueling DQN to uncover the best attention zone is that it updates all the four Q values at every step while other policies only update one Q value at each step. The Q function measures the expected sum of future rewards when taking that action and following the optimal policy thereafter. In particular, for the specific step t, we have:

$$\begin{aligned} Q(s_t, a_t) &= \mathbb{E}(r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} \dots) \\ &= \sum_{n=0}^{\infty} \gamma^k r_{t+k+1} \end{aligned} \tag{5.8}$$

[2]For example, for the same accuracy increment 10%, $acc : 90\% \rightarrow 100\%$ can earn a higher reward increment than $acc : 50\% \rightarrow 60\%$.

where $\gamma \in [0, 1]$ is the decay parameter that trade-off the importance of immediate and future rewards while n denotes the number of following step. The value function $V(s_t)$ estimate the expected reward when the agent is in state s. The Q function is related to the pair (s_t, a_t) while the value function only associate with s_t.

Dueling DQN learns the Q function through the value function $V(s_t)$ and the advantage function $A(s_t, a_t)$ and combines them by the following formula

$$Q(s_t, a_t) = \theta V(s_t) + \theta' A(s_t, a_t) \tag{5.9}$$

where $\theta, \theta' \in \Theta$ are parameters in the dueling DQN network and are optimized automatically. Equation: 5.9 is unidentifiable which can be observed by the fact that we can not recover $V(s_t)$ and $A(s_t, a_t)$ uniquely with the given $Q(s_t, a_t)$. To address this issue, we can force the advantage function equals to zero at the chosen action. That is, we let the network implement the forward mapping:

$$Q(s_t, a_t) = V(s_t) + [A(s_t, a_t) - \max_{a_{t+1} \in \mathcal{A}} (A(s_t, a_{t+1}))] \tag{5.10}$$

Therefore, for the specific action $a*$, if

$$argmax_{a_{t+1} \in \mathcal{A}} Q(s_t, a_{t+1}) = argmax_{a_{t+1} \in \mathcal{A}} A(s_t, a_{t+1}) \tag{5.11}$$

then we have

$$Q(s_{t+1}, a*) = V(s_t) \tag{5.12}$$

Thus, as shown in the Figure 6.2 (the second last layer of the agent part), the stream $V(s_t)$ is forced to learn an estimation of the value function, while the other stream produces an estimation of the advantage function.

To assess the Q function, we optimize the following cost function at the i-th iteration:

$$\begin{aligned} L_i(\Theta_i) &= \mathbb{E}_{s_t, a_t, r_t, s_{t+1}}[(\bar{y}_i - Q(s_t, a_t))^2] \\ &= \mathbb{E}_{s_t, a_t, r_t, s_{t+1}}[(\bar{y}_i - \theta V(s_t) + \theta' A(s_t, a_t))^2] \end{aligned} \tag{5.13}$$

with

$$\bar{y}_i = r_t + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) \tag{5.14}$$

The gradient update method is

$$\begin{aligned}\nabla_{\Theta_i} L_i(\Theta_i) &= \mathbb{E}_{s_t,a_t,r_t,s_{t+1}}[(\bar{y}_i - Q(s_t,a_t))\nabla_{\Theta_i} Q(s_t,a_t)] \\ &= \mathbb{E}_{s_t,a_t,r_t,s_{t+1}}[(\bar{y}_i - \theta V(s_t) - \theta' A(s_t,a_t)) \\ &\nabla_{\Theta_i}(\theta V(s_t) + \theta' A(s_t,a_t))]\end{aligned} \tag{5.15}$$

5.1.4 Weighted Average Spatial LSTM Classifier

In this section, we propose Weighted Average Spatial LSTM classification for two purposes. The first attempt is to capture the cross-relationship among feature dimensions in the optimized attention zone $\bar{\mathbf{x}}_i^*$. The LSTM-based classifier is widely used for its excellent sequential information extraction ability which is approved in several research areas such as natural language processing [273, 274]. Compared to other commonly employed spatial feature extraction methods, such as Convolutional Neural Networks, LSTM is less depends on the hyper-parameters setting. However, the traditional LSTM focuses on the temporal dependency among a sequence of samples. Technically, the input data of traditional LSTM is 3-D tensor shaped as $[n_b, n_t, \bar{K}]$ where n_b and n_s denote the batch size and the number of temporal sample, separately. The WAS-LSTM aims to capture the dependency among various dimensions at one temporal point, therefore, we set $n_t = 1$ and transpose the input data as: $[n_b, n_t, \bar{K}] \rightarrow [n_b, \bar{K}, n_t]$.

The second advantage of WAS-LSTM is that it could stabilize the performance of LSTM via moving average method [275]. Specifically, we calculate the LSTM outputs $\mathbf{O}_i$ by averaging the past two outputs instead of only the final one (Figure 5.1):

$$\mathbf{O}_i = (\mathbf{O}_{i(\bar{K}-1)} + \mathbf{O}_{i\bar{K}})/2 \tag{5.16}$$

The predicted label is calculated by $y_i' = \mathcal{L}(\bar{\mathbf{x}}_i^*)$ where $\mathcal{L}$ denotes the LSTM algorithm. ℓ_2-norm (with parameter λ) is adopted as regularization to prevent overfitting. The sigmoid activation function is used on hidden layers. The loss function is cross-entropy and is optimized by the AdamOptimizer algorithm [231].

Datasets	Type	Task	#-S	#-C	Samples	#-D	S-rate (Hz)
EID	EEG	PID	8	8	168,000	14	128
RSSI	RFID	AR	6	21	3,100	12	2
PAMAP2	IMU	AR	9	8	120,000	14	100

Table 5.1: Datasets description. PID, AR, and S-rate denote Person Identification, Activity Recognition, and Sampling rate, respectively. #-S, #-C, #-D separately denote the number of subjects, classes, and dimensions.

5.2 Experiments

In this section, we evaluate the proposed approach over 3 sensor signal datasets (separately collected by EEG headset, environmental sensor, and wearable sensor) including 2 widely used public datasets and 2 limited but more practical local datasets. Firstly we describe the details of each dataset. Secondly, we demonstrate the effectiveness and robustness by comparing the performance of our approach to baselines and state-of-the-art. Lastly, we provide the efficiency of the alternative reward model designed in Section 5.1.3.

5.2.1 Datasets

More details refer to Table 5.1.

- **EID.** The EID (EEG ID identification) is collected in a constrained setting where 8 subjects (5 males and 3 females) aged 26 ± 2. EEG signal monitors the electrical activity of the brain. This dataset gathers the raw EEG signals by Emotiv EPOC+ headset with 14 channels at the sampling rate of 128 Hz.
- **RSSI.** The RSSI (Radio Signal Strength Indicator) [276] collects the signals from passive RFID tags. 21 activities, including 18 ADLs (Activity of Daily Living) and 3 abnormal falls, are performed by 6 subject aged 25 ± 5. RSSI measures the power present in a received radio signal, which is a convenient environmental measurement in ubiquitous computing.

Methods		EID Dataset			
		Acc	**Pre**	**Rec**	**F1**
Non-DL	**SVM**	0.1438	0.1653	0.1545	0.1445
	RF	0.9365	0.9261	0.9142	0.9457
	KNN	0.9413	0.9471	0.9298	0.9511
	AB	0.2518	0.2684	0.2491	0.2911
	LDA	0.1485	0.1524	0.1358	0.1479
DL	**LSTM**	0.4315	0.5132	0.4278	0.4532
	GRU	0.4314	0.455	0.4288	0.4218
	1-D CNN	0.8031	0.8127	0.805	0.8278
	WAS-LSTM	0.9518	0.9657	**0.9631**	**0.9658**
	Ours	**0.9621**	**0.9618**	0.9615	0.9615

Table 5.2: Comparison of EID

Methods		RSSI Dataset			
		Acc	**Pre**	**Rec**	**F1**
Non-DL	**SVM**	0.8918	0.8924	0.8908	0.8805
	RF	0.9614	0.9713	0.9652	0.9624
	KNN	0.9612	0.9628	0.9618	0.9634
	AB	0.4704	0.4125	0.4772	0.3708
	LDA	0.8842	0.8908	0.8845	0.8802
DL	**LSTM**	0.7421	0.6505	0.6132	0.6858
	GRU	0.7049	0.7728	0.6584	0.6915
	1-D CNN	0.9714	0.9676	0.9635	0.9645
	WAS-LSTM	0.9553	0.9533	0.9545	0.9592
	Ours	**0.9838**	**0.9782**	**0.9669**	**0.9698**

Table 5.3: Comparison of RSSI

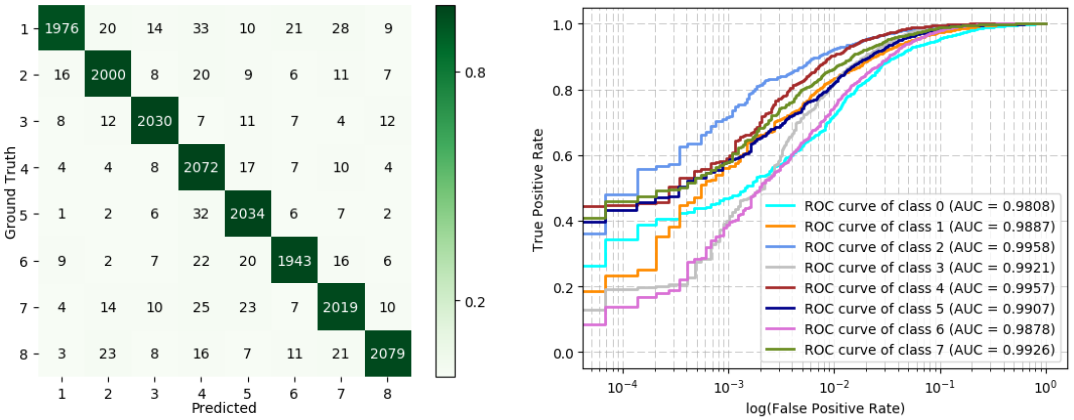
- **PAMAP2.** The PAMAP2 [277] is collected by 9 participants (8 males and 1 females) aged 27 ± 3. 8 ADLs are selected as a subset of our paper. The activity is measured by 1 IMU attached to the participants' wrist. The IMU collects sensor signal with 14 dimensions including two 3-axis accelerometers, one 3-axis gyroscopes, one 3-axis magnetometers and one thermometer.

Methods		PAMAP2 Dataset			
		Acc	**Pre**	**Rec**	**F1**
Non-DL	**SVM**	0.7492	0.7451	0.7522	0.7486
	RF	0.9817	0.9893	0.9711	**0.9801**
	KNN	0.9565	0.9651	0.9625	0.9638
	AB	0.5776	0.4298	0.5814	0.4942
	LDA	0.7127	0.7175	0.7298	0.7236
DL	**LSTM**	0.7925	0.7487	0.7478	0.7482
	GRU	0.8625	0.8515	0.8349	0.8431
	1-D CNN	0.9819	0.9715	0.9721	0.9718
State-of-the-Arts	**[277]**	0.96	-	-	-
	[278]	0.8488	-	-	0.841
	[279]	0.967	-	-	-
	[280]	0.9336	-	-	-
	WAS-LSTM	0.9821	**0.9981**	0.9459	0.9713
	Ours	**0.9882**	0.9804	**0.9756**	0.9780

Table 5.4: Comparison of PAMAP2

5.2.2 Results

In this section, we compare the proposed approach with baselines and the state-of-the-art methods. Our method focuses on the attention zone which is optimized by deep reinforcement learning and then explores the dependency between sensor signal elements by a deep learning classifier. All the three datasets are randomly split into the training set (90%) and the testing set (10%). Each sample is one sensor vector recording collected at one time point. Through the previous experimental tuning and the Orthogonal Array based hyper-parameters tuning method [2], the hyper-parameters are set as following. In the selective attention learning: the order of autoregressive is 3; $\bar{K} = 128$, the Dueling DQN has 4 lays and the node number in each layer are: 2 (input layer), 32 (FCL), 4 ($A(s_t, a_t)$) + 1 ($V(s_t)$), 4 (output). The decay parameter $\gamma = 0.8$, $n_e = n_s = 50$, $N = 2,500$, $\epsilon = 0.2$, learning rate= 0.01, memory size = 2000, length penalty coefficient $\beta = 0.1$, and the minimum length of attention zone is set as 10. In the deep learning classifier: the node number in the input layer equals to the number of feature dimensions, three hidden layers with 164 nodes, two layers of LSTM cells and one output layer. The learning rate = 0.001,



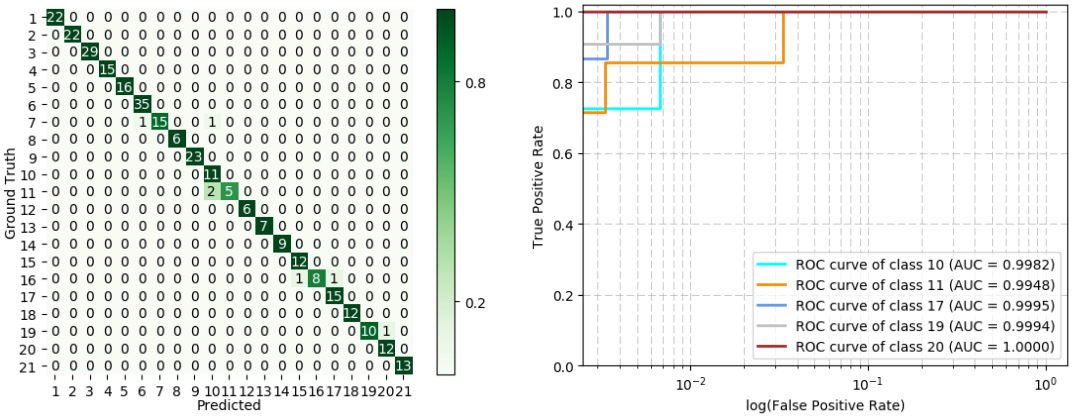
(a) EID CM (b) EID ROC

Figure 5.3: Confusion matrix and ROC curves of EID

ℓ_2-norm coefficient $\lambda = 0.001$, forget bias = 0.3, batch size = 9, and iterate for 1000 iterations.

Tables 5.2 $\sim$ 5.4 show the classification metrics comparison between our approach and baselines including Non-DL and DL baselines. Since the EID and RSSI are local datasets, we only compare with state-of-the-art over the public dataset PAMAP2. Table 5.4 shows that our approach achieves the highest accuracy on both datasets. DL represents deep learning. The notation and hyper-parameters of the baselines are listed here. RF denotes Random Forest, AdaB denotes Adaptive Boosting, LDA denotes Linear Discriminant Analysis. In addition, the key parameters of the baselines are listed here: Linear SVM ($C = 1$), RF ($n = 200$), KNN ($k = 3$). In LSTM, $n_{steps} = 5$, another set is the same as the WAS-LSTM classifier, along with the GRU (Gated Recurrent Unit [281]). The CNN works on sensor data and contains 2 stacked convolutional layers (both with stride $[1, 1]$, patch $[2, 2]$, zero-padding, and the depth are 4 and 8, separately.) and followed by one pooling layer (stride $[1, 2]$, zero-padding) and one fully connected layer (164 nodes). Relu activation function is employed in the CNN. The results from Tables 5.2 $\sim$ 5.4 show that:

- Our approach outperforms all the baselines and the state-of-the-arts over all the local and public datasets ranging from EEG, RFID to wearable IMU sen-



(a) RSSI CM (b) RSSI ROC

Figure 5.4: Confusion matrix and ROC curves of RSSI. This dataset overall contains 21 classes and we only select several representative classes in ROC curves.

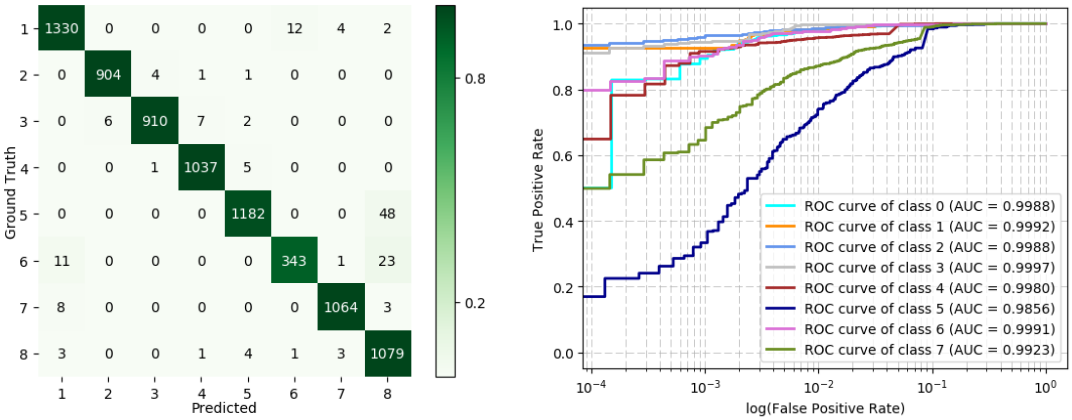
sors;

- The sensor spatial based WAS-LSTM classifier achieves a high-level performance, which indicates the method that extracting inter-dimension dependency for classification is effective;
- Our method (WAS-LSTM with attention zone) performs better than WAS-LSTM, which illustrates that the learned informative attention is effective.

To have a closer observation, the CM (confusion matrix) and the ROC curves (including the AUC score) of the datasets are reported in Figure 5.3 to Figure 5.5. The CMs illustrate that the robustness of the proposed approach keeps high accuracy even over few samples and numerous categories.

5.2.3 Reward Model Efficiency Demonstration

In this section, we propose a new reward model to replace the original reward model: $\mathcal{G} \rightarrow \mathcal{F}$. The original $\mathcal{F}$, in our case, refers to the WAS-LSTM classifier (Section 5.1.4), intuitively. $\mathcal{F}$ requires a large amount of training time to find the



(a) PAMAP CM (b) PAMAP ROC

Figure 5.5: Confusion matrix and ROC curves of PAMAP.

optimal attention zone $\bar{\mathbf{x}}^*$. Take the EID dataset as an example, $\mathcal{F}$ needs around 4000 sec on the Titan X (Pascal) GPU for each step while the whole attention zone optimization contains N ($N > 2000$) iterations. Therefore, to save training time, we attempt to employ $\mathcal{G}$ to approximate $\mathcal{G}$ to update the reward. Thus, two prerequisites are demanded: 1) $\mathcal{G}$ should have high correlation with $\mathcal{F}$ to guarantee $\arg\max_{\bar{\mathbf{x}}^*} \mathcal{G} \approx \arg\max_{\bar{\mathbf{x}}^*} \mathcal{F}$; 2) the training time of $\mathcal{G}$ should be shorter than $\mathcal{F}$. In this section, we demonstrate the two prerequisites by experimental analyzes.

First, on the attention zone optimization procedure on EID dataset, we conduct an experiment to measure a batch of data pairs of the reward (represents the reward of $\mathcal{G}$) and the WAS-LSTM classifier accuracy (represents the reward of $\mathcal{F}$). The relationship between the reward and the accuracy is shown in Figure 5.6. The figure illustrates that the accuracy has an approximately linear relationship with the reward. The correlations coefficient is 0.8258 (with p-value as 0.0115), which demonstrates the accuracy and reward are highly positive related. As a result, we can estimate $\arg\max_{\bar{\mathbf{x}}^*} \mathcal{F}$ by $\arg\max_{\bar{\mathbf{x}}^*} \mathcal{G}$. Moreover, another experiment is carried on to measure the single step training time of two reward models $\mathcal{G}$ and $\mathcal{F}$. The training times are marked as T1 and T2, respectively. Figure 5.7 qualitatively shows that T2 is much higher than T1 (8 states represent 8 different attention zones). Quantitatively, the sum of T1 over 8 states is 35237.41 sec while the sum of T2

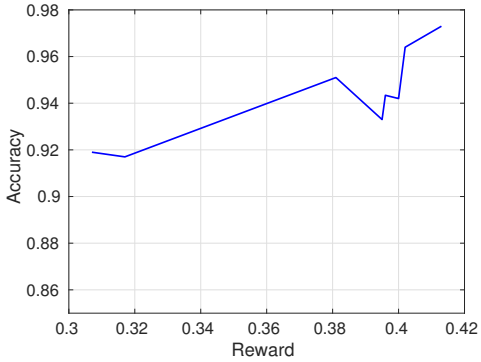


Figure 5.6: The relationship between the classifier accuracy and the reward. The correlationship coefficient is 0.8258 while the p-value is 0.0115.

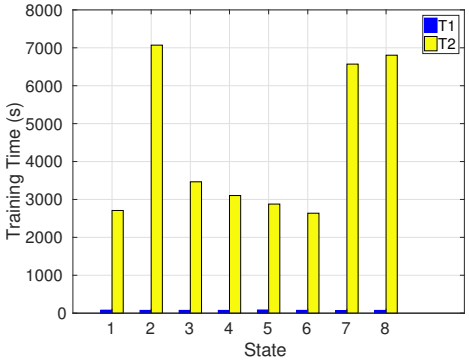


Figure 5.7: Reward model training time in various states. T1 and T2 separately denote the training time in reward model $\mathcal{G}$ and $\mathcal{F}$.

is 601.58 sec. This results demonstrate that the proposed approach, designing a $\mathcal{G}$ to approximate and estimate the $\mathcal{F}$, saves **98.3**% training time in attention zone optimization.

5.2.4 Discussions

In this section, we discuss several characteristics of the proposed approach.

First, we propose a robust, universal, and adaptive classification framework which can efficiently deal with adaptive sensor data. Specifically, our approach works better on high-dimensional feature space in that the information of inter-dimension dependency is richer.

In addition, we propose a novel idea that adopts an alternative reward model to estimate and replace the original reward model. In this way, the disadvantages of the original model, such as expensive computation, can be eliminated. The key is to keep the reward produced by the new model highly related to the original reward. The higher correlation coefficient, the better. This sheds light on the possible combination of deep learning classifier and reinforcement learning.

Nevertheless, one weakness is that the reinforcement learning policy only works well in the specific environment in which the model is trained. The dimension indexes should be consistent in training and testing stages. Various policies should be trained according to different sensor combinations.

Furthermore, the proposed WAS-LSTM directly focuses on the dependency among the sensor dimensions and can produce a predicted label for each point. This provides the foundation for the quick-reaction online detection and other applications which require instantaneous detection. However, this retires an enough number of signal dimensions to carry sufficient information for the aim of accurately recognition.

5.3 Conclusion

In this chapter, we present a robust and efficient adaptive sensor data classification framework which integrates selective attention mechanism, deep reinforcement learning, and WAS-LSTM classification. In order to boost the chance of inter-dimension dependency in sensor features, we replicate and shuffle the sensor data. Additionally, the optimal spatial dependency is required for high-quality classification, for which we introduce the attention zone with attention mechanism. Furthermore, we extended the LSTM to exploit the cross-relationship among spatial dimensions, which is called WAS-LSTM, for classification. The proposed approach is evaluated on three different sensor datasets, namely, EEG, RFID and wearable IMU sensors. The experimental results show that our approach outperforms the state-of-the-art baselines. Moreover, the designed reward model saves **98.3%** of the training time in reinforcement learning.

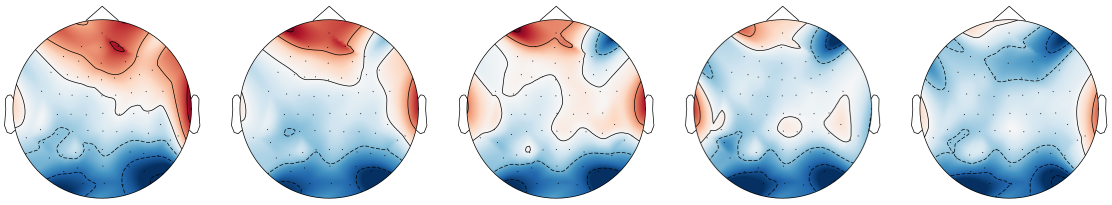
Chapter 6

Adaptive Cognitive Activity Recognition with Reinforced CNN

In the previous chapter, we proposed an adaptive learning framework which works well on different sensory signals. In this chapter, we attempt to refine the task and focuses on a general model to deal with various EEG scenarios.

The accuracy and robustness of EEG classification model have promising meanings to identify cognitive activities in the realms of movement intention recognition, person identification, and neurological diagnosis. Cognitive activity recognition systems [282] provide a bridge between the inside cognitive world and the outside physical world. They are recently used in assisted living [9], smart homes [2], and entertainment industry [283]; EEG-based person identification technique empowers the security systems deployed in bank or customs [284, 8]; EEG signal-based neurological diagnosis can be used to detect the organic brain injury and abnormal synchronous neuronal activity such as epileptic seizure [285, 286].

The classification of cognitive activity faces several challenges. First, the EEG data preprocessing and feature extraction methods (e.g., filtering, Discrete Wavelet



(a) T-2 (b) T-1 (c) T (d) T+1 (e) T+2

Figure 6.1: EEG topography with continuous samples. The interval among samples is 0.00625 second.

Transformation, and feature selection) which are employed by most existing EEG classification studies [268, 283] are time-consuming and highly depend on expertise. Meanwhile, the hand-crafted features require extensive experiments to generalize well to diverse settings such as filtering bands and wavelet orders. Therefore, an effective method which can directly work on raw EEG data is necessary.

Second, most current EEG classification methods are designed based on domain-specific knowledge and thus may become ineffective or even fail in different scenarios [287]. For example, the approach customized for EEG-based neurological diagnosis may not work well on intention recognition. Therefore, a general EEG signal classification method is expected to be both efficient and robust across various domains for better usability and suitability.

Third, EEG signals have a low signal-to-noise ratio and more chaotic than other sensor signals such as wearable sensors. Thus, the segment-based classification which is widely used in sensing signal classification may not fit cognitive activity recognition. A segment contains some continuous EEG samples clipped by the sliding window method [288] while a single EEG sample (also called EEG instance) is collected at a specific time point. In particular, segment-based classification has two drawbacks compared with sample-based classification: 1) in a segment with many samples, the sample diversity may offset by other inverse changed samples as EEG signals vary rapidly (Section 6.1). 2) segment-based classification requires more training data and a longer data-collecting time. For example, suppose each segment has ten samples without overlapping; for the same training batch size, segment-based

classification requires ten times of the data size and the data-collecting time than sample-based classification. As a result, segment-based classification cannot exploit the immediate intention of changing and thus achieves low precision in practical deployment. To this end, sample-based classification is more attractive.

To address the aforementioned issues, first, we propose a novel framework which can automatically learn distinctive features from raw EEG signals by developing a deep convolutional mapping component. Additionally, to grasp the characteristic information from different EEG application circumstance adaptively, we design a reinforced selective attention component that combines the benefits of attention mechanism [289] and deep reinforcement learning. Moreover, we overstep the challenge of chaotic information by working on EEG samples instead of segments. The single EEG sample only contains spatial information without spatial clue[1]. The main contributions of this chapter are highlighted as follows:

- We propose a general framework for automatic cognitive activity recognition to facilitate a scope of diverse cognitive application domains.
- We design the reinforced selective attention model, by combining the deep reinforcement learning and attention mechanism, to automatically extract the robust and distinct deep features. Specially, we design a non-linear reward function to encourage the model to select the best attention area that leads to the highest classification accuracy. Besides, we customize the states and actions based on our cognitive activity recognition environment.
- We develop a convolutional mapping method to explore the distinguishable spatial dependency and feed it to the classifier for classification, among selected EEG signals.
- We demonstrate the effectiveness of the proposed framework using four real-world datasets concerning three representatives and challenging cognitive ap-

[1]We do not deny the usefulness of temporal information, but this chapter emphasizes on spatial information, which may easier to be captured.

Table 6.1: Time domain and correlation coefficient analysis. *n-points* denotes the values are measured by the samples with n sampling points. We compare EEG signals with other sensing data (such as wearable sensor data and smartphone data) over five different scales and the results constantly show that EEG signals have the highest instability.

	Signals	5-points		50-points		100-points		500-points		1000-points		Average	
		STD	Range	STD	Range	STD	Range	STD	Range	STD	Range	STD	Range
Time Domain	**Phone**	0.0025	0.0061	0.0179	0.0494	0.0166	0.0612	0.0253	0.1177	0.0259	0.1281	0.0882	0.3625
	Wearable	0.0012	0.0029	0.0107	0.0369	0.0147	0.0519	0.0197	0.1041	0.016	0.1058	0.0623	0.3016
	EEG	0.0087	0.0218	0.0199	0.0824	0.0245	0.1195	0.0299	0.1619	0.0308	0.1802	**0.1138**	**0.5658**
	Signals	**5-points**		**50-points**		**100-points**		**500-points**		**1000-points**		**Average**	
		STD	**Range**	**STD**	**Range**	**STD**	**Range**	**STD**	**Range**	**STD**	**Range**	**STD**	**Range**
Correlation Coefficient	**Phone**	0.0015	0.0038	0.0243	0.0832	0.0248	0.0964	0.0244	0.104	0.0247	0.104	0.0997	0.3914
	Wearable	0.01	0.0252	0.0155	0.0702	0.0147	0.0866	0.0469	0.2299	0.0729	0.3905	0.16	0.8024
	EEG	0.0392	0.0991	0.1077	0.4096	0.0955	0.4849	0.1319	0.7626	0.1533	0.99	**0.5276**	**2.7462**

plications. The experiment results demonstrate that the proposed framework outperforms the state-of-the-art and strong baselines by consistently achieving the accuracy of more than 96% and low latency.

Note that all the necessary reusable codes and datasets have been open-sourced for reproduction, please refer to this link[2].

6.1 Analysis of EEG Signals

In this section, we demonstrate EEG signals' unique characteristics (e.g., rapid-varying and chaotic) and that single samples are more suitable than segments for classification. By comparing EEG signals with two typical sensor signals collected by smartphone (accelerometers in Samsung Galaxy S2) and wearable sensors (Colibri wireless IMU). The participants are walking during the data collection session.

The brain activity is very complex and rapid varying, but EEG signals can only capture a few information through the discrete sampling of biological current. Figure 6.1 demonstrates the characteristics of rapidly varying and complex of EEG signals and provides the EEG topography of consecutive 5 samples. The sampling

[2] `https://github.com/xiangzhang1015/know_your_mind`

rate is 160 Hz while the sampling interval is 0.00625 second. It can be observed that the topography changes dramatically within such a tiny time interval.

Furthermore, to illustrate the chaotic of EEG signals, we compare EEG with smartphone and wearable sensors in two aspects: the time domain and the inter-samples correlations.

In the time domain, we evaluate the STD and range of sensor signals on five levels of sample length: 5, 50, 100, 500, 1000 continuous samples. The evaluations on the above five scales are expected to show the tendency that how the EEG characteristic varies with the sampling period.

The inter-sample correlation coefficient calculates the average cosine correlations between the specific sample and its neighbor samples (5, 50, 100, 500, and 1000 samples). A low correlation coefficient represents EEG signals dramatically and rapidly varying all the time.

As a result, Table 6.1 present the STD and range values in the time domain and correlation coefficient. We observe that EEG signals have the highest STD and range over all the five sample window scales both on time domain and correlation coefficient, compared with wearable sensor data and smartphone signals. This demonstrates that the EEG sample has more unstable correlations with neighbors and the instability is very high even in the nearest five samples. More specifically, EEG signals are very chaotic and rapidly changing at each single sampling point.

6.2 Proposed Method

Based on the above analysis, we propose reinforced attentive CNN to classify raw EEG signals accurately and efficiently directly. The overall workflow is shown in Figure 6.2. The replicate and shuffle component and the reinforced selective attention component are very similar to Section 5.1.2 and Section 5.1.3. Here, we mainly

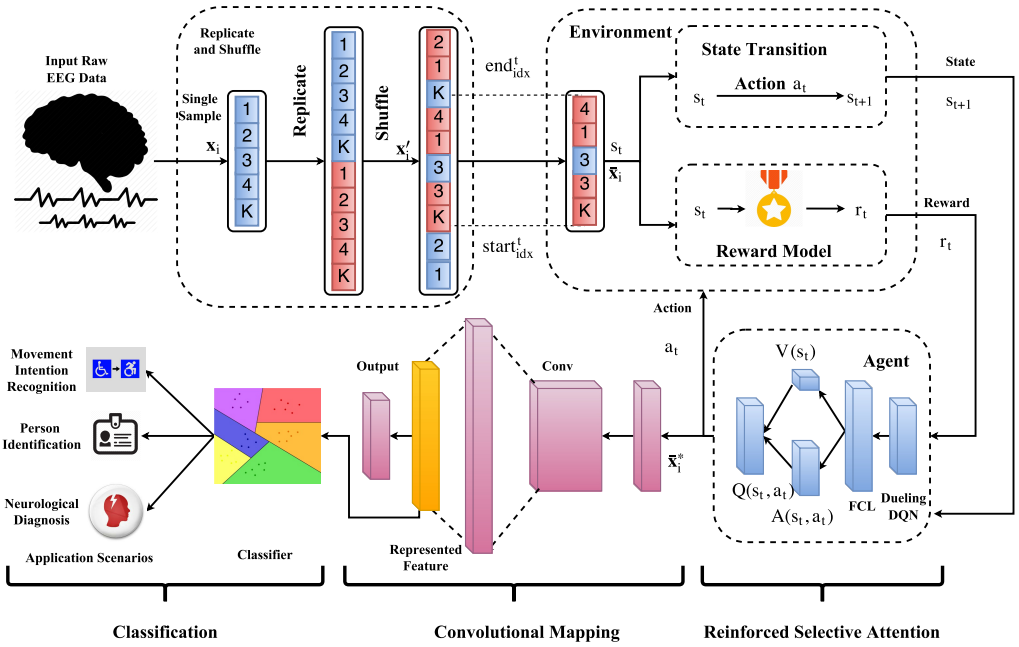


Figure 6.2: Flowchart of the proposed approach. The input raw EEG single sample $\mathbf{x}_i$ (K denotes the Kth element) is replicated and shuffled to provide more latent spatial combinations of feature dimensions. Then, an attention zone $\bar{\mathbf{x}}_i$, which is a fragment in $\mathbf{x}'_i$, with the state $s_t = \{start^t_{idx}, end^t_{idx}\}$ is selected. The selected attention zone is input to the state transition and the reward model. In each step t, one action is selected by the state transition to update s_t based on the agent's feedback. The reward model evaluates the quality of the attention zone by the reward score r_t. The dueling DQN is employed to discover the best attention zone $\bar{\mathbf{x}}^*_i$ which will be fed into the convolutional mapping procedure to extract the spatial dependency representation. The represented features will be used for the classification. FCL denotes a fully connected layer. The reward model is the combination of the convolutional mapping and the classifier.

introduce the convolutional mapping.

6.2.1 Convolutional Mapping

For each attention zone, we further exploit the potential spatial dependency of selected features $\bar{\mathbf{x}}^*_i$. Since we focus on a single sample, the EEG sample only contains a numerical vector with very limited information and is easily corrupted by noise. To amend this drawback, we attempt to mapping the EEG single sample

from the original space $\mathcal{O} \in R^K$ to a sparsity space $\mathcal{T} \in R^M$ by a CNN structure.

To extract as more potential spatial dependencies as possible, we employ a convolutional layer [290] with many filters to scan on the learned attention zone $\bar{\mathbf{x}}_i^*$. The convolutional mapping structure contains five layers (as shown in Figure 6.2): the input layer receives the learned attention zone, the convolutional layer followed by one fully connected layer, and the output layer. The one-hot ground truth is compared with the output layer to calculate the training loss.

The Relu non-linear activation function is applied to the convolutional outputs. We describe the convolutional layer as follows:

$$x_{ij}^c = ReLU(\sum_{b=1}^{\bar{b}} W_c \bar{x}_{ij}^*) \tag{6.1}$$

where x_{ij}^c denotes the outcome of the convolutional layer while $\bar{b}$ and W_c denote the length of filter and the filter weights, respectively. The pooling layer aims to reduce the redundant information in the convolutional outputs to decrease the computational cost. In our case, we try to keep as much information as possible. Therefore, our method does not employ a pooling layer. Then, in the fully connected layer and output layer

$$x_i^f = ReLU(W^f x_i^c + b^f) \tag{6.2}$$

$$y_i' = softmax(W^o x_i^f + b^o) \tag{6.3}$$

where W^f, W^o, b^f, b^o denote the corresponding weights and biases, respectively. The y' denotes the predicted label. The cost function is measured by cross entropy, and the ℓ_2-norm (with parameter λ) is adopted as regularization to prevent overfitting.:

$$cost = -\sum_x y_i' log(y_i) + \lambda \ell_2 \tag{6.4}$$

The AdamOptimizer algorithm optimizes the cost function. The fully connected layer extracts as the represented features and fed them into a lightweight nearest neighbor classifier. The convolutional mapping updates for N' iterations. The proposed adaptive cognitive activity recognition with reinforced attentive convolutional neural networks is shown in Algorithm 1.

ALGORITHM 1: The Proposed Approach

Input: Raw EEG signals $\mathbf{X}$

Output: Predicted cognitive activity label y_i'

Initialization s_0;

RS: $\bar{\mathbf{x}}_i \leftarrow \mathbf{x}_i'$;

Reinforced Selective Attention:

if $t < N$ **then**

    $a_t = argmax_{a_t \in \mathcal{A}} Q(s_t, a_t)$

    $s_{t+1} = \pi(s_t, a_t)$

    $r_t = \mathcal{F}(s_t)$

    $\varepsilon_{t+1} = \varepsilon_t + \varepsilon_0 N$

    $\bar{\mathbf{x}}_i^* \leftarrow \bar{\mathbf{x}}_i, a_t, s_t, r_t$

end if

Convolutional Mapping & Classifier:

if $iteration < N'$ **then**

    $y_i' \leftarrow \bar{\mathbf{x}}_i^*$

end if

return y_i'

6.3 Experiments

In this section, we report our evaluation of the proposed approach on three datasets corresponding to different application scenarios, with a focus on accuracy, latency, and resilience.

Table 6.2: Comparison with baselines

Scenarios	Datasets	Metrics	Non-Deep Learning Baselines					Deep Learning Baselines			
			SVM	RF	KNN	AB	LDA	LSTM	GRU	CNN	Ours
MIR	EEGMMIDB	Accuracy	0.5596	0.6996	0.5814	0.3043	0.5614	0.648	0.6786	0.91	**0.9632**
		Precision	0.5538	0.7311	0.6056	0.2897	0.5617	0.6952	0.8873	0.9104	0.9632
		Recall	0.5596	0.6996	0.5814	0.3043	0.5614	0.6446	0.6127	0.9104	0.9632
		F1-score	0.5396	0.6738	0.5813	0.2037	0.5526	0.6619	0.7128	0.9103	0.9632
PI	EEG-S	Accuracy	0.6604	0.9619	0.9278	0.35	0.6681	0.9571	0.9821	0.998	**0.9984**
		Precision	0.6551	0.9625	0.9336	0.3036	0.6779	0.9706	0.9858	0.998	0.9984
		Recall	0.6604	0.962	0.9279	0.35	0.6681	0.9705	0.9857	0.998	0.9984
		F1-score	0.6512	0.9621	0.9282	0.2877	0.668	0.9705	0.9857	0.998	0.9984
ND	TUH	Accuracy	0.7692	0.92	0.9192	0.5292	0.7675	0.6625	0.6625	0.9592	**0.9975**
		Precision	0.7695	0.9206	0.923	0.7525	0.7675	0.6538	0.6985	0.9593	0.9975
		Recall	0.7692	0.92	0.9192	0.5292	0.7675	0.6417	0.6583	0.9592	0.9975
		F1-score	0.7692	0.9199	0.9188	0.3742	0.7675	0.6449	0.6685	0.9592	0.9975

6.3.1 Application Scenarios and Datasets

Application Scenarios We evaluate our approach on various datasets in three applications of EEG-based Brain-Computer Interfaces.

Movement Intention Recognition (MIR). EEG signals measure human brain activities. Intuitively, different human intention will lead to diverse EEG patterns [9]. Intention recognition plays a significant role in practical scenarios such as smart home, assisted living [2], brain typing [9], and entertainment. For the disabled and elders, intent recognition can help them to interact with external smart devices such as wheelchairs or service robots real-time BCI systems. Besides, for people without vocal ability, they may have the chance to express their thoughts with the help of certain intention recognition technologies (e.g., brain typing). Even for the healthy human being, intent recognition can be used in video game playing and other daily living applications.

Person Identification (PI). EEG-based biometric identification [284] is an emerging person identification approach, which is highly attack-resilient. It has the unique advantage of avoiding or alleviating the threat of being deceived which is often faced by other identification techniques. This technique can be deployed in identification and authentication scenarios such as bank security system and customs security check.

Neurological Diagnosis (ND). EEG signals collected in the unhealthy state differ significantly from the ones collected in the normal state concerning frequency and pattern of neuronal firing [287]. Therefore, EEG signals have been used for neurological diagnosis for decades [6]. For example, the epileptic seizure is a common brain disorder that affects around 1% of the population, and an EEG analysis of the patient could detect its octal state.

Datasets To evaluate how the proposed approach works in the aforementioned application scenarios, we choose several EEG datasets with various collection equipment, sampling rates, and data sources. We utilize motor imagery EEG signals from a public dataset *eegmmidb* for intention recognition, the *EEG-S* dataset for person identification, and the *TUH* dataset for neurological diagnosis.

EEGMMIDB. This dataset is introduced in Section 3.3.1.

EEG-S. EEG-S is a subset of eegmmidb, in which the data were gathered while the subject kept eyes closed and stayed relaxed. Eight subjects were involved and each subject generated 7,000 samples. Labels are the subjects' IDs, which range within [0-7].

TUH. TUH [291] is a neurological seizure dataset of clinical EEG recordings. The EEG recording is associated with 22 channels from a 10/20 configuration and a sampling rate of 250 Hz. We selected 12,000 samples from each of five subjects (2 males and three females). Half of the samples were labeled as epileptic seizure state. The remaining samples were labeled as the normal state.

6.3.2 Parameter Settings

We configured the default settings of our approach as follows. In the selective attention learning: $\bar{K} = 128$, the Dueling DQN had 4 lays and the node number in each layer were: 2 (input layer), 32 (FCL), 4 ($A(s_t, a_t)$) + 1 ($V(s_t)$), and 4 (output).

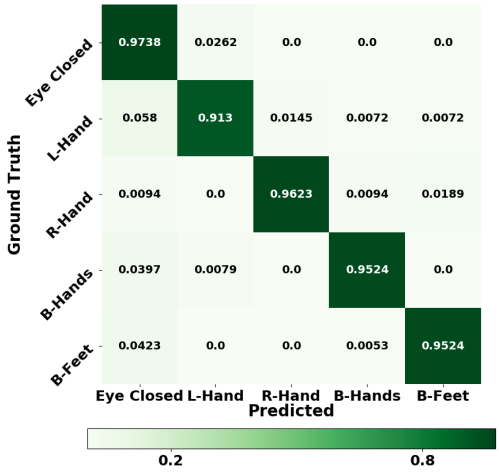
Table 6.3: Comparison with the state-of-the-art approaches

Scenarios	Datasets	Metrics	State-of-the-art					
MIR	EEGMMIDB	**Method**	**Rashid [292]**	**Zhang [9]**	**Ma [293]**	**Alomari [217]**	**Sita [294]**	**Alomari [295]**
		Accuracy	0.9193	0.9561	0.6820	0.8679	0.7584	0.8515
		Precision	0.9156	0.9566	0.6971	0.8788	0.7631	0.8469
		Recall	0.9231	0.9621	0.7325	0.8786	0.7702	0.8827
		F1-score	0.9193	0.9593	0.7144	0.8787	0.7666	0.8644
		Method	**Shenoy [220]**	**Szczuko [296]**	**Stefano[297]**	**Pinheiro [236]**	**Kim [298]**	**Ours**
		Accuracy	0.8308	0.9301	0.8724	0.8488	0.8115	**0.9632**
		Precision	0.8301	0.9314	0.8874	0.8513	0.8128	0.9632
		Recall	0.8425	0.9287	0.8874	0.8569	0.8087	0.9632
		F1-score	0.8363	0.9300	0.8874	0.8541	0.8107	0.9632
PI	EEG-S	**Method**	**Ma [299]**	**Yang [300]**	**Rodrigues [301]**	**Frashini [288]**	**Thomas [302]**	**Ours**
		Accuracy	0.88	0.99	0.8639	0.956	0.9807	**0.9984**
		Precision	0.8891	0.9637	0.8721	0.9458	0.9799	0.9984
		Recall	0.8891	0.9594	0.8876	0.9539	0.9887	0.9984
		F1-score	0.8891	0.9615	0.8798	0.9498	0.9843	0.9984
ND	TUH	**Method**	**Ziyabari [303]**	**Harati [304]**	**Zhang [305]**	**Goodwin [306]**	**Golmoh [307]**	**Ours**
		Accuracy	0.9382	0.9429	0.994	0.924	0.9479	**0.9975**
		Precision	0.9321	0.9503	0.9951	0.9177	0.9438	0.9975
		Recall	0.9455	0.9761	0.9951	0.9375	0.9522	0.9975
		F1-score	0.9388	0.9630	0.9951	0.9275	0.9480	0.9975

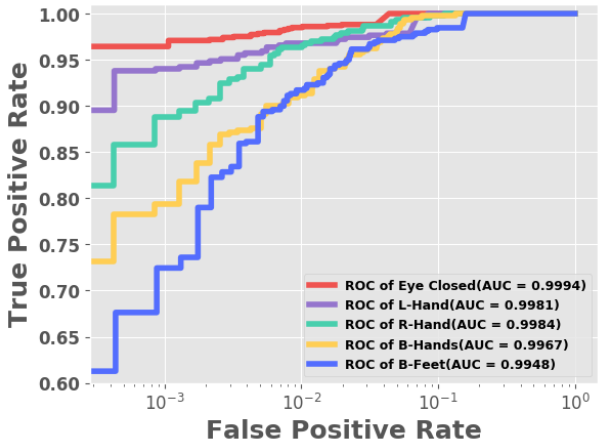
The decay parameter $\gamma = 0.8$, $n_e = n_s = 50$, $N = 2,500$, $\epsilon = 0.2$, $\epsilon_0 = 0.002$, learning rate= 0.01, memory size = 2000, length penalty coefficient $\beta = 0.1$, and the minimum length of attention zone was set as 10. In the convolutional mapping, the node number in the input layer equaled to the number of attention zone dimensions. In the convolutional layer: the stride had the shape $[1, 1]$, the filter size was set to $[1, 2]$, the depth to 10, and the non-linear function as ReLU. The padding method was zero-padding. No pooling layer was adopted. The subsequent fully connected layer had 100 nodes. The learning rate was 0.001 while the ℓ_2-norm coefficient λ equaled 0.001. The transformation was trained for 2000 iterations. In addition, we configured the key parameters of the baselines as follows: Linear SVM ($C = 1$), Random Forest (RF, $n = 200$), KNN ($k = 1$). In LSTM (Long Short-Term Memory) and GRU (Gated Recurrent Unit), $n_{steps} = 5$, other settings were the same as [2]. The CNN had the same structure and hyper-parameters setting with the convolutional mapping component in the proposed show.

6.3.3 Overall Comparison

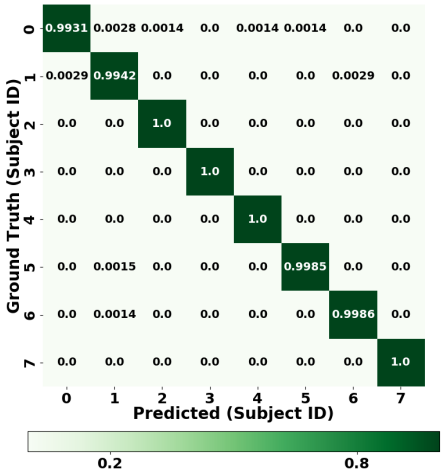
Comparison Baselines To measure the accuracy of the proposed method, we compared with a set of baseline methods including five non-deep learning and three



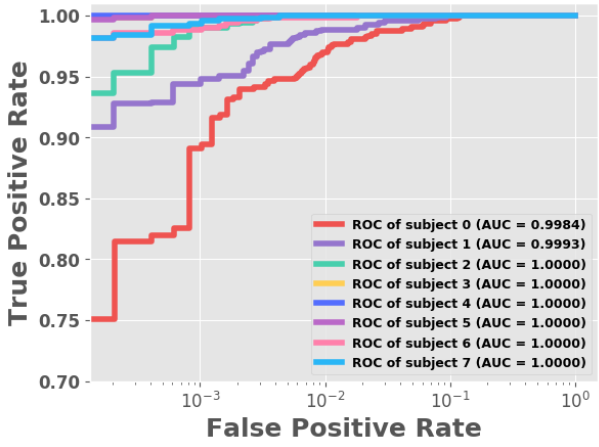
(a) CM of eegmmidb



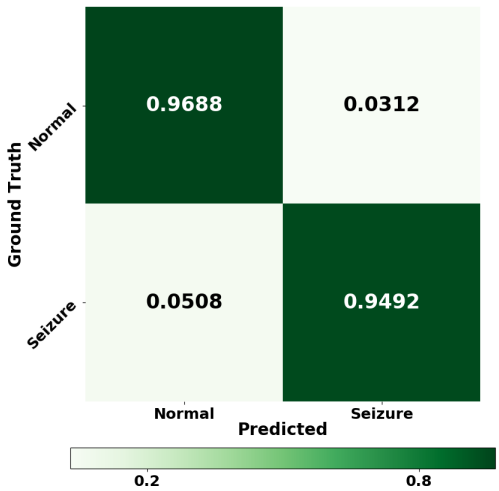
(b) ROC of eegmmidb



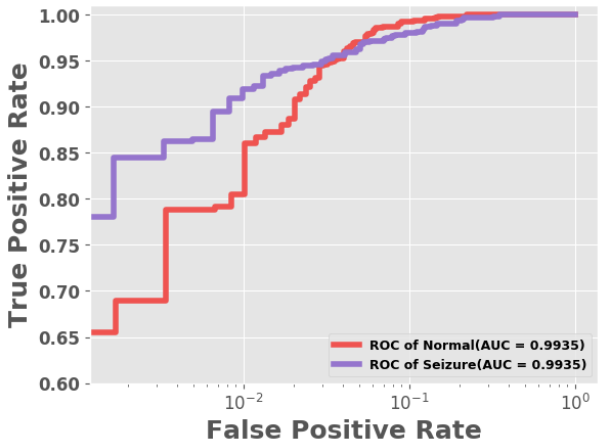
(c) CM of EEG-S



(d) ROC of EEG-S



(e) CM of TUH



(f) ROC of TUH

Figure 6.3: Confusion matrix and ROC curves with AUC scores of each dataset. CM denotes confusion matrix.

deep learning based baselines. Furthermore, we chose some competitive state-of-the-art algorithms for every single task separately.

1) MIR Baselines:

Rashid et al. [292] use Discrete Wavelet Transform (DWT) to extract features and feed into Levenberg-Marquardt Algorithm (LMA) based neural network for motor imagery EEG intention recognition.

Zhang et al. [9] design a joint convolutional recurrent neural network to learn robust high-level feature presentations by low-dimensional dense embeddings from raw MI-EEG signals.

Ma et al. [293] transform the EEG data into a spatial sequence to learn more valuable information through RNN.

Alomari et al. [217] analyze the EEG characteristics by the Coiflets wavelets and manually extract features using different amplitude estimators. The extracted features are inputted into SVM classifier for EEG data recognition.

Sita et al. [294] employ independent component analysis (ICA) to extract features which are fed to a quadratic discriminant analysis (QDA) classifier.

Alomari et al. [295] use wavelet transformation to filter and process EEG signals. Then calculate the Root Mean Square and Mean Absolute Value features for EEG recognition.

Shenoy et al. [220] propose a regularization approach based on shrinkage estimation to handle small sample problem and retain subject-specific discriminative features.

Szczuko [296] design a rough set based classifier for the aim of EEG data classification.

Stefano et al. [297] extract the mu ($7 \sim 13Hz$) and beta ($13 \sim 30Hz$) bands' power spectral density (PSD) as manual features to discriminate different motor imagery

intentions.

Pinheiro et al. [236] adopt a C4.5 decision tree as the classifier to distinguish the manually extracted EEG features such as arithmetic mean and maximum value of the Fourier transform.

Kim et al. [298] use a multivariate empirical mode decomposition to obtain the mu and beta rhythms from the nonlinear EEG signals.

2) PI Baselines:

Ma et al. [299] adopt a CNN structure to automatically extract an individual's best, unique neural features with the aim of person identification.

Yang et al. [300] present an approach for biometric identification using EEG signals based on features extracted with the Hilbert-Huang Transform (HHT).

Rodrigues et al. [301] propose the Flower Pollination Algorithm under different transfer functions to select the best subset of channels that maximizes the accuracy, which is measured using the Optimum-Path Forest classifier.

Frashini et al. [288] decompose EEG signals into standard frequency bands by a band-pass filter and estimate the functional connectivity between the sensors using the Phase Lag Index. The resulting connectivity matrix was used to construct a weighted network for person identification.

Thomas et al. [302] extract sample entropy features from the delta, theta, alpha, beta and gamma bands of 64 channel EEG data, which are evaluated for subject-identification.

3) ND Baselines:

Ziyabari et al. [303] adopt a hybrid deep learning architecture, including LSTM and stacked denoising Autoencoder, that integrates temporal and spatial context to detect the seizure.

Harati et al. [304] demonstrate that a variant of the filter bank-based approach and provides a substantial reduction in the overall error rate.

Zhang et al. [305] extract a list of 24 feature types from the scalp EEG signals and found 170 out of the 2794 features to classify epileptic seizures accurately.

Goodwin et al. [306] combine recent advances in RNN with access to textual data in EEG reports to automatically extracting word- and report-level features and infer underspecified information from EHRs (electronic health records).

Golmohammadi et al. [307] propose a seizure detection method by using hidden Markov models (HMM) for sequential decoding and deep learning networks.

Results Tables 6.2 presents the classification metrics comparison between our approach and well-known baselines (including Non-DL and DL baselines), where DL, AdaB, LDA represent deep learning, Adaptive Boosting, and Linear Discriminant Analysis, respectively. The results show that our approach achieved the highest accuracy on all the datasets. Specifically, the proposed approach achieved the highest accuracy of 0.9632, 0.9984, and 0.9975 on eegmmidb, EEG-S, and TUH dataset, respectively. Further, we conducted an ablation study by comparing our method, which mainly combined selective attention mechanism and CNN, with the solo CNN. It turned out that our approach outperformed CNN, demonstrating the proposed selective attention mechanism improved the distinctive feature learning.We show the confusion matrix and ROC curves (including the AUC scores) of each dataset in Figure 6.3. In Figure 6.3a, 'L', 'R', and 'B' denote left, right, and both, respectively.

Besides, to further evaluate the performance of our model, we compared our framework with 21 state-of-the-art methods which using the same dataset. In particular, we compared with 11 competitive state-of-the-art methods over motor imagery classification and five cutting edges separately over person identification and neurological diagnosis scenarios. Table 6.3 shows the comparison results.

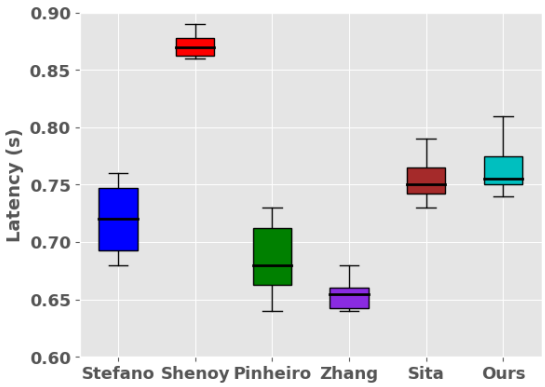


Figure 6.4: Latency comparison

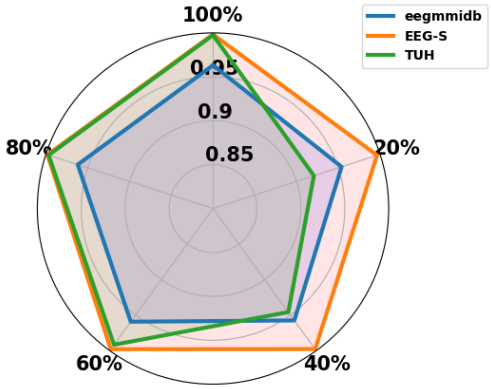


Figure 6.5: Varying # of channels

We could observed that our proposed framework consistently outperformed a set of widely used baseline methods and strong competitors on three different datasets. The performance shows a significant improvement compared with other baselines. These datasets were collected using different EEG hardware, ranging from high-precision medical equipment to off-the-shelf EEG headset with a different number of EEG channels. Regarding the seizure diagnosis in ND, by setting the normal state as impostor while the seizure state as genuine, our approach gained a False Acceptance Rate (FAR) of 0.0033 and a False Rejective Rate (FRR) of 0.0017. This outperformed the existing methods by a large margin [286, 306, 307, 304].

6.3.4 Resilience Evaluation

In this section, we focus on evaluating the resilience of proposed method in coping with various number of EEG signal channels, and incomplete EEG signals.

In practice, the number of EEG channels of EEG devices are diverse due to two reasons. First, different off-the-shelf or on-the-shelf devices have various channels numbers. Intuitively, the quality of signals and the contained information is directly associated with the number of channels. In the meantime, the devices with more channels usually are more expensive and less portable. Second, incomplete

EEG signals cause the degradation of BCI applications. It could happen when some electrical nodes are loosened because of weak maintenance of EEG devices. To investigate the robustness of incomplete EEG signals with missing channels, we also conduct experiments by randomly selecting part of a proportion of signal channels over three datasets. For example, by selecting 20% of channels on the eegmmidb dataset, the selected channel number is $12 = round(64 * 0.2)$. Figure 6.5 shows the experiments results (0.4 denotes the accuracy and 20% denotes the channel percentage used for training). The radar chart demonstrates that eegmmidb and EEG-S, both with 64 channels, can achieve competitive accuracy even with only 20% signal channels. In contrast, TUH (22 channels) is highly dependent on the channel numbers. The reason is that TUH only remains five channels for 20% channel percentage, respectively. According to our experience, the proposed framework requires at least eight EEG channels to achieve high accuracy.

6.3.5 Latency Analysis

Except for the high accuracy of EEG signal classification, the low latency is another critical requirement for the success of real-world BCI applications.

In this section, we take the eegmmidb dataset as an example to compare the latency of the proposed framework with several state-of-the-art algorithms. The results are presented in Figure 6.4. We observed that our approach had competitive latency compared with other methods. The overall latency was less than 1 second. The deep learning based techniques in this section do not explicitly lead to extra latency. One of the main reasons may lie in that the reinforced selective attention has filtered out unnecessary information. To be more specific, the classification latency of the proposed framework was about 0.7$\sim$0.8 seconds, which mainly resulted from the classifying procedure and convolutional mapping. The latency caused by the classifier was around 0.7 seconds. The convolutional mapping only took 0.05 sec on testing although it took more than ten minutes on training.

6.3.6 Reward Model Demonstration

We briefly report the empirical demonstration of the proposed exponential reward model (Section 5.1.3). We compared the proposed reward model in Eq. 5.5 with the traditional reward $r_t = e^{acc}$ over three benchmark datasets (eegmmidb, EEG-S, and TUH). The experiment results show that the novel reward model achieved higher accuracy (0.9632, 0.9984, and 0.9975) than the traditional model (0.9231, 0.9901, and 0.9762).

6.4 Conclusion

This chapter proposes a generic and effective framework for raw EEG signal classification to support the development of BCI applications. The framework works directly on raw EEG data without requiring any preprocessing or feature engineering. Besides, it can automatically select distinguishable feature dimensions for different EEG data, thus achieving high usability. We conduct extensive experiments on three well-known public datasets and one local dataset. The experimental results demonstrate that our approach not only outperforms several state-of-the-art baselines by a large margin but also shows low latency and high resilience in coping with multiple EEG signal channels and incomplete EEG signals. Our approach applies to wider application scenarios such as intention recognition, person identification, and neurological diagnosis.

Part IV

Weakly Supervised Learning

6. Adaptive Cognitive Activity Recognition with Reinforced CNN

This part contains works published in:

[6] **X. Zhang**, L. Yao, and F. Yuan. Adversarial Variational Embedding for Robust Semi-supervised Learning. In *Proceedings of the 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (**KDD 2019**)*. Research Track. Anchorage, Alaska, August 4 - 8, 2019. (CORE Rank A*)

[7] **X. Zhang**, X. Chen, L. Yao, C. Ge, and M. Dong. Deep Neural Network Hyperparameter Optimization with Orthogonal Array Tuning. In *Proceedings of the 26th International Conference On Neural Information Processing (**ICONIP 2019**)*. Sydney, Australia, December 12-15, 2019. (CORE Rank A)

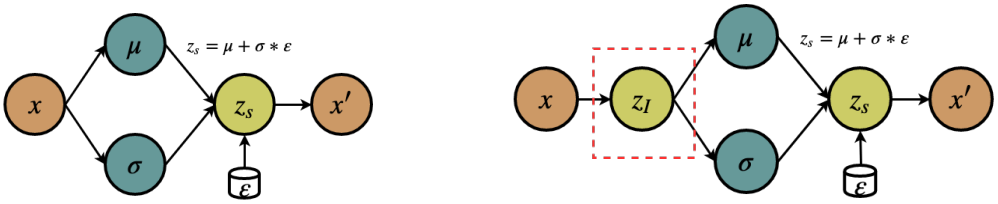
Chapter 7

Adversarial Variational Embedding for Robust Semi-supervised Learning

In most of the real-world scenarios, it is expensive to get the accurate label of the collected EEG signals. Thus, in this chapter, we propose a weakly supervised representation learning frameworks where the the distribution of the labeled data can be enhanced by the distribution of the unlabeled observations.

Semi-supervised learning from data is one of the fundamental challenges in artificial intelligence, which considers the problem when only a subset of the observations has corresponding class labels [308]. This issue is of immense practical interest in a broad range of application scenarios, such as abnormal activity detection [309], neurological diagnosis [310], and computer vision [311]. In these scenarios, it is easy to obtain abundant observations but expensive to gather the corresponding class labels. Among existing approaches, Variational Autoencoders (VAEs) [312, 313] have recently achieved state-of-the-art performance in semi-supervised learning.

VAE models provide a general framework for learning latent representations: a model is specified by a joint probability distribution both over the data and over



(a) Standard VAE (b) VAE++

Figure 7.1: Comparison of the standard VAE and the proposed VAE++. $\boldsymbol{x}$ and $\boldsymbol{x'}$ denote the input and the reconstructed data. $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ denote the learned expectation and standard deviation, $\boldsymbol{z}_s$ denotes the stochastically sampled latent representation which is composed by $\boldsymbol{\mu}$, $\boldsymbol{\sigma}$, and $\boldsymbol{\varepsilon}$, where $\boldsymbol{\varepsilon}$ is randomly sampled from $\mathcal{N}(0, 1)$. In standard VAE, $\boldsymbol{z}_s$ is regarded as the learned representation while, in VAE++, $\boldsymbol{z}_I$ denotes the proposed exclusive latent representation which can be used for classification.

latent random variables, and a representation can be found by considering the posterior on latent variables given specific data [314]. The learned representations can not only be used for generation but also for classification. For instance, VAE provides a latent feature representation of the input observations, where a separate classifier can be thereafter trained using these representations. The high quality of latent representations enables accurate classification, even with a limited number of labels. A number of studies have applied VAE in semi-supervised classification in the computer vision area [312, 315, 314].

Why we propose the VAE++ . One major challenge faced by the existing VAE-based semi-supervised methods is that the latent representations are stochastically sampled from the prior distribution instead of being directly rendered from the explicit observations. In particular, as shown in Figure 7.1a, the learned latent representations $\boldsymbol{z}_s$ are *randomly sampled* from a multivariate Gaussian distribution (see Equation 7.1). Thus, for a specific sample, the corresponding latent representation is not exclusive (i.e., the representation is not repeatable in different runnings), which makes it inappropriate for classification. To solve this problem, in the latent space, we propose a new variable $\boldsymbol{z}_I$ (see Figure 7.1b) which is directly learned from the input data. The exclusive latent code $\boldsymbol{z}_I$ is guaranteed to keep invariant for

a specific input $\boldsymbol{x}$ in different runnings. The modified VAE is called VAE++. In addition, the learned expectation $\boldsymbol{\mu}$ only contains a part of information of the input observations, which is not enough to represent the observations in classification task, even though $\boldsymbol{\mu}$ is exclusive[1]. The comparison of performance among $\boldsymbol{z}_I$, $\boldsymbol{z}_s$ and $\boldsymbol{\mu}$ will be presented in Section 7.3.

Why VAE++ needs the semi-supervised GAN. In the proposed VAE++, it is necessary to reduce the information loss between the two latent representations $\boldsymbol{z}_I$ and $\boldsymbol{z}_s$ to guarantee the learned $\boldsymbol{z}_I$ is representative. The commonly used constraints between two distributions (e.g., Kullback-Leibler divergence) can only utilize the information of the observations but fail to exploit the information of labels. In this paper, we use a novel approach to take advantage of both unlabelled and labelled data by jointly training the VAE++ and a semi-supervised GAN.

Why semi-supervised GAN needs the VAE++. GAN based approaches [316, 317] have recently shown promising results in semi-supervised learning. The semi-supervised GAN trains a generative model and a discriminator with inputs belonging to one of K classes. Different from the regular GAN, the semi-supervised GAN requires the discriminator to make a $K+1$ class prediction with an extra class added, corresponding to the generated fake samples. In this way, the observations' properties can be used to improve decision boundaries and allow for more accurate classification than using the labelled data alone. However, the generated samples are sampled from pre-defined distribution (e.g., Gaussian noise) [318]. Such pre-defined prior distributions are often independent from the input data distributions and may obstruct the convergence and can not guarantee the distribution of the generated data. This drawback can be amended by gearing with VAE++ which can provide a meaningful prior distribution that can represent the distribution of the input data.

We introduce a recipe for semi-supervised learning, a robust Adversarial Variational Embedding (AVAE) framework, which learns the exclusive latent representations by combining VAE and semi-supervised GAN. To utilize the generative ability of GAN

[1]For the same reason, $\boldsymbol{\sigma}$ can not be used as the exclusive code.

and the distribution approximating power of VAE, the proposed approach employs GAN to encourage VAE for the aim of learning the more robust and informative latent code. We present the framework in the context of VAE, adding a new exclusive code in latent space which is directly rendered from the data space. The generator in VAE++ also works as a generator of GAN. Both the exclusive code (marked as real) and the generated representation (marked as fake) are fed into the discriminator in order to force them to have similar distribution [319].

Although a small set of models combining VAE and GAN have been previously explored, they are all focused on the generation perspective. To our knowledge, we are in the first batch of work that focuses on classification by aggregating VAE and GAN. We mark the following contributions:

- We present a novel semi-supervised Adversarial Variational Embedding approach to harness the deep generative model and generative adversarial networks collectively under a trainable unified framework. The reproducible codes and datasets are publicly available[2].
- We propose a new structure, VAE++, to automatically learn an exclusive latent code for accurate classification. A novel semi-supervised GAN, which exploits both the unlabelled data distribution and categorical information, is proposed to gear with the VAE++ in order to encourage the VAE++ to learn a more effective and robust exclusive code.
- We evaluate the proposed approach over four real-world applications (activity reconstruction, neurological diagnosis, image classification, and recommender system). The results demonstrate that our approach outperforms all the state-of-the-art methods.

[2]https://github.com/xiangzhang1015/Adversarial-Variational-Semi-supervised-Learning

7.1 Related Work

There are a host of studies that have been investigated to apply VAE for semi-supervised learning [312, 314, 313, 320]. [312] explores semi-supervised learning with deep generative models by building two VAE-based deep generative models for latent representation extraction. Afterward, [314] attempts to learn disentangled representations that encode distinct aspects of the data into separate variables. However, in all the existing semi-supervised VAE models, the learned representations do not only depend on the posterior distribution but also on the latent random variables. It is necessary that learning the exclusive code which is only related to the posterior distribution for the specified data.

Another recent arising semi-supervised method is semi-supervised GAN [316, 321] . SGAN [316] extends GAN to the semi-supervised context by forcing the discriminator network to output class labels. The CatGAN [321] modifies the objective function to take into account the mutual information between observed examples and their predicted class distributions. In the above methods, the generator chooses simple factored continuous noise which is independent from the input data distribution, for generation. As a result, it is possible that the noise will be used by the generator in a highly entangled way, increasing the difficulty to control the distribution of the generated data. Conditional GAN [319] and InfoGAN address this drawback by utilizing external information (e.g., categorical information) as a restriction, but they both pay attention to generation or supervised classification and have limited help in semi-supervised classification.

Despite the few works attempting to combine VAE and GAN [315], most of them focus on generation instead of classification. For example, the VAE/GAN and CVAE-GAN employ the standard VAE to share the encoder with the generator of GAN in order to generate new observations. For semi-supervised classification, we care about the latent code instead of the observations. The Adversarial Autoencoder (AAE [315]) integrates VAE and GAN but only employs GAN to replace KL di-

vergence as a penalty to impose a prior distribution on the latent code, which is a totally different direction from our work.

Summary. Unlike the existing VAE- and GAN-based studies, the proposed model 1) focuses on semi-supervised classification instead of generation; 2) attempts to learn an exclusive latent representation instead of a stochastic sampled representation; 3) works on improvement of latent space instead of data space. Moreover, the semi-supervised GAN in our work partly adopts the improved GAN [317], but there are a number of key differences: 1) [317] adopts the semi-supervised strategy for classification while we adopt this strategy as a constraint to reduce information loss in the transformation from z_I to z_s in order to force the proposed AVAE to learn a more robust and effective latent code; 2) [317] employs the discriminator of GAN as the classifier while we adopt an extra non-parametric classifier since the former has poor performance in our case (take the PAMAP2 dataset as an example, [317] and our model achieve the accuracy around 65% and 85%, respectively); 3) we employ weighted loss function to balance the significance of the unlabelled and labelled observations.

7.2 Methodology

Suppose the input dataset has two subsets, one of which contains labelled samples while the other contains unlabelled samples. In the former subset, the observations appear as pairs $(\boldsymbol{X}^L, \boldsymbol{Y}^L) = \{(\boldsymbol{x}_1^L, \boldsymbol{y}_1), (\boldsymbol{x}_2^L, \boldsymbol{y}_2), \cdots, (\boldsymbol{x}_{N_L}^L, \boldsymbol{y}_{N_L})\}$ with the i-th observation $\boldsymbol{x}_i^L \in \mathbb{R}^M$ and the corresponding one-hot label $\boldsymbol{y}_i \in \mathbb{R}^K$ where K denotes the number of classes. N_L denotes the number of labelled observations while M denotes the number of the observation dimensions. In the latter subset, only the observations $\boldsymbol{X}^U = \{\boldsymbol{x}_1^U, \boldsymbol{x}_2^U, \cdots, \boldsymbol{x}_{N_U}^U\}$ are available and N_U denotes the number of unlabelled observations $\boldsymbol{x}_i^U \in \mathbb{R}^M$. The total data size N equals to the sum of N_L and N_U. In terms of effective classification, we attempt to learn a latent representation with distinguishable information. Then the learned representations can be fed

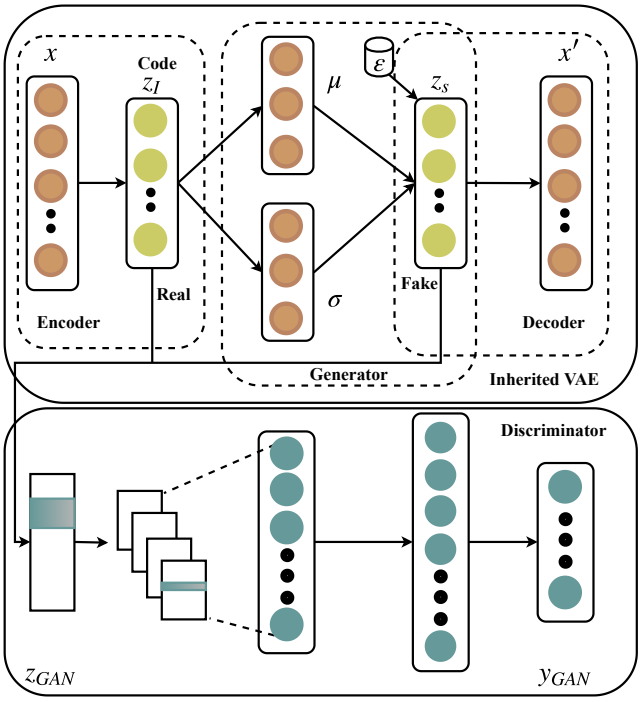


Figure 7.2: AVAE is composed of VAE++ and a semi-supervised GAN. The generated $\boldsymbol{z}_s$ (labelled as fake) and the exclusive code $\boldsymbol{z}_I$ (labelled as real) are fed into the discriminator. The discriminator can exploit both the labelled and unlabelled observations. The generator in VAE++ also works as a generator of GAN.

into a classifier for recognition. In this chapter, we mainly focus on the latent code learning.

In the semi-supervised learning, due to the lack of labelled observations, it is significant to learn latent variable distribution based on the observations without label[3]. Thus, we are required to build an encoder to provide an embedding or feature representation which allows accurate classification even with limited observations.

[3]For simplification, we omit the index and directly use variable $\boldsymbol{x}$ to denote observations.

7.2.1 VAE++

The VAE is demonstrated to provide a latent feature representation for semi-supervised learning [312, 314], compared to a linear embedding method or a regular autoencoder. The VAE maps the input observation $\boldsymbol{x}$ to a compressed code $\boldsymbol{z}_s$, and decodes it to reconstruct the observation. The latent representation is calculated through the reparameterization trick [55]:

$$\boldsymbol{z}_s = \mu_{\boldsymbol{x}} + \sigma_{\boldsymbol{x}} * \boldsymbol{\varepsilon} \tag{7.1}$$

with $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, 1)$ to impose the posterior distribution of the latent code on $p(\boldsymbol{z}_s|x) \sim \mathcal{N}(\mu_{\boldsymbol{x}}, \sigma^2_{\boldsymbol{x}})$. $\mu_{\boldsymbol{x}}$ and $\sigma_{\boldsymbol{x}}$ denote the expectation and standard deviation of the posterior distribution of $\boldsymbol{z}_s$, which are learned from $\boldsymbol{x}$. For the efficient generation and reconstruction, VAE imposes the code $\boldsymbol{z}_s$ on a prior Gaussian distribution:

$$\bar{p}(\boldsymbol{z}_s) = \mathcal{N}(\boldsymbol{z}_s|\boldsymbol{0}, \boldsymbol{I}) \tag{7.2}$$

Through minimizing the reconstruction error between $\boldsymbol{x}$ and $\boldsymbol{x}'$ and restricting the distribution of $\boldsymbol{z}_s$ to approximate the prior distribution $\bar{p}(\boldsymbol{z}_s)$, VAE is supposed to learn the representative latent code $\boldsymbol{z}_s$ which can be used for classification or generation.

Due to the strong feature representation ability, VAE has been employed for feature extraction and semi-supervised learning [322, 314]. However, one limitation of the standard VAE is that the learned latent code $\boldsymbol{z}_s = g(\mu_{\boldsymbol{x}}, \sigma_{\boldsymbol{x}}, \boldsymbol{\varepsilon})$, as shown in Equation (7.1), is not exclusive. In other words, for a specific observation $\boldsymbol{x}$ and a fixed embedding model $p(\boldsymbol{z}_s|\boldsymbol{x})$, the corresponding latent code $\boldsymbol{z}_s$ is not exclusive as it contains a stochastic variable $\boldsymbol{\varepsilon}$ which is randomly sampled from the prior distribution $\bar{p}(\boldsymbol{z}_s)$. For instance, in a pre-trained fixed VAE encoder, the specific input $\boldsymbol{x}$ will lead to a variety of $\boldsymbol{z}_s$ in different running. At high level, the latent code $\boldsymbol{z}_s$ is determined by two factors: the prior distribution of observation $\bar{p}(\boldsymbol{x})$ which affects $\boldsymbol{z}_s$ through the learned $\mu_{\boldsymbol{x}}$ and $\sigma_{\boldsymbol{x}}$, and the stochastically sampled data $\boldsymbol{\varepsilon}$. However, the stochastically sampled latent code is unstable and will corrupt the

features for classification. Furthermore, the posterior distribution of $\boldsymbol{z}_s$ is forced to approximate the manually set prior distribution (commonly Normal Gaussian distribution), which inevitably leads to information loss.

In order to completely sidestep the above-mentioned issue, we propose a novel VAE++ model to learn an exclusive latent code $\boldsymbol{z}_I$. The VAE++ contains three key components: the encoder, the generator, and the decoder (see Figure 7.2). The encoder transforms the observation into a latent code $\boldsymbol{z}_I \in \mathbb{R}^D$ which is directly determined by the input $\boldsymbol{x}$. D denotes the dimension of $\boldsymbol{z}_I$. We learn the:

$$p_{\boldsymbol{\theta}_{en}}(\boldsymbol{z}_I|\boldsymbol{x}) = f(\boldsymbol{z}_I; \boldsymbol{x}, \boldsymbol{\theta}_{en}) \tag{7.3}$$

where f denotes a non-linear transformation while $\boldsymbol{\theta}_{en}$ denotes encoder parameters. The non-linear transformation f is generally chosen as a deep neural network for the excellent ability of non-linear approximation. Then, in the generator, we measure the expectation $\mu(\boldsymbol{z}_I)$ and the standard derivation $\sigma(\boldsymbol{z}_I)$ from the latent code $\boldsymbol{z}_I$ and update Equation (7.1). The generated variable $\boldsymbol{z}_s$ can be calculated by:

$$\boldsymbol{z}_s = \mu(\boldsymbol{z}_I) + \sigma(\boldsymbol{z}_I) * \boldsymbol{\varepsilon} \tag{7.4}$$

At last, the decoder is employed to reconstruct the sample:

$$p_{\boldsymbol{\theta}_{de}}(\boldsymbol{x}'|\boldsymbol{z}_s) = f'(\boldsymbol{x}'; \boldsymbol{z}_s, \boldsymbol{\theta}_{de}) \tag{7.5}$$

where f' denotes another non-linear rendering, called decoder, with parameters $\boldsymbol{\theta}_{de}$ and $\boldsymbol{x}'$ denotes the reconstructed observation.

The loss function of VAE++ can be calculated by:

$$\begin{aligned} \mathcal{L}_{\text{VAE}} = & -\mathbb{E}_{\boldsymbol{z}_s \sim p_{\boldsymbol{\theta}_{en}}(\boldsymbol{z}_s|\boldsymbol{x})}[\log p_{\boldsymbol{\theta}_{de}}(\boldsymbol{x}'|\boldsymbol{z}_s)] \\ & + KL(p_{\boldsymbol{\theta}_{en}}(\boldsymbol{z}_s|\boldsymbol{x})||\bar{p}(\boldsymbol{z}_s)) \end{aligned} \tag{7.6}$$

The first component is the reconstruction loss, which equals to the expected negative log-likelihood of the observation. This term encourages the decoder to reconstruct

the observation $\boldsymbol{x}$ based on the sampling code $\boldsymbol{z}_s$ which is under Gaussian distribution. The lower reconstruction error indicates the encoder learned a better latent representation. The second component is the Kullback-Leibler divergence which measures the distance between the prior distribution of the latent code $\bar{p}(\boldsymbol{z}_s)$ and the posterior distribution $p(\boldsymbol{z}_s|\boldsymbol{x})$. This divergence reflects the information loss when we use $p(\boldsymbol{z}_s|\boldsymbol{x})$ to represent $\bar{p}(\boldsymbol{z}_s)$.

In the latent space of the novel VAE++, there are two compressed informative codes $\boldsymbol{z}_I$ and $\boldsymbol{z}_s$. The former represents directly-encoded $\boldsymbol{x}$ whilst the latter is stochastically sampled from the posterior distribution, which makes the former more suitable for classification. Therefore, we choose $\boldsymbol{z}_I$ as the compressed latent code in VAE++ instead of the $\boldsymbol{z}_s$ in standard VAE.

From equation (7.4), we can observe that the expectation and standard deviation of $\boldsymbol{z}_s$ and $\boldsymbol{z}_I$ are invariant. In particular, for a specific sample $\boldsymbol{x}_i$, the corresponding $\boldsymbol{z}_{si}$ and $\boldsymbol{z}_{Ii}$ have the same statistical characteristics. Thus, we have

$$\boldsymbol{z}_s \leftarrow \mu(\boldsymbol{z}_I), \sigma(\boldsymbol{z}_I), \boldsymbol{\varepsilon} \tag{7.7}$$

which indicates that the generated $\boldsymbol{z}_s$ is affected by both the distribution of $\boldsymbol{z}_I$ and the prior distribution $\bar{p}(\boldsymbol{z}_s)$ (or $\boldsymbol{\varepsilon}$). In summary, the $\boldsymbol{z}_s$ inherits the statistical characteristics of $\boldsymbol{z}_I$.

7.2.2 Adversarial Variational Embedding

One significant sufficient condition of a well-trained VAE++ is less information loss in the transformation from $\boldsymbol{z}_I$ to $\boldsymbol{z}_s$ to guarantee the learned $\boldsymbol{z}_I$ is representative. As mentioned before, the information in $\boldsymbol{z}_s$ is partly inherited from $\boldsymbol{z}_I$ and the other part is randomly sampled from the prior distribution $\bar{p}(\boldsymbol{z}_s)$. Since the conditional distribution $p_{\boldsymbol{\theta}_{en}}(\boldsymbol{z}_I|\boldsymbol{x})$ has a better description of the input observation $\boldsymbol{x}$, we attempt to increase the proportion of inherited part and decrease the proportion of stochastically sampled part.

As shown in Figure 7.2, in the proposed AVAE the generator $\boldsymbol{G}$ generates $\boldsymbol{z}_s$ based on the joint probability $p(\mu, \sigma, \bar{p}(\boldsymbol{z}_s))$ instead of the noise in standard GAN. The $\boldsymbol{z}_s$ is regarded as 'fake' while $\boldsymbol{z}_I$ is marked as 'real'. Specifically, for the labelled observations $\boldsymbol{x}^L$, VAE++ encodes the input to the latent code $\boldsymbol{z}_I^L \in \mathbb{R}^D$ and generates $\boldsymbol{z}_s^L \in \mathbb{R}^D$; similarly, for unlabelled observations $\boldsymbol{x}^U$, we have $\boldsymbol{z}_I^U \in \mathbb{R}^D$ and generates $\boldsymbol{z}_s^U \in \mathbb{R}^D$. To exploit the information of the labels, we extend the $\boldsymbol{y} \in \mathbb{R}^K$ which has K possible classes to $\boldsymbol{y}_{GAN} \in \mathbb{R}^{K+1}$ which has $K+1$ possible classes by regarding the generated fake samples $\boldsymbol{z}_s$ as the $(K+1)$-th class [317, 316]. In the VAE++, the unspecified $\boldsymbol{z}_s$ denotes both $\boldsymbol{z}_s^L$ and $\boldsymbol{z}_s^U$ whenever we don't care whether the observation is labelled or not. This rule also applies to $\boldsymbol{z}_I$. Similarly, we use $\boldsymbol{z}_{GAN}$ to denote the input of the discriminator $\boldsymbol{D}$, which contains both $\boldsymbol{z}_I$ and $\boldsymbol{z}_s$. The discriminator can be described by

$$\boldsymbol{q}_{\boldsymbol{\varphi}}(\boldsymbol{y}_{GAN}|\boldsymbol{z}_{GAN}) = h(\boldsymbol{y}_{GAN}; \boldsymbol{z}_{GAN}, \boldsymbol{\varphi}) \tag{7.8}$$

where $\boldsymbol{\varphi}$ denotes the parameters of $\boldsymbol{D}$ while h denotes the non-linear transformation which is implemented by a Convolutional Neural Networks (CNN) [290]. Therefore, we can use $\boldsymbol{q}_{\boldsymbol{\varphi}}(\boldsymbol{y}_{GAN} = K+1|\boldsymbol{z}_{GAN})$ to supply the probability where $\boldsymbol{z}_{GAN}$ is fake (from $\boldsymbol{z}_s$) and use $\boldsymbol{q}_{\boldsymbol{\varphi}}(\boldsymbol{y}_{GAN}|\boldsymbol{z}_{GAN}, \boldsymbol{y}_{GAN} < K+1)$ to supply the probability where $\boldsymbol{z}_{GAN}$ is real ((from $\boldsymbol{z}_I$)) and is correctly classified.

For the labelled input, same as supervised learning, the discriminator is supposed to not only tell whether the input $\boldsymbol{z}_{GAN}$ is real or generated, but also classify it into the correct class. Therefore, we have the supervised loss function

$$\mathcal{L}_{label} = -\mathbb{E}_{\boldsymbol{z}_{GAN}, \boldsymbol{y}_{GAN} \sim p_j}[log \boldsymbol{q}_{\boldsymbol{\varphi}}(\boldsymbol{y}_{GAN}|\boldsymbol{z}_{GAN}, \boldsymbol{y}_{GAN} < K+1)] \tag{7.9}$$

where p_j denotes the joint probability.

For the unlabelled input, we only require the discriminator to perform a binary classification: the input is real or fake. The former probability can be calculated by $1 - \boldsymbol{q}_{\boldsymbol{\varphi}}(\boldsymbol{y}_{GAN} = K+1|\boldsymbol{z}_{GAN})$ whilst the latter can be calculated by $\boldsymbol{q}_{\boldsymbol{\varphi}}(\boldsymbol{y}_{GAN} = K+1|\boldsymbol{z}_{GAN})$.

Thus, the unsupervised loss function:

$$\mathcal{L}_{unlabel} = -\mathbb{E}_{\boldsymbol{z}_{GAN}\sim p_{\boldsymbol{\theta}_{en}}(\boldsymbol{z}_I|\boldsymbol{x})}[log(1 - \boldsymbol{q}_{\boldsymbol{\varphi}}(\boldsymbol{y}_{GAN} = K + 1|\boldsymbol{z}_{GAN}))]$$
$$- \mathbb{E}_{\boldsymbol{z}_{GAN}\sim p_{\boldsymbol{\theta}_{en}}(\boldsymbol{z}_s|\boldsymbol{x})}[log(\boldsymbol{q}_{\boldsymbol{\varphi}}(\boldsymbol{y}_{GAN} = K + 1|\boldsymbol{z}_{GAN}))]$$

In summary, the final loss function of the discriminator

$$\mathcal{L}_{GAN} = w_1 * flag * \mathcal{L}_{label} + w_2 * (1 - flag) * \mathcal{L}_{unlabel} \tag{7.10}$$

where w_1, w_2 are weights and $flag$ is a switch function

$$flag = \begin{cases} 1 & labelled \\ 0 & unlabelled \end{cases}$$

If the specific observation is labelled, we calculate the labelled loss function. Otherwise, we calculate the unlabelled loss function. From empirical experiments, we observe that the $\mathcal{L}_{unlabel}$ is much easier to converge than $\mathcal{L}_{label}$ and the real/fake classification accuracy is much higher than the K classes classification accuracy. To encourage the optimizer to focus on the former part which is more difficult to converge, we set $w_1 = 0.9$ and $w_2 = 0.1$.

The discriminator receives $\boldsymbol{z}_{GAN}$ as input and extracts the dependencies through CNN filters. Two fully connected layers follow the convolutional layer for dimension reduction. At last, a softmax layer is employed to work on the low-dimension features to estimate the log normalization of the categorical probability distribution which is output as $\boldsymbol{y}_{GAN}$.

The overall aim of the proposed AVAE (as described in Algorithm 2) is to train a robust and effective semi-supervised embedding method. The VAE loss $\mathcal{L}_{VAE}$ and the GAN loss $\mathcal{L}_{GAN}$ are trained simultaneously by the Adam optimizer. After convergence, the compressed representative code $\boldsymbol{z}_I$ is fed into a non-parametric nearest neighbors classifier for recognition.

ALGORITHM 2: Adversarial Variational Embedding

Input: labelled observations ($\boldsymbol{X}^L$, $\boldsymbol{Y}^L$) and unlabelled observations $\boldsymbol{X}^U$

Output: Representation $\boldsymbol{z}_I$

Initialize network parameters $\boldsymbol{\theta}_{en}$, $\boldsymbol{\theta}_{de}$, $\boldsymbol{q}_{\boldsymbol{\varphi}}$
for $\boldsymbol{x} \in \{\boldsymbol{X}^L, \boldsymbol{X}^U\}$ **do**
    $\boldsymbol{z}_I \leftarrow \boldsymbol{x}$
    $\boldsymbol{\mu}, \boldsymbol{\sigma} \leftarrow \boldsymbol{z}_I$
    Sampling $\boldsymbol{\varepsilon}$ from $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$
    $\boldsymbol{z}_s = \mu(\boldsymbol{z}_I) + \sigma(\boldsymbol{z}_I) * \boldsymbol{\varepsilon}$
    $\boldsymbol{x}' \leftarrow \boldsymbol{z}_s$
    $\mathcal{L}_{VAE} \leftarrow \boldsymbol{x}, \boldsymbol{x}', p(\boldsymbol{z}_s|\boldsymbol{x})$
    for $z_I, z_s, \boldsymbol{y} \in \boldsymbol{Y}^L$ **do**
        $\boldsymbol{y}_{GAN} \leftarrow z_I, z_s$
        $\mathcal{L}_{GAN} \leftarrow \boldsymbol{y}_{GAN}, \boldsymbol{y}$
    end for
    Minimize $\mathcal{L}_{VAE}$ and $\mathcal{L}_{GAN}$
end for
return $\boldsymbol{z}_I$

7.3 Experiments

In this section, we demonstrate the effectiveness and validation of the proposed method over four applications.

Table 7.1: Overall comparison of semi-supervised classification accuracy (%) on activity recognition. All the baselines and our approach are working on the same dataset and sharing the same experiment settings for each specific application.

Dataset	Rate (%)	Algorithm-related State-of-the-art				Application-related State-of-the-art				Ablation Study			Ours
		M2	AAE	LVAE	ADGM	[323]	[324]	[325]	[4]	VAE (μ)	VAE	VAE++	AVAE
Activity Recognition (PAMAP2)	20	64.83±0.16	63.67±0.23	69.82±0.69	67.31±0.45	72.31±0.16	70.95±0.08	67.31±0.14	76.68±0.31	58.43±0.13	76.51±0.53	78.12±0.55	78.63±0.38
	40	68.92±0.23	76.83±0.25	76.43±0.19	78.21±0.38	80.51±0.21	75.38±0.12	77.28±0.21	80.15±0.16	62.74±0.12	78.78±0.22	80.88±0.38	81.37±0.29
	60	72.35±0.21	77.39±0.19	78.69±0.27	79.34±0.29	80.29±0.21	76.89±0.05	79.69±0.15	82.49±0.33	67.85±0.08	79.63±0.29	81.94±0.19	84.91±0.17
	80	75.88±0.35	78.28±0.11	81.41±0.23	80.38±0.16	82.12±0.16	79.95±0.18	81.65±0.09	83.56±0.11	73.43±0.06	81.75±0.17	82.08±0.26	85.56±0.21
	100	77.59±0.17	80.79±0.14	84.39±0.18	83.66±0.16	83.64±0.12	81.96±0.11	82.38±0.13	84.59±0.24	76.85±0.00	82.37±0.25	83.29±0.18	86.41±0.06

Note: If the compared method can not deal with unsupervised samples, it will be trained only by the supervised samples.

Table 7.2: Overall comparison of semi-supervised classification accuracy (%) on neurological diagnosis

Dataset	Rate (%)	Algorithm-related State-of-the-art				Application-related State-of-the-art				Ablation Study			Ours
		M2	AAE	LVAE	ADGM	[303]	[304]	[121]	[306]	VAE (μ)	VAE	VAE++	AVAE
Neurological Diagnosis (TUH)	20	71.28±0.16	80.13±0.95	82.31±0.19	86.32±0.12	87.66±0.23	86.38±0.36	82.19±0.24	86.33±0.21	80.58±0.69	86.37±0.24	0.86±0.53	93.69±0.16
	40	75.32±0.16	82.95±0.26	84.38±0.16	86.99±0.05	89.25±0.19	91.58±0.35	84.21±0.08	89.25±0.34	81.35±0.24	89.69±0.27	91.28±0.25	94.32±0.28
	60	76.32±0.29	86.21±0.52	87.51±0.26	87.65±0.16	91.28±0.37	92.58±0.26	85.36±0.32	90.38±0.24	82.59±0.63	90.58±0.27	92.87±0.31	95.21±0.21
	80	79.65±0.37	88.53±0.28	89.56±0.25	88.05±0.12	92.59±0.26	93.25±0.31	85.16±0.24	91.59±0.16	83.21±0.21	91.69±0.35	93.96±0.28	97.86±0.26
	100	82.59±0.31	89.58±0.25	90.25±0.21	88.65±0.26	93.32±0.18	94.29±0.25	86.42±0.26	92.4±0.25	84.21±0.65	92.38±0.41	94.65±0.24	98.13±0.32

7.3.1 Activity Recognition

Experiment Setup

Activity recognition is an important area in data mining. We evaluate our approach over the well-known PAMAP2 dataset [326], which is collected by 9 participants (8 males and 1 female) aged 27 ± 3. We select 5 most commonly used activities (Cycling, standing, walking, lying, and running, labelled from 0 to 4) as a subset for evaluation. For each subject, there are 12,000 instances. The activity is measured by 3 Inertial Measurement Units (IMU) attached to the participants' wrist, chest, and the outer ankle. Each IMU includes 13 dimensions: two 3-axis accelerometers, one 3-axis gyroscopes, one 3-axis magnetometers and one thermometer. The experiments are performed by a Leave-One-Subject-Out strategy to ensure the practicality.

The time window is set as 10 with 50% overlapping. The dataset is split into a training set (80% proportion) and a testing set (20% proportion). For semi-supervised learning, the training dataset contains both labelled observations and unlabelled observations. We present a term called'supervision rate' as a handle on

the relative weight between the supervised and unsupervised terms. For the given number of labelled observations N^L and the number of unlabelled observations N^U, the supervision rate γ is defined by $N^L/(N^L + N^U)$.

Parameter Setting

We introduce the default parameter settings and the settings in other applications keep the same if not mentioned. The input observations are first normalized by Z-score normalization and fed to the input layer of the unsupervised VAE++. The neuron amount in the first hidden layer, which is denoted by $\boldsymbol{z}_I$, is a quarter of M. The second hidden layer contains 2 components which represent the expectation and the standard deviation respectively. The third hidden layer $\boldsymbol{z}_s$ has the sample shape with $\boldsymbol{z}_I$. An Adam optimizer with a learning rate of 0.00001 is employed to minimize the loss function of VAE++.

After each epoch of VAE++, the first hidden layer $\boldsymbol{z}_I$ and the third hidden layer $\boldsymbol{z}_s$ are labelled as 'real' and 'fake', respectively, and fed to the discriminator $\mathcal{D}$. The discriminator contains one convolutional layer followed by two fully-connected layers. There is a softmax layer to obtain the categorical probability before the output layer which has $K + 1$ neurons. The convolutional layer has 10 filters which have shape $[2, 2]$ and the stride size $[1, 1]$. The padding method of the convolutional operation is set as 'same' while the activation function is ReLU. The following hidden layer has $M/4$ neurons and the sigmoid activation function. The loss function is optimized by Adam update rule with learning rate of 0.0001. The object functions of the VAE++ and the discriminator are trained simultaneously. After the convergence, the semi-supervised learned latent representation $\boldsymbol{z}_I$ is fed into a supervised non-parametric nearest neighbor classifiers with $k = 3$.

Table 7.3: Overall comparison of semi-supervised classification accuracy (%) on image classification

Dataset	Rate (%)	Algorithm-related State-of-the-art				Application-related State-of-the-art				Ablation Study			Ours
		M2	AAE	LVAE	ADGM	[316]	[321]	[327]	[328]	VAE (μ)	VAE	VAE++	AVAE
Image Classification (MNIST)	20	93.22±0.62	90.25±0.25	93.25±0.26	89.61±0.27	95.23±0.34	94.25±0.13	94.58±0.25	92.96±0.28	91.58±0.24	92.31±0.53	93.59±0.31	95.12±0.19
	40	93.25±0.34	93.21±0.23	93.28±0.46	91.58±0.25	95.27±0.53	95.56±0.08	95.21±0.26	93.21±0.56	93.65±0.21	94.21±0.19	94.68±0.28	96.43±0.35
	60	96.24±0.51	96.35±0.27	95.34±0.21	93.21±0.34	96.38±0.22	96.54±0.08	96.48±0.32	96.28±0.57	94.89±0.21	95.34±0.14	96.42±0.25	97.21±0.21
	80	98.19±0.25	95.32±0.37	96.11±0.52	95.01±0.15	97.82±0.11	97.21±0.13	97.86±0.34	97.63±0.15	96.78±0.25	97.63±0.15	98.71±0.16	99.79±0.12
	100	98.65±0.21	0.98.25±0.61	96.35±0.26	95.38±0.82	99.21±0.26	98.64±0.27	99.06±0.22	98.53±0.17	97.41±0.18	98.35±0.09	99.67±0.23	99.85±0.11

Table 7.4: Overall comparison of semi-supervised classification accuracy (%) on recommender system

Dataset	Rate (%)	Algorithm-related State-of-the-art				Application-related State-of-the-art				Ablation Study			Ours
		M2	AAE	LVAE	ADGM	[329]	[330]	[331]	[332]	VAE (μ)	VAE	VAE++	AVAE
Recommender System (Yelp)		66.42±0.17	58.27±0.35	66.35±0.36	54.27±0.38	40.55±0.27	47.58±0.36	65.99±0.62	66.21±0.24	64.28±0.12	64.39±0.62	65.58±0.37	70.19±0.87
	20	69.36±0.37	61.55±0.62	68.16±0.24	55.35±0.26	40.28±0.32	48.65±0.27	67.53±0.31	66.59±0.29	64.37±0.25	67.23±0.95	71.05±0.29	72.21±0.35
	40	72.58±0.19	62.15±0.39	68.59±0.93	57.63±0.23	42.15±0.16	50.95±0.24	66.58±0.29	67.95±0.38	67.56±0.35	69.58±0.37	72.19±0.62	75.34±0.35
	60	72.39±0.64	62.89±0.62	74.28±0.37	58.34±0.15	43.21±0.15	52.15±0.38	67.65±0.31	68.23±0.15	69.25±0.18	71.39±0.56	73.21±0.58	78.54±0.38
	80	74.58±0.62	63.51±0.86	72.59±0.36	59.58±0.23	45.86±0.22	54.10±0.12	68.03±0.17	70.61±0.25	73.24±0.68	73.28±0.69	76.53±0.28	79.38±0.59

Baselines

To measure the effectiveness of the proposed method, we compare it with a set of competitive state-of-the-art models. The state-of-the-art methods are composed of two categories: algorithm-related and application-related. The former denotes other VAE/GAN based semi-supervised classification algorithms, which are the same for all the applications. The comparison is used to demonstrate our framework has the highest semi-supervised representation learning ability. The latter denotes the state-of-the-art models in each application, which are varied for the different applications. The comparison is used to demonstrate our work is effective in the real-world scenarios.

The algorithm-related semi-supervised learning solutions in our comparison are listed as follows:

- M2. [312] proposes a probabilistic model that describes the data as being generated by a latent class variable in addition to a continuous latent representation.
- Adversarial Autoencoders (AAE). [315] employs the GAN to perform variational inference by matching the aggregated posterior of the hidden represen-

tation of the autoencoder.

- Ladder Variational Autoencoders (LVAE). [313] proposes an inference model which recursively corrects the generative distribution by a data dependent likelihood.

- Auxiliary Deep Generative Models (ADGM). [320] extends deep generative models with auxiliary variables, which improves the variational approximation.

We design ablation study to demonstrate the necessity of each key component of the proposed approach. In the ablation study, we set four control experiments with single variable among the components of AVAE. We adopt the following four methods to discover the latent representations: 1) VAE ($\boldsymbol{\mu}$) with $\boldsymbol{\mu}$ as the latent representation; 2) standard VAE ($\boldsymbol{z}_s$ as the latent representation); 3) VAE++ ($\boldsymbol{z}_I$ as the latent representation); 4) AVAE. The extracted representations are fed into the same classifier for classification.

The application-related state-of-the-art models on activity recognition are listed here:

- Chen et al. [323] adopt an attention mechanism to select the most distinguishable features from the activity signals and send them to a CNN structure for classification.

- Lara et al. [324] apply both statistical and structural detectors features to discriminate among activities.

- Guo et al. [325] exploit the diversity of base classifiers to construct a good ensemble for multimodal activity recognition, and the diversity measure is obtained from both labelled and unlabelled data.

- Zhang et al. [4] combine deep learning and the reinforcement learning scheme to focus on the crucial dimensions of the signals.

Results and Discussion

First, we report the overall performance of all the compared algorithms. From Table 7.1, we can observe that the proposed approach (AVAE) outperforms all the algorithm-related and application-related state-of-the-art models, illustrating the effectiveness of the latent space in providing robust representations for easier semi-supervised classification. The advantage is demonstrated under all the supervision rates.

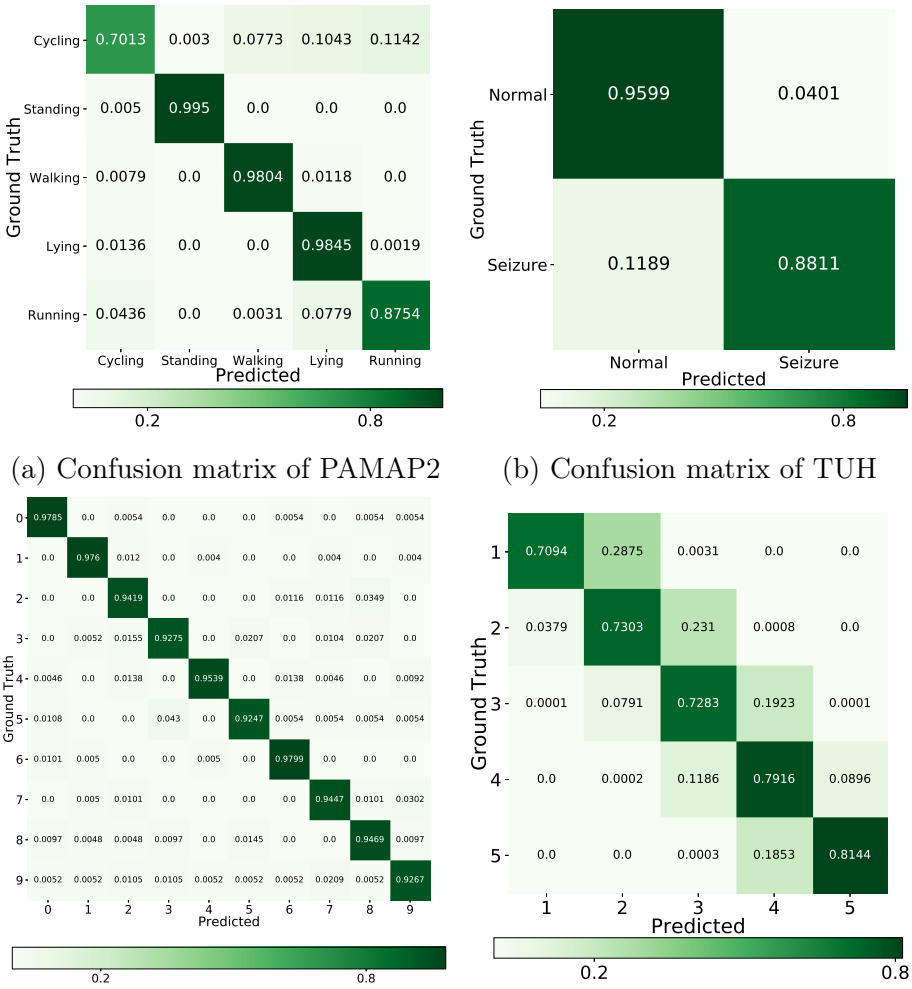
In Table 7.1, through the ablation study, it is observed that each component (especially GAN) contributes to the performance enhancement. Additionally, the proposed AVAE achieves a significant improvement which yields around 5% and 3% growth than the standard VAE and the VAE++ (under 60% supervision rate), respectively. This observation demonstrates that the proposed latent layer $\boldsymbol{z}_I$ and the adversarial training (between the discriminator and VAE++) encourages the proposed model to learn and refine the informative latent code. Take 60% supervision rate as an example, more details of the classification are shown in the confusion matrix (Figure 7.3a) and ROC curves with AUC score (Figure 7.4a).

7.3.2 Neurological Diagnosis

Experiment Setup

EEG signal collected in the unhealthy state differs significantly from the ones collected in the normal state [287]. The epileptic seizure is a common brain disorder that affects about 1% of the population and its octal state could be detected by the EEG analysis of the patient. In this application, we evaluate our framework with raw EEG data to diagnose the epileptic seizure of the patient.

We choose the benchmark dataset TUH [333] for epileptic seizure diagnosis. The TUH is a neurological seizure dataset of clinical EEG recordings associated with 22



(a) Confusion matrix of PAMAP2

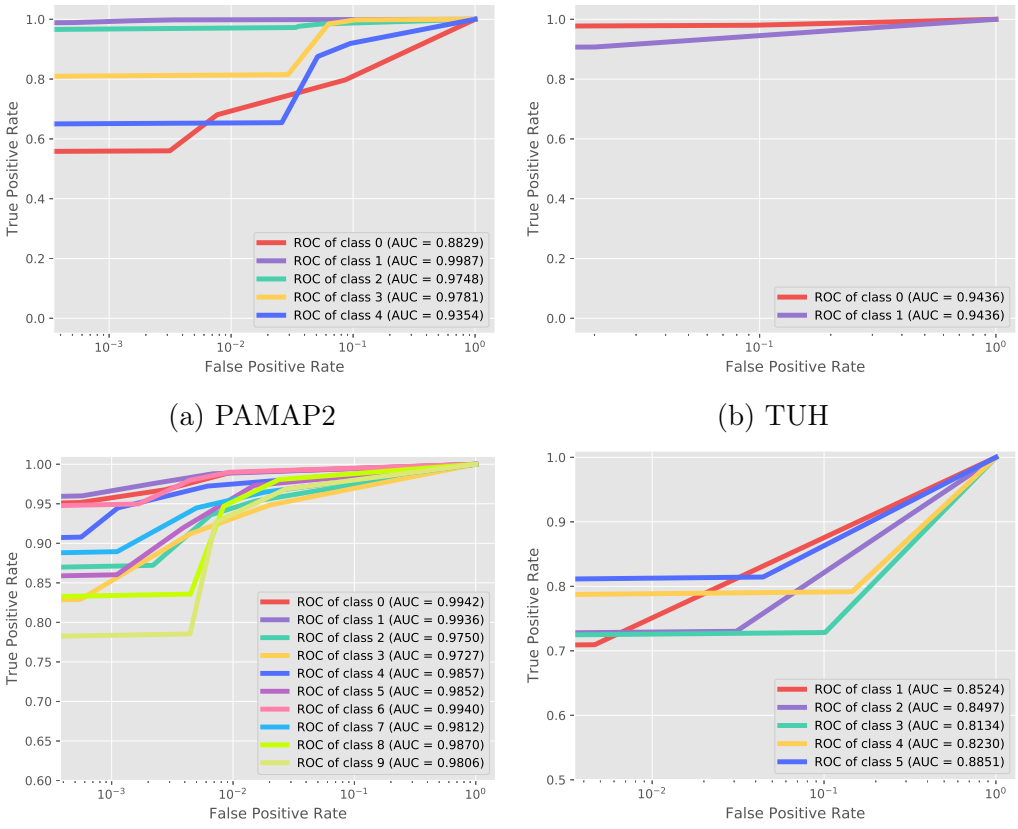
(b) Confusion matrix of TUH

(c) Confusion matrix of MNIST

(d) Confusion matrix of Yelp

Figure 7.3: Confusion matrix of PAMAP2, TUH, MNIST, and Yelp datasets.

channels from a 10/20 configuration. The sampling rate is set as 250 Hz. We select 12,000 samples from each of 18 subjects. Half of the samples are labelled as epileptic seizure state (labelled as 1) and the remaining samples are labelled as normal state (labelled as 0). The experiment and parameter settings are the same as the activity recognition applications.



(a) PAMAP2 (b) TUH

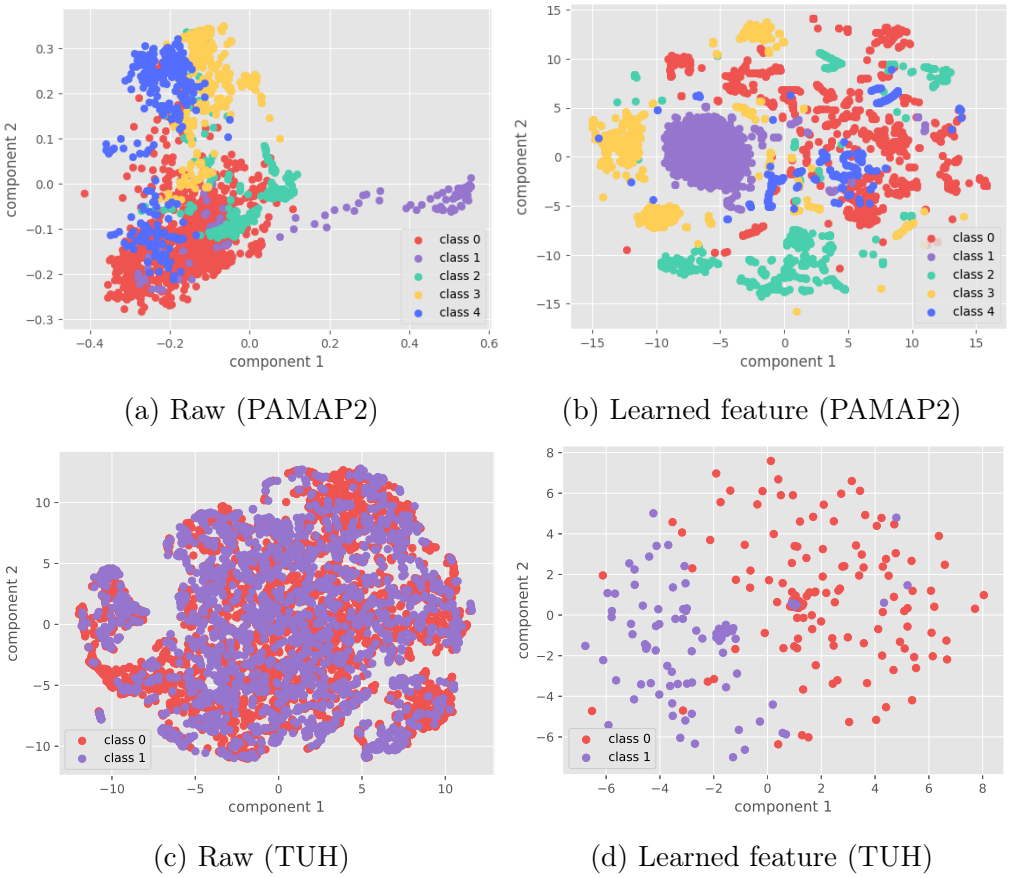
(c) MNIST (d) Yelp

Figure 7.4: ROC curves of PAMAP2, TUH, MNIST, and Yelp datasets. The X-axis is in logarithmic scale.

Baselines

The application-related state-of-the-art approaches in neurological diagnosis are listed here:

- Ziyabari et al. [303] adopt a hybrid deep learning architecture, including LSTM and stacked denoising Autoencoder, which integrates temporal and spatial context to detect the seizure.
- Harati et al. [304] demonstrate that a variant of the filter bank-based approach, coupled with first and second derivatives, provides a reduction in the overall



(a) Raw (PAMAP2)

(b) Learned feature (PAMAP2)

(c) Raw (TUH)

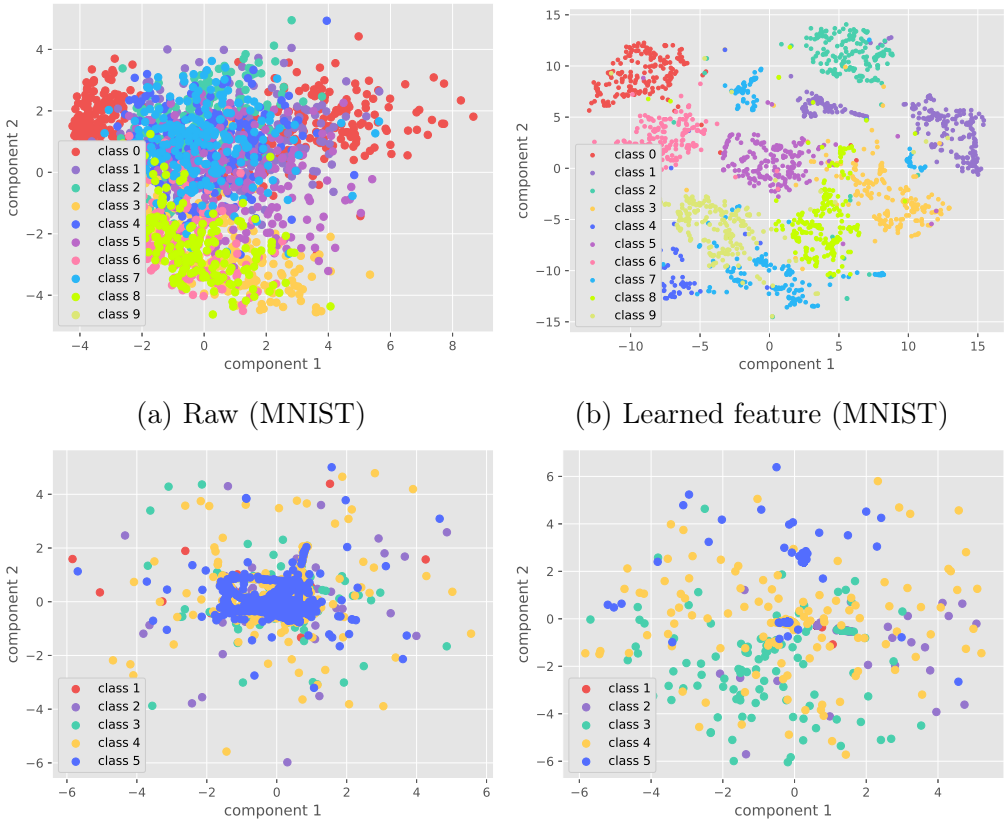
(d) Learned feature (TUH)

Figure 7.5: Visualization comparison between raw data and the semi-supervised learned features (PAMAP2 and TUH)

error rate.

- Schimeister et al. [121] attempt to improve the performance of seizure detection by combining deep ConvNets with training strategies such as exponential linear units.

- Goodwin et al. [306] combine RNN with access to textual data in EEG reports in order to automatically extracting word- and report-level features and infer underspecified information from EHRs (electronic health records).



(a) Raw (MNIST)

(b) Learned feature (MNIST)

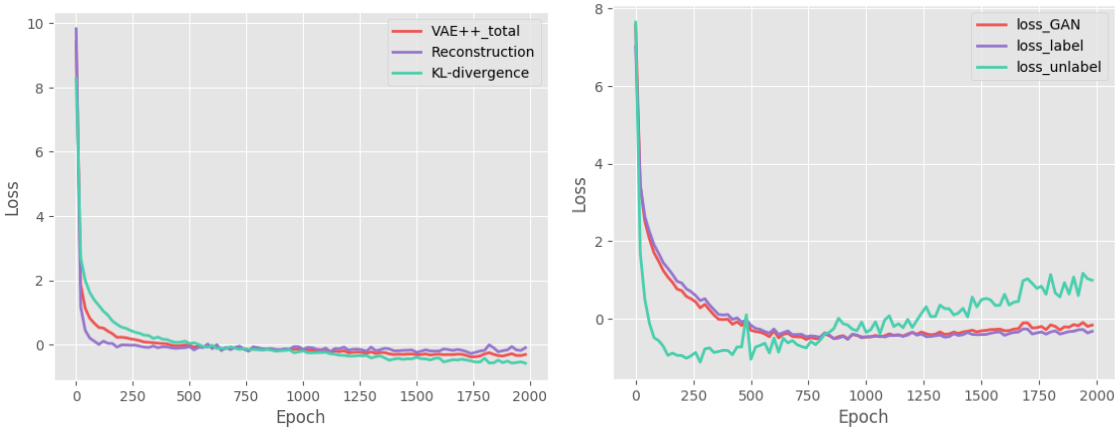
(c) Raw (Yelp)

(d) Learned feature (Yelp)

Figure 7.6: Visualization comparison between raw data and the semi-supervised learned features (MNIST and Yelp)

Results and Discussion

From Table 7.2, we can observe that our approach outperforms all the competitive baselines on TUH dataset. For instance, under 60% supervision level, the proposed approach achieves the highest accuracy of 95.21% which claims around 4% improvement over other methods. The corresponding confusion matrix (Figure 7.3b) and ROC curves (Figure 7.4b) infer that the normal state has higher accuracy than the seizure state. One possible reason is that the start and end stage of the seizure has similar symptoms with the normal state which may lead to misclassification.



(a) VAE++ (b) GAN

Figure 7.7: Convergence curve of the VAE++ and GAN

7.3.3 Image Classification

Experiment Setup

To evaluate the representation learning ability in images, we test our approach on the benchmark dataset MNIST [4]. MNIST contains 60,000 handwritten digital images (50,000 for training and 10,000 for testing) with $28 * 28$ pixels. The labels of this dataset are from 0 to 9, corresponding to the 10 digits.

Parameter Settings

Images are more informative compared to other application scenarios. The encoder of AVAE is designed to be stacked by two convolutional layers. The first convolutional layer has 32 filters with shape $[3, 3]$, the stride size $[1, 1]$, 'SAME' padding, and ReLU activation function. The followed pooling layer has $[2, 2]$ window size, $[2, 2]$ stride, and 'SAME' padding. The second convolutional layer has 64 filters with $[5, 5]$. The residual parameters of the second convolutional layer and pooling layer

[4]http://yann.lecun.com/exdb/mnist/

are the same with the former. Similarly, the decoder contains two de-convolutional layers with the same parameter settings.

Baselines

We reproduce the following methods under different supervision rate for comparison:

- Augustus [316] proposes a semi-supervised GAN (SGAN) by forcing the discriminator network to output class labels.
- Springenberg [321] proposes CatGAN to modify the objective function taking into account the mutual information between observation and the prediction distribution.
- Weston et al. [327] apply kernel methods for a nonlinear semi-supervised embedding algorithm.
- Miyato et al. [328] propose a regularization method based on virtual adversarial loss: a new measure of local smoothness of the conditional label distribution given the inputs.

7.3.4 Results and Discussion

As shown in Table 7.3, AVAE outperforms the counterparts with a slight gain with the same supervision level. The confusion matrix and ROC curves are reported in Figure 7.3c and Figure 7.4c. The results show that our approach is enabled to automatically learn the discriminative features by joint training the VAE++ and the semi-supervised GAN.

7.3.5 Recommender System

Experiment Setup

We apply our framework on recommender system scenarios, in particular, a restaurant rating prediction task based on the widely used Yelp dataset.

The Yelp Dataset[5] includes 192,609 Businesses, 1,637,138 Users, and 6,685,900 Ratings. Each business has 13 attributes (like 'near garage?', 'have valet?') which can describe the quality and convenience of the business. Meanwhile, each business is rated by a series customers. The ratings range from 1 to 5, which can reflect the customers' satisfactory degree. Our recommender task considers a unseen business's attributes as input data and predict the possible ratings from the potential customers. If the rating is high enough, the new business will be recommended to the public.

Baselines

We compare our approach with the state-of-the-art recommender system models which exploit the content information of items. Since these methods are used to make rating predictions for each user-item pair, we select those users who have 200 and more ratings in the Yelp dataset, generating a set of 1,111 users. After collecting the predicted ratings for all user-item pairs, we take the average item ratings over the users, which are further rounded to serve as the predicted labels.

- Pazzani et al. [329] summarizes basic content-based recommendation approaches, from which we select the cosine similarity-based nearest neighbor method as our fundamental baseline.

[5]https://www.yelp.com/dataset

- Rendle [330] proposes the original implementation of factorization machine(FM) which is capable of incorporating item features with explicit feedbacks. We concatenate only the item indication vector and its feature after each user indication vector following the format in [330].
- He et al. [331] enhances the original FM using deep neural networks to learn high-order interactions between different item features.
- Chen et al. [332] applies feature- and item-level attention on item features, which is capable of emphasizing on the most important features.

Results and Discussion

From Table 7.4, we can observe that our approach outperforms both the competitive semi-supervised algorithms and the content-based recommender system state-of-the-art methods. The rating prediction details can be found in Figure 7.3d and Figure 7.4d. The classification performance is not good as in other applications. One possible reason is that the attributes data are very sparse. The experiment results illustrate that our approach is effective in recommender system scenarios.

7.3.6 Further Analysis

Supervision Rate. We conduct extensive experiments to investigate the impact of supervision rate $\boldsymbol{\lambda}$. The supervision rate ranges from 20% to 100% with 20% interval and each setting runs for at least three times with the average accuracy recorded. From Table 7.2 to Table 7.4, it is noticed that the proposed model obtains competitive performance at each supervision level.

Visualization. Figure 7.5 visualizes the raw data and the learned features on different datasets. The visualization comparison demonstrates the capability of our approach for feature learning.

Convergence. Take PAMAP2 as an example, Figure 7.7 presents the relationship between the loss function values and epoch numbers. The VAE++ loss includes the reconstruction loss and the KL-divergence whilst the loss of the discriminator in GAN includes labelled loss and unlabelled loss (with weights 0.9 and 0.1, respectively). We can observe that the proposed method shows good convergence property as it stabilizes in around 200 epochs.

7.4 Discussion

In this section, we discuss several future scopes of the proposed AVAE.

First, our model requires adequate labelled training samples. Through from supervision rate analyze results, the proposed approach demands a sufficient supervised proportion. The lower supervision rate should be one major goal for the future.

Moreover, the proposed approach partially depends on the hyper-parameter settings. The hyper-parameter tuning (not presented in details due to space limitation), especially the key parameters like learning rate, is necessary for different data types. In our experiments, we choose one setting for activation recognition and the other setting for EEG signal recognition. For the parameter tuning, the Orthogonal Array experiment method is suggested [2]. The more generalized framework which is not sensitive to data types is one future scope.

Furthermore, the classification accuracy of the discriminator in GAN is not excellent even though the overall framework achieves a competitive performance. In the discriminator, the unlabelled classification (fake or real) achieves the accuracy of almost one hundred percent. However, for the labelled observations, the classification accuracy ($K + 1$ classes) is much lower. We believe this is a meaningful and crucial future direction in that an end-to-end semi-supervised classification framework could be built if the discriminator achieves satisfied performance.

7.5 Conclusion

In this chapter, we present an effective and robust semi-supervised latent representation framework, AVAE, by proposing a modified VAE model and integration with generative adversarial networks. The VAE++ and GAN share the same generator. In order to automatically learn the exclusive latent code, in the VAE++, we explore the latent code's posterior distribution and then stochastically generate a latent representation based on the posterior distribution. The discrepancy between the learned exclusive latent code and the generated latent representation is constrained by semi-supervised GAN. The latent code of AVAE is finally served as the learned feature for classification. The proposed approach is evaluated on four real-world applications and the results demonstrate the effectiveness and robustness of our model.

Chapter 8

Deep Neural Network Hyperparameter Optimization with Orthogonal Array Tuning

In this dissertation, we mainly focus on deep representation learning algorithms due to the excellent non-linear fitting ability. Nevertheless, deep learning faces an important challenge that the performance of the algorithm highly depends on the selection of hyper-parameters. Compared with traditional machine learning algorithms, deep learning requires hyper-parameter tuning more urgently because deep neural networks: 1) have more hyper-parameters to be tunned; 2) have higher dependency on the configuration of hyper-parameters. [2] reports the deep learning classification accuracy dramatically fluctuates from 32.2% to 92.6% due to the different selection of hyper-parameters. Therefore, an effective and efficient hyper-parameter tuning method is necessary.

However, most of the existing hyper-parameter tuning methods have some drawbacks. In particular, grid search traverses all the possible combinations of different

hyper-parameters, which is a time-consuming and ad-hoc process [334]. Random Search, which is developed based on grid research, set up a grid of hyper-parameter values and selects random combinations to train the algorithm [334]. Random search method oversteps some disadvantages of grid search such as time-consuming but meanwhile brings a major disadvantage which cannot converge to the global optimum [335]. The randomly selected hyper-parameter combinations cannot guarantee a steady and competitive result. Apart from the manually tuning methods, automated tuning methods being more popular in recent years [336]. Bayesian Optimization, a most widely-used automated hyper-parameter tunning approach, attempts to find the global optimum in a minimum number of steps. Nevertheless, the results of Bayesian optimization are sensitive to parameters of the surrogate model and the performance is highly depending on the quality of the learning model [337].

To address the aforementioned issue, we propose the Orthogonal Array Tuning Method (OATM) which can achieve a trade-off of the less tuning time and competitive performance. In detail, the OATM manner is proposed based on Taguchi Approach [232]. The OATM is a highly fractional orthogonal design method that is based on a design matrix and allows the user to consider a selected subset of combinations of multiple factors at multiple levels. Additionally, the OATM is balanced to ensure that all possible values of all hyper-parameters are considered equally. Moreover, OATM has been commonly used as an experimental design method in a wide variety of domains like mechanical engineering [338] and electrical engineering [339]. To our best knowledge, our work is *the first batch of work* adopting orthogonal array into parameter tuning in deep learning.

The proposed OATM adopts the orthogonal array to extract the most representative and balanced combinations from the whole set of possible combinations. The proposed OATM will be explained in detail in the context of two popular deep learning structures (Section 8.3). In addition, the OATM is evaluated over three datasets, which demonstrate the universality and adaptability. We notice that source codes performing grid search, random search, and especially Bayesian Optimization on

deep learning are hard to online acquire. Thus, we provide the reusable source codes and datasets for reproduction[1].

8.1 Orthogonal Array Tuning

In this section, we first provide the background knowledge of orthogonal array, namely, the definition, the compose principles, and the terminology. Then, we report the working procedure of OATM.

8.1.1 Orthogonal Array Tuning Method

In this section, we propose the Orthogonal Array Tuning Method inspired by the basic principles of orthogonal array. Although deep learning algorithms can achieve good performance in many research areas, tuning the hyper-parameters (e.g., the number of layers, the number of nodes in each layer and the learning rate) is time-consuming and dependent on user's expertise.

In OATM, the hyper-parameters are regarded as factors and different values of each hyper-parameter are regarded as levels. The procedure is listed as follows.

- **Step 1:** Build the F-L (factor-level) table. Determine the number of to-be-tuned factors and the number of levels for each factor. The levels should be determined by experience and literature. We further suppose each factor has the same number of levels[2].

[1]https://github.com/xiangzhang1015/OATM

[2]For the sake of simplicity, we consider all the factors with the same number of levels. More advanced knowledge can be found in [232] for more complex situations.

- **Step 2:** Construct Orthogonal Array Tuning table. The constructed table should obey the basic composition principles. Here[3] shows some commonly used tables. The Orthogonal Array Tuning table is marked as $L_M(h^k)$ which has k factors, h levels, and totally M rows.

- **Step 3:** Run the experiments with the hyper-parameters determined by the Orthogonal Array Tuning table.

- **Step 4:** Range analysis. This is the key step of OATM. Based on the experiment results in the previous step, range analysis method is employed to analyze the results and figure out the optimal levels and importance of each factor. The importance of a factor is defined by its influence on the results of the experiments. Note that range analysis optimizes each factor and combines the optimal levels together, which means that the optimized hyper-parameter combination is not restricted to the existing Orthogonal Array table.

- **Step 5:** Run the experiment with the optimal hyper-parameters setting.

8.2 Experimental Setting

To evaluate the proposed OATM, we design extensive experiments to tune the hyper-parameters of two most widely used deep learning structures, i.e., the RNN and CNN. Both of the two deep learning structures are employed on three real-world applications: 1) a human intention recognition task based on the Electroencephalography (EEG) signals [340]; 2) activity recognition based on wearable sensors like Inertial Measurement Unit (IMU); 3) activity recognition based on pervasive sensors like Radio Frequency IDentification (RFID).

[3] `https://www.york.ac.uk/depts/maths/tables/taguchi_table.htm`

8.2.1 Data Setting

The proposed OATM is evaluated over three different tasks on three benchmark datasets where each is divided into a training set (80%) and a testing set (20%).

EEG-based Intention Recognition. We select the widely used EEG dataset from PhysioNet eegmmidb database which contains 5 different categories (refers to Section 3.3.1).

IMU-based Activity Recognition. This dataset is collected by 9 participants [326], which contains 1200000 samples. 8 ADLs are selected as a subset of our paper. The activity is measured by 3 IMUs and each IMU collects sensor signal with 14 dimensions including two 3-axis accelerometers, one 3-axis gyroscopes, one 3-axis magnetometers, and one thermometer.

RFID-based Activity Recognition. We collect the signals from passive RFID tags [341] and have 3100 samples in total. 21 activities, including 18 ADLs (Activity of Daily Living) and 3 abnormal falls, are performed by 6 subjects. Each sample has 12 dimensions corresponding to 12 RFID tags. RSSI measures the power present in a received radio signal, which is a convenient environmental measurement in ubiquitous computing.

8.2.2 Deep Learning Structures

In this section, we briefly describe RNN and CNN structures and then introduce the key hyper-parameters that will be tuned in the experiments.

RNN Structure

The RNN structure used in this section is shown in Figure 8.1. In the hidden layer, to implement the recurrent function, two LSTM (Long Short-Term Memory) layer

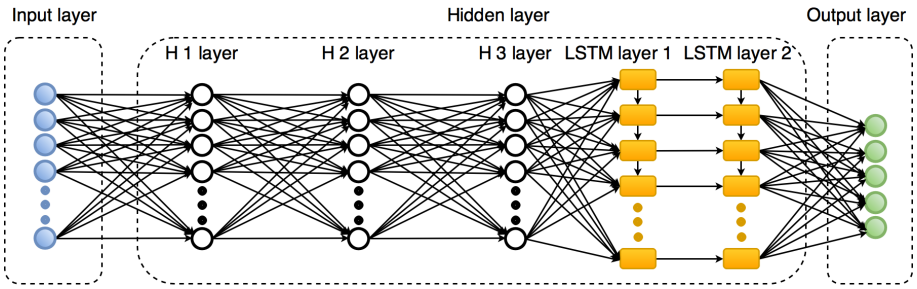


Figure 8.1: The schematic diagram of RNN structure. 'H' denotes Hidden, where, for example, the *H 1 layer* denotes the first hidden layer.

is concentrated. LSTM is a simple cell structure which can be used to build a recurrent neural network. Different from other fully connected layers, LSTM layer is composed of cells (shown as rectangles) instead of neural nodes (shown as circles).

In this RNN structure, based on the deep learning hyper-parameters tuning experience, the learning rate, the regularization, and the number of nodes in each hidden layer are key factors affecting the algorithm performance. The loss is calculated by cross-entropy function, and the regularization method is ℓ_2 norm with the coefficient λ, The loss is finally optimized by the AdamOptimizer algorithm. In summary, we choose four factors as to-be-tuned hyper-parameters: the learning rate lr, the regularization coefficient λ, the number of hidden layers n_l, and the number of nodes[4] in each hidden layer n_n.

CNN Structure

The CNN diagram is presented in Figure 8.2. The loss function, regularization method, and optimizer are the same as those in the RNN structure. Based on hyper-parameters tuning experience on CNN, we choose four most crucial factors to be tuned by OATM: the learning rate lr', the filter size f', the number of convolutional and pooling layers n_l'[5], and the number of nodes n_n' in the second fully connected

[4]Assume all the hidden layers have the same fixed number of nodes.

[5]We consider each convolutional layer and the following pooling layer as whole.

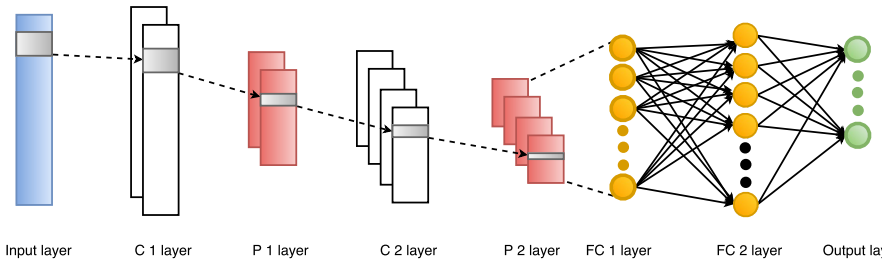


Figure 8.2: The schematic diagram of CNN structure. *C*, *P*, and *FC* denote convolutional layer, pooling layer, and fully connected layer, respectively.

layer.

8.3 Results and Analysis

8.3.1 Overall Comparison

In this section, we compare the proposed OATM with the most competitive state-of-the-art hyper-parameter tuning approaches including two manually methods (grid search and random search) and an automated one (Bayesian Optimization). It's easy to compute that there are $81 = 3^4$ exhausted combinations in grid search since we have four factors with three levels of the hyper-parameters. Thus, grid search requires 81 runnings to get the optimal hyper-parameters. On the other hand, our method requires only 9 runnings described in the corresponding orthogonal array table (see Section 8.3.2). Due to the numbers of runnings in random search and Bayesian Optimization are manually set, they are set as 9 runnings which is same with our method in order to keep fair comparison. The baselines are introduced here: 1) Grid search simply goes through all the possible combinations according to the values provided which is exhaustive [334]; 2) Random search randomly picks combinations from all possible ones. It may not find a decent combination but is widely adopted in industry for the high-efficiency [335]; 3)Bayesian optimization uses a Gaussian process to minimize the loss function in order to maximize performance

Table 8.1: Comparison with the state-of-the-art methods over three datasets and two deep learning architectures. The F1 $\sim$ F4 represent four tuning factors. Acc, Prec and F-1 denote accuracy, precision and F-1 score, respectively. #-R refers to the number of runnings.

Data	Models	Methods	Optimal Factors				Metrics					
			F1	F2	F3	F4	#-R	Time (s)	Acc	Prec	Recall	F-1
EEG	RNN	**Grid**	0.005	0.004	6	64	81	6853.6	0.9251	0.9324	0.9139	0.9231
		Random	0.01	0.008	6	32	9	766.8	0.7941	0.8003	0.7941	0.7947
		BO	0.0135	0.0049	5	32	9	703.4	0.718	0.7246	0.6474	0.6838
		Ours	0.005	0.004	6	64	9	821.9	0.925	0.9335	0.9223	0.9279
	CNN	**Grid**	0.005	4	3	192	81	31891.5	0.828	0.8137	0.8256	0.8269
		Random	0.003	2	1	128	9	662.8	0.7268	0.7277	0.7269	0.7266
		BO	0.001	4	3	139	9	721.9	0.7244	0.7302	0.7244	0.7263
		Ours	0.003	4	1	128	9	680.4	0.797	0.7969	0.8112	0.8003
IMU	RNN	**Grid**	0.005	0.004	6	96	81	3027.2	0.9936	0.9909	0.9976	0.9971
		Random	0.015	0.004	4	32	9	1008.5	0.9139	0.9209	0.9412	0.9156
		BO	0.0132	0.0041	4	48	9	1078.8	0.9872	0.9877	0.9851	0.9863
		Ours	0.005	0.004	6	64	9	1138.2	0.9913	0.9924	0.9905	0.9919
	CNN	**Grid**	0.003	2	1	128	81	41804.9	0.9732	0.9708	0.9708	0.9707
		Random	0.003	2	2	128	9	7089.2	0.9692	0.9691	0.9692	0.9691
		BO	0.0012	2	2	192	9	6559.7	0.9696	0.9702	0.9701	0.9701
		Ours	0.003	2	2	128	9	6809.8	0.9702	0.9699	0.9703	0.9702
RFID	RNN	**Grid**	0.005	0.008	6	96	81	2846.1	0.9342	0.9388	0.9201	0.9252
		Random	0.005	0.012	4	32	9	642.3	0.8891	0.9138	0.8826	0.8895
		BO	0.0142	0.0093	6	79	9	452.2	0.9071	0.8556	0.8486	0.8436
		Ours	0.005	0.008	6	64	9	497.1	0.9134	0.9138	0.9029	0.9162
	CNN	**Grid**	0.005	4	2	192	81	7890.8	0.9316	0.9513	0.9316	0.9375
		Random	0.005	2	1	128	9	1210.3	0.8683	0.9113	0.8684	0.8779
		BO	0.005	5	3	64	9	872.9	0.9168	0.9058	0.9194	0.9086
		Ours	0.005	4	3	192	9	980.3	0.9235	0.9316	0.9188	0.9326

[336].

The hyper-parameter levels are selected based on empirical values. For grid search, random search, and our OATM, the empirical values are discrete as listed in Table 8.2 (take eegmmidb as an example). For Bayesian Optimization, the hyper-parameter ranges from the maximum and minimum of each factor.

The comparison results are shown in Table 8.1. It can be observed that: 1)under the same running numbers (9 runnings), our method outperforms the random search and Bayesian Optimization over all the datasets and deep learning architectures; 2) our method performs slightly lower than grid search but still competitive, however, take EEG dataset with RNN as an example, our approach saves 88% tuning time which is indicated from that the OATM only requires 9 runnings and costs 821.9s while grid search requires 81 runnings and 6853.6s; 3) the optimal factors selected

Table 8.2: Factor-Level table of RNN and CNN.

		Factor 1 (lr)	**Factor 2** (λ)	**Factor 3** (n_l)	**Factor 4** (n_n)
RNN	**Level 1**	0.005	0.004	4	32
	Level 2	0.01	0.008	5	64
	Level 3	0.015	0.012	6	96
		Factor 1 (lr')	**Factor 2** (f')	**Factor 3** (n'_l)	**Factor 4** (n'_n)
CNN	**Level 1**	0.001	[1,2]	1	64
	Level 2	0.003	[1,4]	2	128
	Level 3	0.005	[1,6]	3	192

by our method approximate to the global optimal factors selected by grid search.

8.3.2 Case Study in RNN and CNN

In this section, we take EEG classification as an example to present the detailed procedure of OATM in RNN and CNN architectures. The overall paradigm can be divided into five steps.

Step 1: Build the F-L table According to the description in Section 8.2.2, OATM will work on four different hyper-parameters (factors): the learning rate lr, the l-2 norm coefficient λ, the number of hidden layers n_l, and the number of nodes n_n. The number of levels h is set to be 3 which could be much larger in real-world applications. Based on the related work and tuning experience [2], the empirical values are shown in Table 8.2.

Step 2: OATM table Choose a suitable Orthogonal Array table with 4 factors and 3 levels for our experiments in this link[6] wich contains 9 combinations. The OATM table should satisfy two basic principles: 1) in each column, different levels have the same appear times; 2) in any two randomly-selected columns, nine differently-ordered element combinations are completed and balanced.

[6] `https://www.york.ac.uk/depts/maths/tables/taguchi_table.htm`

Table 8.3: Range analysis of RNN

Row No.	**Factor 1 (lr)**	**Factor 2 (λ)**	**Factor 3 (n_l)**	**Factor 4 (n_n)**	**Acc**
1	0.005	0.004	4	32	0.875
2	0.005	0.008	5	64	0.8
3	0.005	0.012	6	96	0.521
4	0.01	0.004	5	96	0.888
5	0.01	0.008	6	32	0.797
6	0.01	0.012	4	64	0.451
7	0.015	0.004	6	64	**0.897**
8	0.015	0.008	4	96	0.335
9	0.015	0.012	5	32	0.471
R_{level1}	2.196	2.66	1.661	2.143	
R_{level2}	2.136	1.932	2.159	2.148	
R_{level3}	1.703	1.443	2.215	1.744	
A_{level1}	**0.732**	**0.887**	0.554	0.714	
A_{level2}	0.712	0.644	0.720	**0.716**	
A_{level3}	0.568	0.481	**0.738**	0.581	
Lowest Acc	0.568	0.481	0.554	0.581	
Highest Acc	0.732	0.887	0.738	0.716	
Range	0.164	0.406	0.184	0.135	
Importance		$lambda > n_l > lr > n_n$			
Best Level	**Level 1**	**Level 1**	**Level 3**	**Level 2**	
Optimal Value	**0.005**	**0.004**	**6**	**64**	**0.925**

Table 8.4: Range analysis of CNN

Row No.	**Factor 1 (lr')**	**Factor 2 (f')**	**Factor 3 (n_l')**	**Factor 4 (n_n')**	**Acc**
1	0.001	[1,2]	1	64	0.707
2	0.001	[1,4]	2	128	0.771
3	0.001	[1,6]	3	192	0.775
4	0.003	[1,2]	2	192	0.779
5	0.003	[1,4]	3	64	0.752
6	0.003	[1,6]	1	128	**0.797**
7	0.005	[1,2]	3	128	0.784
8	0.005	[1,4]	1	192	0.782
9	0.005	[1,6]	2	64	0.756
R_{level1}	2.253	2.27	2.993	2.215	
R_{level2}	2.328	2.305	2.306	2.352	
R_{level3}	2.322	2.328	2.311	2.336	
A_{level1}	0.751	0.757	**0.998**	0.738	
A_{level2}	**0.776**	0.768	0.769	**0.784**	
A_{level3}	0.774	**0.776**	0.770	0.779	
Lowest Acc	0.751	0.757	0.769	0.738	
Highest Acc	0.776	0.776	0.998	0.784	
Range	0.025	0.019	0.229	0.046	
Importance		$n_l' > n_n' > lr' > f'$			
Best Level	**Level 2**	**Level 3**	**Level 1**	**Level 2**	
Optimal Value	**0.003**	**[1,6]**	**1**	**128**	**0.797**

Step 3: Run the experiments Following the OATM table, run the 9 experiments and record the classification accuracy. In our case, each experiment runs 5 times with the corresponding average accuracy recorded. Each experiment is trained for 1,000 iterations to guarantee the convergence.

Step 4: Range analysis This is the key step of Orthogonal Array Tuning. The overall range analysis procedure and results are shown in Table 8.3. The first 9 rows are measured and recorded in Step 3. R_{leveli} denotes the sum of accuracy under level i. For example, R_{level1} in factor 1 is the sum of the accuracy in the first 3 rows ($2.196 = 0.875 + 0.8 + 0.521$), where factor 1 is on level 1. A_{leveli} denotes the average accuracy of level i, calculated by $A_{leveli} = R_{leveli}/h$. In the above example, we calculate A_{level1} as $0.732 = 2.196/3$. Lowest and highest accuracy values, measuring the maximum and minimum of A_{leveli} respectively, are used to calculate the *range* of A_{leveli}. The importance denotes how important the factor is, which is ranked by the range value. *Best level* is the selected optimal level based on the *Highest Acc* while *Optimal Value* represents the corresponding value of the best level.

Step 5: Run the optimal setting Run the experiment with the optimal hyperparameters ($lr = 0.004$, $\lambda = 0.005$, $n_l = 6$, and $n_n = 64$) and finally achieve the optimal accuracy as 0.925. It can be observed that: 1) the optimal accuracy 0.925 is higher than the maximum of the accuracy (0.897) in the OATM experiments, which demonstrates that the OATM is enabled to approximate the global optimal instead of the local optimal; 2) the importance of each factor is ranked through the range analysis: $lambda > n_l > lr > n_n$, which can guide the researcher to grasp the dominating variable in the RNN structure and be helpful in the future algorithm development.

The OATM paradigm of CNN is similar to RNN. Here, we only report the F-L table (Table 8.2) and the range analysis table (Table 8.4).

8.4 Conclusion

One disadvantage of OATM is that it requires the empirical values as prerequisites. The values of the F-L table should be chosen appropriately. However, this is the common drawback of all the tuning methods. For instance, the hyper-parameter ranges in Bayesian Optimization are also pre-defined based on empirical values.

In summary, we present an efficient OATM approach for hyper-parameter tuning in the context of deep learning. The proposed OATM is evaluated over two popular deep learning structures(RNN and CNN) over three real-world datasets. The experiment results show that our approach outperforms state-of-the-art hyper-parameter tuning methods.

Part V

BCI Applications

8. Hyperparameter Optimization with Orthogonal Array Tuning

This part contains works published in:

[8] **X. Zhang**, L. Yao, S. S. Kanhere, Y. Liu, T. Gu, and K. Chen. MindID: Person identification from brain waves through attention-based recurrent neural network. In *Proceedings of the ACM International Conference on Pervasive and Ubiquitous Computing (**UbiComp 2018**)*, 2(3), 149, 2018. (CORE Rank A*)

[9] **X. Zhang**, L. Yao, Q. Z. Sheng, S. S. Kanhere, T. Gu, and D. Zhang. Converting your thoughts to texts: Enabling brain typing via deep feature learning of EEG signals. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (**PerCom 2018**)* (pp. 1-10). IEEE 2018. (CORE Rank A*)

[10] **X. Zhang**, L. Yao, C. Huang, S. S. Kanhere, and D. Zhang. Brain2Object: Printing Your Mind from Brain Signals with Spatial Correlation Embedding. *Pattern Recognition*, 2018. (Under Review, SJR Q1)

[11] **X. Zhang**, X. Chen, M. Dong, H. Liu, C. Ge, and L. Yao. Multi-task Generative Adversarial Learning on Geometrical Shape Reconstruction from EEG Brain Signals. In *Proceedings of the 26th International Conference On Neural Information Processing (**ICONIP 2019**)*. Sydney, Australia, December 12-15, 2019. (CORE Rank A)

Chapter 9

MindID: EEG-based Person Identification System through Attention-based Recurrent Neural Network

One of the most promising BCI applications is the EEG-based user identification for the unique characteristics of EEG signals (such as fake-resistance).

Over the past decade, biometric information has been widely used in identification and have gained more acceptance due to their reliability and adaptability. Existing biometric identification systems are mainly based on individuals' unique intrinsic physiological features (e.g., face [342], iris [343], retina [344], voice [345], and fingerprint [346]). However, the state-of-the-art person identification systems have been shown to be vulnerable, e.g., anti-surveillance prosthetic masks can thwart face recognition, contact lenses can trick iris recognition, vocoder can compromise voice identification and fingerprint films can deceive fingerprint sensors. In this

perspective, the EEG based biometric person identification systems are emerging as promising alternatives due to their high attack-resilience [347, 348]. EEG-based identification systems measure an individual's brain response to a number of stimuli in the form of EEG signals, which record the electromagnetic and invisible electrical neural oscillations. An individual's EEG signals are virtually impossible to mimic for imposter, thus making this approach highly resilient to spoofing attacks encountered by other identification techniques [1].

EEG signals, compared with other biometrics, have the following significant inherent advantages [348]:

- *Attack-Resilience.* EEG data is invisible and untouchable and is impossible to be cloned and duplicated. Therefore, an EEG-based identification system is strengthened to verify human ID and robust against faked identities.
- *Universality.* One's EEG signals are typically associated with the subject all the time and hence security can be enforced anywhere and anytime.
- *Uniqueness.* Each individual processes his/her EEG signals which are unique, independent and different from other's [349]. This can potentially achieve high identification accuracy.
- *Accessibility.* We have seen an increasing effort in recent years in the development of low-cost and easy-to-wear EEG headsets. For example, the behind-the-ear EEG collection equipment[350] can be easily attached to the ear (similar to wireless earphone).

We put up a table showing the comparison of EEG with other biometric information on several key characteristics in Table 9.1.

[1]For example, people can easily trick a fingerprint-based identification system by using a fake fingerprint film (*http://www.instructables.com/id/How-To-Fool-a-Fingerprint-Security-System-As-Easy-/*) or a face-recognition-based identification system by simply wearing a 200 dollars' worth anti-surveillance mask (*http://www.urmesurveillance.com/urme-prosthetic/*)

However, research on EEG-based identification is still in its infancy, and several key challenges exist. One of the most significant issues is the low identification accuracy as a result of the inherent low precision of EEG signals. Accurate identification is challenging because the EEG data has very low signal-to-noise ratio. The state-of-the-art approaches can achieve accuracy in the range of 80% to 95% [351, 347, 352, 353], which is not sufficient for practical deployments, particularly in high security environments. Additionally, the identification algorithms are highly dependent on the environment in which the EEG signals were collected and thus not robust and adaptable to a broader range of scenarios. Changes in the application environment (e.g., the number of channels, the sampling rate, and the training data size) may lead to the decrease of accuracy[2]. Thus, an EEG-based identification model that may work well under one kind of application environment (e.g., 64 channels and 160 Hz), may not achieve good performance in another application environment (e.g., 14 channels and 128 Hz). So far, we have not seen a universal EEG-based identification algorithm which can perform well in a variety of real environments.

One of the most significant challenges is poor stability (the identification system may work well at one time but fail another time due to the EEG signals are easy to be interfered). This may due to the user's physiological and psychological states such as fatigue and angry [354, 355]. Intuitively, the states shift brought by the fluctuation of user states can be divided into two categories: the dramatically shift (e.g., hysterical, drunk, or under threaten) and the slight shift (e.g., headache or exciting). On one hand, the EEG signals divergence bought by the former can help to enhance the robustness of the identification system. For example, the phenomena could enhance the security of the identification system that it fails to recognize the subject who is under threatening (e.g., Hijacked by kidnapper). Knyazev [356] infers that EEG signals are affected by the inherent factors such as panic, sustained pain, and sexual arousal. On the other hand, however, the latter will reduce the signal quality but more commonly occurred in the real world. For instance, the identification system could identify the user when s/he is happy but fail when upset. Thus the slight shift

[2]This statement can be demonstrated in Section 9.4.7.

should be overcome for its negative effect.

To address the aforementioned problems, we propose MindID, a Delta pattern EEG-based person identification algorithm which is based on an attention-based recurrent neural network. At first, to eliminate the interference of the slight shift brought by the environmental noise and the physical and mental state of the individuals, we attempt to learn the robust and reliable representation by decomposing the EEG patterns. For this, we decompose the full spectrum of EEG data into specific patterns (Delta, Theta, Alpha, Beta, and Gamma). Decomposed EEG patterns (e.g., Theta, Alpha, Beta, and Gamma), have been employed for EEG signal classification (e.g., movement task classification [357]) in some works. However, there is few existing work that has focused on the Delta pattern. In this chapter, we discover that the Delta pattern is the most discriminative and efficient pattern through our analysis in Section 9.2. Moreover, we introduce the attention-based RNNs (Recurrent Neural Networks) [358] which can automatically detect the most distinguishable information from the input EEG data. More importantly, the attention mechanism[3] automatically re-allocates the weights to extract most discriminative features that are resilient to the change in environmental factors. Therefore, the proposed approach is robust under different collection environments with changes to the EEG collection hardware, sampling rate, and channel numbers. The efficiency of attention-based RNN framework has been demonstrated by the studies in speech recognition [358, 359], NLP (Natural Language Processing) [360, 361, 362], and computer version [363]. Our main contributions in this chapter are highlighted as follows:

- We present an EEG-based identification approach, MindID, which adopts a novel attention-based Encoder-Decoder RNN framework for learning discriminative features among the user's brainwaves and utilizes the learned features to identify user identity through a boosting classifier. The attention mechanism

[3]Simply, attention mechanism refers to select the most pertinent piece of information rather than using all available information. Attention Mechanisms in Neural Networks are based on the visual attention mechanism found in humans, and has been applied in computer version, NLP areas.

enables our approach to automatically search the most discriminative features for identification, and consequently achieve robust and adaptive operation over different datasets collected from environments with varying characteristics.

- We analyze the EEG pattern decomposition and propose that the Delta pattern is the most steady and distinguishable pattern for user identification. Moreover, we design and conduct a set of experiments to verify the proposed hypothesis.
- We design and conduct an experiment for collecting EEG data and use it to collect two two real-world local datasets (EID-M and EID-S) which are under single and multi trial settings, respectively[4].
- We evaluate the proposed approach on 3 datasets (2 local and 1 public). The results illustrate that our model achieves an accuracy of **0.982** which significantly outperforms the state-of-the-art and baselines. We demonstrate the robustness and adaptability by the comparison between 3 datasets.

Note that all the necessary reusable codes and datasets in this this chapter have been open-sourced for reproduction, please refer to this link [5].

9.1 Related Work

In this section, we separately present literature on three aspects: EEG-based person identification models, EEG pattern decomposition, and applications of attention-based RNN.

[4]Single trial refers to that the dataset is collected in one session (the period from one subject putting the EEG headset on until all the experiment are finished then putting off). Multi-trials represents the EEG data is collected from different trials, which considered the effect on EEG data quality caused by the headset position errors.

[5]`https://drive.google.com/open?id=1t6tL434ZOESb06ZvA4Bw1p9chzxzbRbj`

Table 9.1: Comparison of various biometrics. EEG have considerable rttack-resilient which is the most significant character of identification systems. ↑ denotes the higher the better while ↓ denotes the lower the better.

Biometrics	Attack-Resilient ↑	Univer--sality ↑	Unique--ness ↑	Stability ↑	Accessi--bility ↑	Perform--ance ↑	Cost ↓
Face/Vedio	Medium	Medium	Low	Low	High	Low	High
Finger/Palmprint	Low	High	High	High	Medium	High	Medium
Iris	Medium	High	High	High	Medium	High	High
Retina	High	Medium	High	Medium	Low	High	High
Signature	Low	High	Low	Low	High	Low	Medium
Voice	Low	Medium	Low	Low	Medium	Low	Low
face	Medium	High	Medium	Medium	Medium	Medium	High
Gait	High	Medium	High	Medium	Medium	High	Low
EEG	**High**	High	High	Low	Medium	High	Low

9.1.1 EEG-based Person Identification

Since EEG can be gathered in a safe and non-intrusive way, researchers have paid great attention to exploring this kind of brain signals. For person identification, EEG is promising for being confidential and attack-resilient but on the other hand, complex and hard to be analyzed [353]. Jayarathne et al. [364] decompose the EEG data and pay attentions on the Alpha and Beta wave. The Common Spatial Patterns (CSP) values were extracted as main features to train the LDA classifier which achieves accuracy of 96.97% for a 12 participants dataset. Thomas and Vinod [353] take advantage of individual alpha frequency and delta band signals to compose specific feature vectors. They also prefer PSD features but only perform the extraction merely on gamma band. However, all of the above approaches only work in one specific environment. Few studies attempt to build a universal EEG-based identification model.

9.1.2 EEG Pattern Decomposition

Generally, the EEG data could be decomposed into several patterns (delta, theta, alpha, beta, and gamma) corresponding to various brain states [365]. So far, the majority of user ID identification studies have focused on features generated from

the Alpha and Beta patterns.[348, 352]. Moreover, these works assume that the EEG data is collected from the most favorable settings, i.e., when the subject is resting/relaxed for Alpha waves or concentrating for Beta waves. The rest and relax states are represented by the Alpha wave, therefore, a number of studies decompose EEG raw signals into the Alpha pattern for future analysis. Bashar et al. [351] use the filtered signals with frequency ranges from $0.5 - 59Hz$ (including Delta, Theta, Alpha, Beta and part of Gamma patterns) and calculate the statistics for user ID classification. Kumari and Vaish [352] employ wavelet analysis to decompose original EEG signals into 5 patterns (Delta to Gamma) and extract statistical measures of each pattern. Thomas and Vinod [353] take Alpha peak frequency and peak power and Delta band power as recognition features and achieves the highest recognition rate as 0.9. To our best knowledge, this chapter is the very first work which specially focused on the decomposition and analysis of Delta pattern and studies the person identification based on it (the justification is given in Section 9.2).

9.1.3 Attention-based RNN

Attention-based RNN [363] introduces an attention mechanism to the RNN framework. The attention mechanism enables RNN to allocate different weights to different parts of the input, and consequently, improve the exploration of the corresponding relationship between the input sequence and the output sequence. Generally, attention module is added to the original RNN framework as an external module, but is trained instantaneously with the RNN structure [37]. Attention-based RNN has achieved success in speech recognition [358], NLP (Natural Language Processing) [362], and computer version [363]. Bahdanau et al. [358] attempt to build a Large Vocabulary Continuous Speech Recognition (LVCSR) Systems using attention-based RNN and demonstrate that their approach, compared with traditional methods, requires fewer training stages, less auxiliary data, and less domain expertise. Luong et al. [362] explore the architecture of attention-based neural machine translation and examine the effects of two attentional mechanisms (one that focuses on all

Table 9.2: EEG patterns and corresponding characters. Awareness Degree denotes the awareness the degree of being aware of an external world.

Patterns	Hz	Amplitude	Brain State	Awareness	Produced Location
Delta	0.5-4	Higher	Deep sleep pattern	Lower	Frontally and posteriorly
Theta	4-8	High	Light sleep pattern	Low	Entorhinal cortex, hippocampus
Alpha	8-12	Medium	Closing the eyes, relax state	Medium	Posterior regions of head
Beta	12-30	Low	Active thinking, focus, high alert, anxious	High	Most evident frontally
Gamma	30-100	Lower	During cross-modal sensory processing	Higher	Somatosensory cortex

source words and and the other which focuses on a subset of words) on the WMT translation tasks between English and German in both directions. Ba et al. [363] present an attention-based RNN for recognizing multiple objects in images, while only being provided with class labels during training. The results show that the attention-based RNN is more accurate and less computation than the state-of-the-art. To our best knowledge, we are the very first work employing attention-based RNN for EEG-based user identification.

9.2 EEG Pattern Analysis

In this section, we first introduce some background about EEG patterns followed by a topographical analysis of real-world EEG data to discover which specific constituent patterns capture the most distinctive features that allow us to distinguish the subject's identity. Next, we analyze why Delta pattern works best both qualitatively and quantitatively.

The EEG signals collected from any typical EEG hardware can can be divided into several non-overlapping frequency bands (Delta, Theta, Alpha, Beta, and Gamma) based on the strong intra-band correlation with a distinct behavioral state [366, 367, 365]. Each decomposed EEG pattern contains signals associated with particular brain information. The EEG frequency patterns and the corresponding characteristics are listed in Table 9.2. The awareness degree in this chapter denotes the perception of individuals while facing outside stimuli. Each frequency band represents a specific active situation of brain state and a qualitative assessment of

awareness. More specifically,

- **Delta pattern** ($0.5 - 4$ Hz) is associated with deep sleep while the subject has lower awareness.
- **Theta pattern** ($4 - 8$ Hz) corresponds to light sleep in the realm of low awareness.
- **Alpha pattern** ($8 - 12$ Hz) mainly occurs during eyes closed and deeply relaxed state, and corresponds to the medium awareness.
- **Beta pattern** ($12 - 30$ Hz) is the dominant rhythm while the subject's eyes are open and is associated with high awareness. Most of our daily activities (such as eating, walking, and talking) are captured by Beta patterns.
- **Gamma pattern** ($30 - 100$ Hz) represents the joint interaction of several brain areas to carry out a specific motor and cognitive function. This pattern is associated with highest awareness.

In order to investigate which EEG pattern is most intrinsic and rich of distinctive information for user identification, we study the EEG topography of different frequency patterns. Figure 9.1 shows the EEG topography of various subjects on full bands, Delta, Theta, Alpha, Beta, and Gamma patterns, respectively. Moreover, we calculated the cosine-similarity between EEG signals belonging to different subjects in a pairwise manner. The averaged cosine-similarity are as follows: 0.1313 (full patterns), 0.0722 (Delta pattern), 0.1672 (Theta pattern), 0.2819 (Alpha pattern), 0.0888 (Beta pattern), and 0.082 (Gamma pattern). This illustrates that the delta pattern has the lowest inter-subject similarity compared to other patterns and thus is likely to offer the most distinguishable features for person identification. In the following, we present two arguments to explain why Delta patterns are suited for user identification. Prior studies have shown that the functional significance of delta oscillations is not yet fully understood[356]. Our arguments below are based on the current knowledge of Delta patterns.

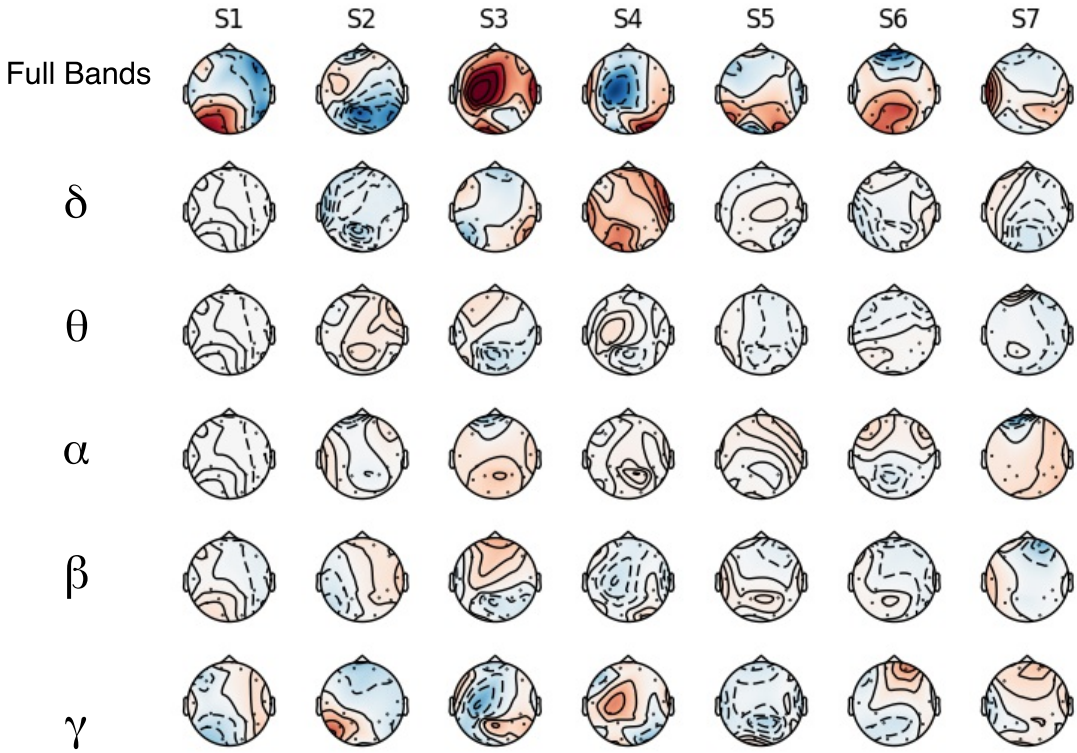


Figure 9.1: EEG topography of various subjects under different frequency patterns. The inter-subject EEG signal cosine-similarity is calculated under each pattern and the results are reported as 0.1313 (full patterns), 0.0722 (Delta pattern), 0.1672 (Theta pattern), 0.2819 (Alpha pattern), 0.0888 (Beta pattern), and 0.082 (Gamma pattern). This illustrates that the delta pattern has the lowest inter-subject similarity compared to other patterns and thus is likely to offer the most distinguishable features for person identification.

Table 9.3: The inter-subject correlation coefficients. S denotes subject. Full denotes the un-decomposed full-frequency band data. The lower coefficients indicate that the subject's EEG data is easier to be distinguished. We used data from the EID-M dataset (detailed in Section 9.4.1).

Subject		**S 1**	**S 2**	**S 3**	**S 4**	**S 5**	**S 6**	**S 7**	**S 8**	**STD**	**Average**
Patterns	Delta	0.137	0.428	0.246	0.179	0.221	0.119	0.187	0.239	0.089554	**0.219**
	Theta	0.447	0.671	0.552	0.31	0.387	0.207	0.199	0.386	0.151929	0.395
	Alpha	0.387	0.629	0.615	0.377	0.299	0.306	0.283	0.457	0.128653	0.419
	Beta	0.249	0.487	0.329	0.308	0.281	0.307	0.238	0.441	0.083224	0.33
	Gamma	0.528	0.692	0.538	0.362	0.521	0.667	0.428	0.537	0.102288	0.534
	Full	0.333	0.329	0.408	0.304	0.297	0.621	0.302	0.447	0.104231	0.38

On one hand, qualitatively, Delta pattern is universal and stable. A widely accepted view about Delta pattern is that it only occurs in deep sleep state. This is a significant reason why most researchers neglect Delta frequency in user identification. However, recent research in neurophysiology claims that the Delta rhythm is often evident during 'quiet' wakefulness in rodents and nonhuman primates [368]. This suggests that the delta patterns can dominate the background activity of some neocortical circuits in awake individuals. In addition, Delta pattern is observed to be related to cognitive processing [369]. It's easy to infer that Delta pattern exists while the subject is awake (processing cognitive tasks). Compared with baseline (a state with no delta waves), delta waves are associated with increase of activity in many brain regions, which suggests that Delta pattern is not associated with a state of brain quiescence, but rather associated with an active state during which brain activity is consistently synchronized to the slow oscillation in specific cerebral regions [370]. Moreover, there is evidence that suggests that Delta patterns are primarily created in the hypothalamus [371] which is associated with a series of life-support body functions such as autonomic regulation (e.g.,blood pressure, heart rate, thermoregulation) and neuroendocrine control [372]. Considerable evidence on the association between delta waves and autonomic and metabolic processes shows that integration of cerebral activity with homeostatic processes might be one of the Delta wave's functions [356]. Since the life-support functions are operational all the time, we can argue that regardless of the state of the individual, Delta oscillations will always be produced.

Next, we present some qualitative arguments to demonstrate that Delta patterns contain the most distinguishable information. We analyze inter-subject correlations of the decomposed EEG patterns, which measure the similarity of two samples belonging to different subjects. For example, the inter-subject correlation of subject 1 is calculated by the following steps: 1) randomly select 10 samples from subject 1; 2) randomly select 10 samples from each of other subjects (subject 2-8) to get 70 samples; 3) calculate the pair wise similarity between the first 10 samples and the latter 70 samples to get 700 similarities; 4) average the 700 similarities to produce the finally inter-subject correlation coefficient of subject 1. We measure the inter-subject correlations for all the frequency patterns in order to discover the most effective pattern. We used data from the EID-M dataset (detailed in Section 9.4.1). The correlation coefficient analysis results are shown in Table 9.3. We can observe that the Delta pattern has the lowest inter-subject correlation coefficients compared with other patterns. This indicates that the Delta patterns are most dissimilar to other samples, and thus most distinctive. Therefore, Delta patterns show promise for user identification. The dedicated comparative experiment between different EEG patterns will be reported in Section 9.4.7.

9.3 Methodology

In this section, we first give an overview of the proposed MindID system and then present the technical details for each component, namely, Preprocessing, EEG pattern decomposition, Attention-based RNN, and Classification.

9.3.1 Overview

Figure 9.2 outlines the specific steps of the proposed MindID system. The brainwaves are collected by the portable EEG acquisition equipment while the user is in a

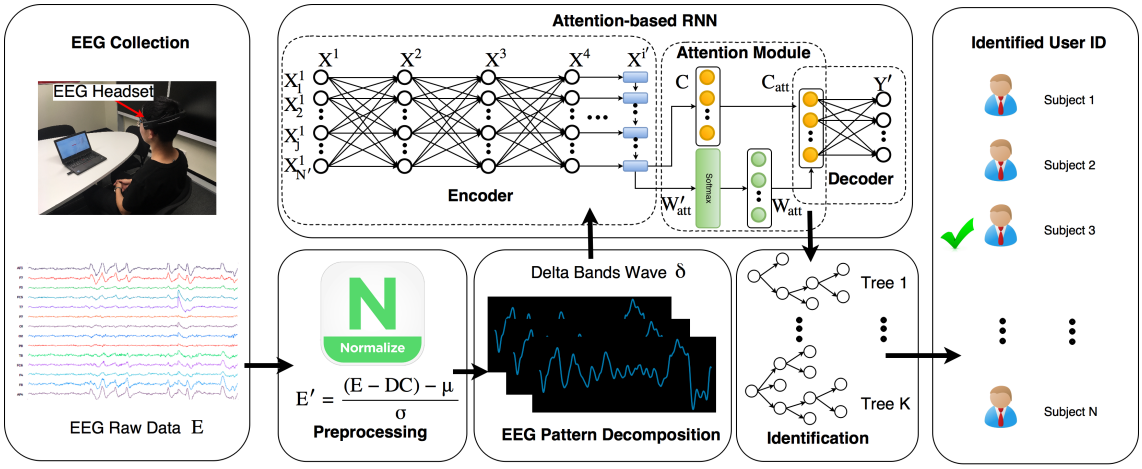


Figure 9.2: Flowchart of the proposed approach. At the beginning of identification, raw EEG data E is collected from the user and then fed to the preprocessing stage. The preprocessed data E' is decomposed to Delta pattern δ which serves as the input to the attention-based RNN. The encoder compresses the input sequence X^1 into an intermediate code C and produces the weights W'_{att} simultaneously. The attention-based module accepts both C and W'_{att} from the LSTM layer $X^{i'}$, processes W'_{att} through a softmax layer, and calculates the attention-based code C_{att}. Finally, a statistical boosting classifier is employed to identify the user.

relaxed state with his/her eyes closed (our preliminary experiment results illustrate that the Delta wave is more domination in relaxed state although still exists in all states). Each EEG sample is a numerical feature vector with N dimensions which correspond to the number of channels of the wearable EEG headset. The EEG samples are first preprocessed to remove the Direct Current (DC) offset and followed by normalization (Section 9.3.2). Next, we employ EEG pattern decomposition to isolate the Delta waves since they contain the most distinctive information which can be used to identify the subject (as outlined in Section 9.2). The delta waves are fed to an attention-based Encoder-Decoder RNN, which identifies the most distinctive channels and adjusts the weights accordingly. This model learns the deep correlations between the delta patterns which are then fed to a statistical boosting classifier (Section 9.3.5) to identify individual users.

9.3.2 Preprocessing

The raw EEG samples are pre-processed to remove the DC offset and normalize the signals. Eliminating DC offset is necessary because EEG headsets invariably introduce a constant noise component in the recorded signals. The specific headset used in our experiments (details in Section 9.4) introduces a DC offset of 4200 muV[6]. In the preprocessing stage, this constant DC offset is first subtracted from the raw signal E.

Normalization also plays a crucial role in a knowledge discovery process for handling different units and scales of features. For example, if two raw data sources, one ranging from 0 to 1 and another ranging from 0 to 100 are together used for analysis then the results will be dominated by the latter if normalization is not employed. Generally, there are three widely used normalization methods: Min-Max Normalization, Unity Normalization, and Z-score Scaling Normalization [3]. Our experiments (not shown for brevity) indicated that Z-score scaling is the most suited for the EEG data. In summary, the preprocessed data E' can be calculated by

$$E' = \frac{(E - DC) - \mu}{\sigma} \tag{9.1}$$

where DC denotes the Direct Current which is 4200 muV, μ denotes the mean of $E - DC$ and σ denotes the standard deviation.

9.3.3 EEG Pattern Decomposition

In Section 3, we used empirical EEG data to show that the part of the EEG signals that belong to the Delta frequency band $(0.5 - 4Hz)$ is particularly well-suited for accurate and robust user identification. To isolate the signals in the Delta band, we use a Butterworth band-pass filter of order 3 with the frequency range of 0.5Hz to 4Hz. The designed filter has the following specifications: the order is three, the low

[6] `https://www.bci2000.org/mediawiki/index.php/Contributions:Emotiv`

cut is $0.5Hz$, and the high cut is set as $4Hz$. The preprocessed signal E' is fed as input to this filter which provides the decomposed Delta pattern δ as output.

9.3.4 Attention-based RNN

Next, the Delta pattern δ is fed into an attention-based Encoder-Decoder RNN structure [37] which aims to learn the most representable features for user identification. The general Encoder-Decoder RNN framework assumes that all feature dimensions of the input sequence are equally important and assigns them equal weights. In the context of EEG data, each dimension refers to a different electrode of the EEG equipment. For example, the first dimension (first channel) collects the EEG data from the $AF3$[7] electrode which is located at the frontal lobe of the scalp while the 7-th dimension is gathered from $O1$ electrode at the occipital lobe.

Since different EEG channels record different aspects of the brain signals, some of which are more representative of the individual, an approach that assumes all dimensions to be equal may not be suitable. On the other words, various EEG channels have different contribution to the person identification task and should be corresponding to different weights. The effectiveness of attention-based RNN has been demonstrated in various domains including wearable sensor based activity recognition [323, 373], natural language processing [360, 361, 362], computer version [363] and speech recognition [358, 359]. Inspired by the wide success of this approach, we introduce the attention mechanism to the Encoder-Decoder RNN model to assign varying weights to different dimensions of the EEG data. The proposed attention-based Encoder-Decoder RNN consists of three components (as shown in Figure 9.2): the encoder, the attention module, and the decoder. The encoder is designed to compress the input Delta δ wave into a single intermediate code C; the attention module calculates a better intermediate code C_{att} by generating a se-

[7]Both $AF3$ and $O2$ are EEG measurement positions in the International 10-20 Systems.

quence of distinct weights W_{att} for the different dimensions; the decoder accepts the attention-based code C_{att} and decodes it to the user ID. Note, this user ID is predicted by the attention-based RNN instead of MindID, and the final identified ID of MindID approach will be introduced in Section 9.3.5.

Suppose the data in i-th layer could be denoted by $X^i = (X^i_j; i \in [1, 2, \cdots, I], j \in [1, 2, \cdots, N^i])$ where j denotes the j-th dimension of X^i. I represents the number of neural network layers in the proposed attention based RNN model while N^i denotes the number of dimensions in X^i. Take the first layer as an example, we have $X^1 = \delta$ which indicates the input sequence is the Delta pattern. Let the output sequence be $Y = (Y_k; k \in [1, 2, \cdots, K])$ where K denotes the number of user ID categories. In this chapter, the user ID is represented by the one-hot label with length K. For simplicity, let's define the operation $\mathcal{T}(\cdot)$ as:

$$\mathcal{T}(X^i) = X^i W + b \tag{9.2}$$

Further more, we have

$$\mathcal{T}(X^{i-1}_j, X^i_{j-1}) = X^{i-1}_j * W' + X^i_{j-1} * W'' + b' \tag{9.3}$$

where W, b, W', W'', b' denote the corresponding weights and biases parameters.

The encoder component contains several non-recurrent fully-connected neural network layers and one recurrent Long Short-Term Memory (LSTM) layer. The non-recurrent layers are employed to construct and fit a non-linear function to purify the input Delta pattern. The necessity of which is demonstrated by our preliminary experiments[8]. The data flow in these non-recurrent layers are calculated as follows,

$$X^{i+1} = \mathcal{T}(X^i) \tag{9.4}$$

The LSTM layer is adopted to compress the output of non-recurrent layers to a length-fixed sequence which is regarded as the intermediate code C. Suppose LSTM

[8]Some optimal designs like the neural network layers are validated by the preliminary experiments but the validation procedure will not be reported in this this chapter for space limitation

is the i'-th layer, the code equals to the output of LSTM, which is $C = X_j^{i'}$. The $X_j^{i'}$ can be measured by

$$X_j^{i'} = \mathcal{L}(c_{j-1}^{i'}, X_j^{i-1}, X_{j-1}^{i'}) \tag{9.5}$$

where $c_{j-1}^{i'}$ denotes the hidden state of the $(j-1)$-th LSTM cell. The operation $\mathcal{L}(\cdot)$ denotes the calculation process of the LSTM structure, which can be inferred from the Equation 3.1 to Equation 3.6.

The attention module accepts the final hidden states as the unnormalized attention weights W'_{att} which can be measured by the mapping operation $\mathcal{L}'(\cdot)$ (similar with Equation 9.5)

$$W'_{att} = \mathcal{L}'(c_{j-1}^{i'}, X_j^{i-1}, X_{j-1}^{i'}) \tag{9.6}$$

and calculate the normalized attention weights W_{att}

$$W_{att} = softmax(W'_{att}) \tag{9.7}$$

The softmax function is employed to normalize the attention weights into the range of $[0, 1]$. Therefore, the weights can be explained as the probability that how the code C is relevant to the output results. Under the attention mechanism, the code C is weighted to C_{att}

$$C_{att} = C \odot W_{att} \tag{9.8}$$

Note, C and W_{att} are trained instantaneously. The decoder receives the attention-based code C_{att} and decodes it to predict the user's identity Y'[9]. Since Y' is predicted at the output layer of the attention based RNN model ($Y' = X^I$), we have

$$Y' = \mathcal{T}(C_{att}) \tag{9.9}$$

At last, we employ the cross-entropy function to calculate the prediction cost between the predicted ID Y' and the ground truth Y. ℓ_2-norm (with parameter λ) is selected to prevent overfitting. The cost is optimized by the AdamOptimizer algorithm [231]. The threshold for the number of iterations of the attention-based RNN

[9]Note, Y' is not the identification results of MindID model. The final identified user ID is I_D calculated in Equation 9.11

is set as n_{iter}. The weighted code C_{att} has a linear relationship with the output layer and the predicted results. If the model is trained well then the weighted code C_{att} could be regarded as the weighted code as a high-quality representation of the identity of the user. We set the learned deep feature X_D equals to C_{att}, $X_D = C_{att}$, and use it to recognize the user in the identification stage.

9.3.5 Identification

In this section, we employ Extreme Gradient Boosting classifier (XGB) [260] to classify the learned deep feature X_D for user identification. The XGB classifier fuses a set of classification and regression trees (CART) and exploits as detailed information as possible from the input features X_D. It builds multiple trees and each tree has its leaves and corresponding scores. Moreover, it proposes a regularized model formalization to prevent over-fitting and it is widely used for its accurate prediction power.

The learned deep feature X_D is used to train a number of the CART (there are M trees) and predict a set of user's IDs. Suppose $x_d \in X_D$ is a single sample of the deep feature. The finally identification result of the input x_d is calculated as

$$y_m = f(x_d) \tag{9.10}$$

$$I_D = F(\sum_{1}^{M} y_m), m = 1, 2, \cdots, M \tag{9.11}$$

where f denotes the classification function of a single tree, y_m denotes the predicted ID of the m-th tree and F denotes the mapping from single tree prediction space to the final prediction space. The I_D is the final identified user ID. The overall procedure is summarized in Algorithm 3.

ALGORITHM 3: The MindID User Identification Algorithm

Input: EEG raw data E

Output: Identification results I_D

Initialization;
Preprocessing: $E' \leftarrow E$;
EEG pattern decomposition: $\delta \leftarrow E'$;
if $iteration < n_{iter}$ **then**
  for $i = 1, 2, \cdots, I$ **do**
    $X^1 = \delta$
    $C \leftarrow X^1, \mathcal{L}(c_{j-1}^{i'}, X_j^{i-1}, X_{j-1}^{i'})$
    $W_{att} \leftarrow C, \mathcal{L}'(c_{j-1}^{i'}, X_j^{i-1}, X_{j-1}^{i'})$
    $C_{att} = C \odot W_{att}$
    $X_D = C_{att}$
  end for
else
  Return X_D
end if
for X_D **do**
  $I_D \leftarrow X_D$
end for
return I_D

9.4 Experiments and results

We first outline the experimental settings in Section 9.4.1. Next, we systematically investigate the following questions: How does MindID compare with state-of-the-art methods and other baselines (Section 9.4.2)? How efficient is MindID (Section 9.4.3)? Is MindID robust under a multi-trial setting (Section 9.4.4)? Does MindID exhibit consistence results when tested with different datasets (Section 9.4.5)?

Table 9.4: Datasets details. In Trial column, M denotes multi-trials and S denotes single-trial. EID-M is used to compare the proposed approach with the state-of-the-art and baselines; the comparison between EID-M and EID-S are used to verify the robustness; the comparison between EID-S and EEG-S are used to verify the adaptability. EED-S-L is used to evaluate the influence of the number of participants and EEG-S is a subset of EEG-S-L.

Name	Source	Channels	Trial	Frequency	Subjects	Comparison	Robustness	Adaptability
EID-M	Local	14	M	128 Hz	8	✓	✓	-
EID-S	Local	14	S	128 Hz	8	-	✓	✓
EEG-S-L	Public	64	S	160 Hz	20	-	-	-
EEG-S	Public	64	S	160 Hz	8	-	-	✓

Do the number of subjects impact the results (Section 9.4.6)? How do other decomposed EEG signals compare with the Delta signals (Section 9.4.7)?

9.4.1 Experimental Settings

Datasets

The proposed MindID system is evaluated by three datasets: a multi-trial local dataset (EID-M), a single-trial dataset (EID-S), and a public dataset (eegmmidb). The details of datasets are introduced in Table 9.4. All the datasets measure the EEG raw data from the subject's scalp while the subject is relaxed.

EID-M denotes EEG based ID recognition with the training set coming from the different trials in the same day. Since a multi-trial scenario is more representative of a practical setting, EID-M dataset is used to compare MindID with the state-of-the-art methods and baselines. The EID-M dataset is collected locally in our lab from 8 subjects (5 males and 3 females) aged from 24 to 28. We use the *Emotiv Epoc+*[10] headset. The experiment setting refers to Figure 4.8 in Chapter 4.

The Emotiv Epoc+ contains 14 channels and the sampling rate is set as 128 Hz. In the experiment, each subject undertakes three trials and each trial produces 7,000

[10] `https://www.emotiv.com/product/emotiv-epoc-14-channel-mobile-eeg/`

Table 9.5: Evaluation report of EID-M dataset. The overall accuracy achieves 0.982 of 21000 testing samples. The support is the number of samples of each class.

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**Average/Total**
Precision	0.9723	0.9789	0.9777	0.9894	0.989	0.9814	0.9898	0.9774	**0.982**
Recall	0.9822	0.9885	0.9945	0.9711	0.9808	0.9821	0.9742	0.9834	0.9821
F1-score	0.9772	0.9837	0.9860	0.9802	0.9849	0.9818	0.9820	0.9804	0.982

EEG samples. Summarily, each subject has 21,000 samples and the whole EID-M dataset contains 168,000 samples.

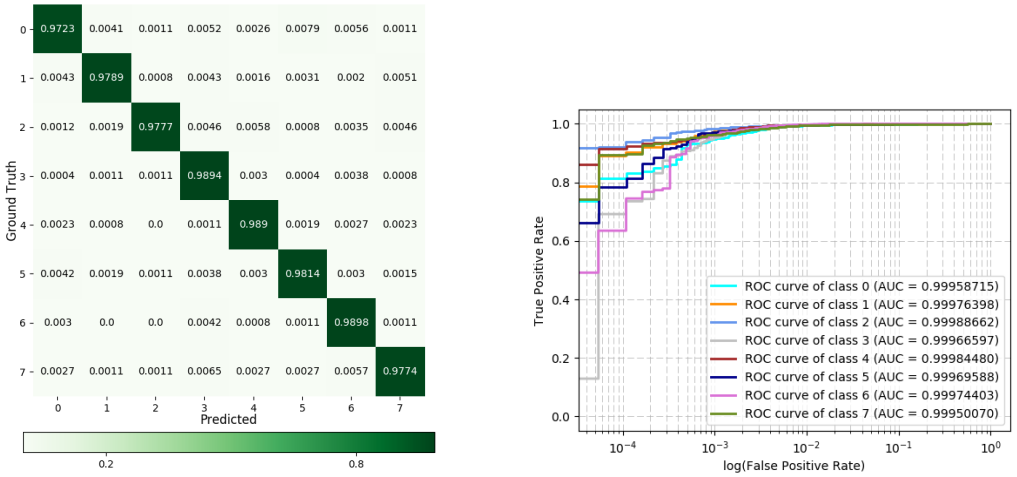
EID-S is collected under the same situation with EID-M (5 males, 3 females, 14 channels, and 128 Hz). The main difference between them is the former dataset is collected in the single trial. EID-S in total contains 56,000 samples belonging to 8 subjects (7,000 samples per subject).

EEG-S-L is a subset of the widely used online public dataset *eegmmidb* (*EEG motor movement/imagery database*). It is collected with the BCI2000 (Brain Computer Interface) instrumentation system [11] [233] (64 channels and 160 Hz sampling rate). EEG-S contains 20 subjects with 7000 samples collected from each subject in a single trial setting.

EEG-S is a subset of EEG-S-L, which only contains 8 subjects. To compare the adaptability of the proposed approach (Section 9.4.5), we randomly select 8 participants from EEG-S-L to compare with EID-S. This allows us to undertake a like-by-like comparison with the only variable being the type of EEG headset used.

To assess the performance of the proposed MindID model, we employ several widely-used evaluation metrics such as accuracy, precision, recall, F1 score, ROC (Receiver Operating Characteristic) curve, support, and AUC (Area Under the Curve).

[11] `http://www.schalklab.org/research/bci2000`

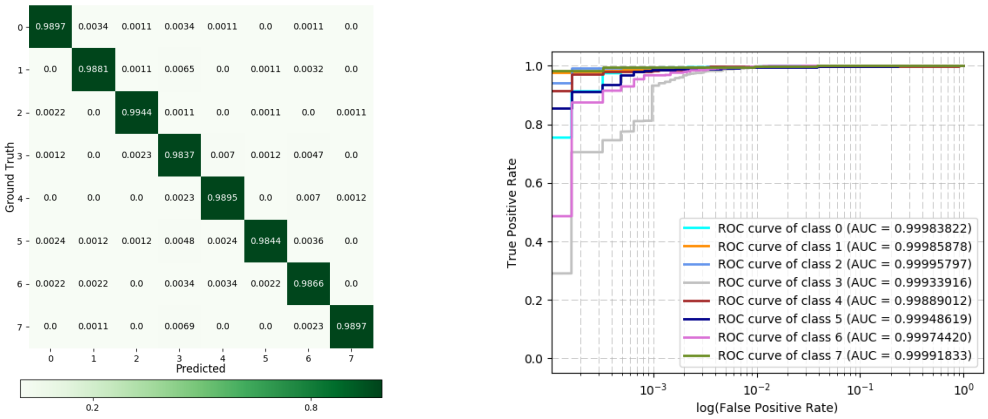


(a) Confusion matrix of EID-M (b) ROC and AUC of EID-M

Figure 9.3: Confusion matrix and ROC curves of EID-M

9.4.2 Overall Comparison

In this section, we firstly report the performance of MindID using the EID-M dataset and then compare the proposed approach with the state-of-the-art approaches and baselines. We randomly select 147,000 samples from EID-M to train the model and the residual 21,000 samples are used to test the performance. Through tuning, the hyper-parameters used in our approach are listed following. In EEG pattern decomposition, we employ a 3 order butter-worth band-pass filter and the passband is $[0.5Hz, 4Hz]$. In the attention-based RNN structure, the encoder consists of 1 input layer (14 nodes), 3 non-recurrent fully-connected hidden layers (164 nodes) and 1 recurrent LSTM layer (164 cells); the decoder includes 1 fully-connected hidden layer (164 nodes) and 1 output layer (8 nodes). The learning rate is 0.001; the parameter of $\ell - 2$ norm is set as 0.001; the encoder and decoder separately have 6 and 2 layers; training dataset is divided into 7 batches with the batch size of 21,000; the number of training iterations is 2000. In the classifier: the learning rate is 0.7; the sub-sampling rate is 0.9; the max depth is set as 6; the training iterations is 500. The ground truth (from 0 to 7) is represented as a one-hot label which corresponding to the ID of subjects.



(a) Confusion matrix of EID-S

(b) ROC and AUC of EID-S

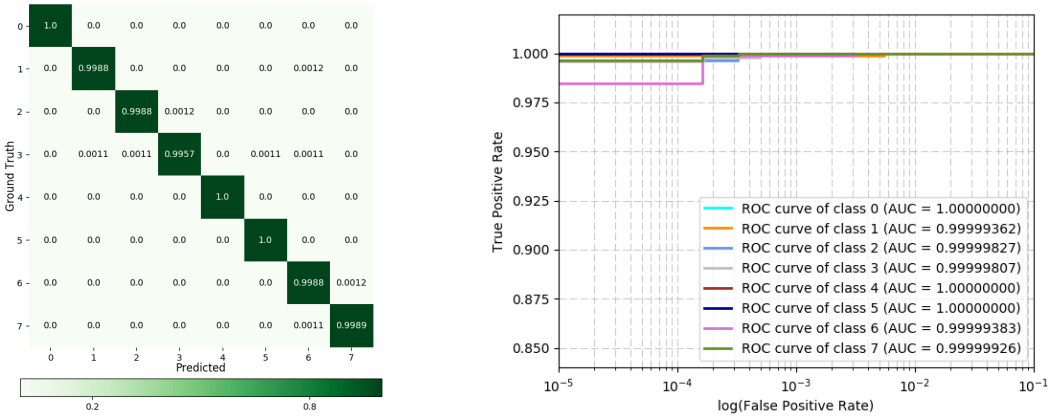
Figure 9.4: Confusion matrix and ROC curves of EID-S

The proposed approach achieves the highest identification accuracy of **0.982**. The detailed confusion matrix, evaluation report, and ROC curves (with AUC scores) are illustrated in Figure 9.3a, Table 9.5, and Figure 9.3b, respectively. Observe that our approach obtains higher than 0.97 precision for each class.

In addition, we compare the accuracy of our method and other state-of-the-art and baselines in Table 9.6. RF denotes Random Forest, AdaBoost denotes Adaptive Boosting, LDA denotes Linear Discriminant Analysis, PD denotes for Pattern Decomposition, AR denotes AutoRegressive method, and XGB denotes for X-Gradient Boosting classifier (the classifier used in our approach). In addition, the key parameters of the baselines are listed here: Linear SVM ($C = 1$), RF ($n = 200$), KNN (k=3), and AR (13 order autoregressive from 40 samples). The setting up of PD, RNN and XGB classifier are same as the hyper-parameters mentioned above. The methods used in the state-of-the-art are introduced as follows:

- Jayarathne et al. [364] focus on the 8 to 30 Hz Alpha and Beta combined frequency band across all EEG channels and extract the Common Spatial Patterns (CSP) values as classification features. LDA is employed as the classifier.

- Bashar et al. [351] first remove noise and artifacts using Bandpass FIR filter.



(a) Confusion matrix of EEG-S (b) ROC and AUC of EEG-S

Figure 9.5: Confusion matrix and ROC curves of EEG-S

Then learn the features through multi-scale shape description (MSD), multi-scale wavelet packet statistics (WPS) and multi-scale wavelet packet energy statistics (WPES). These features are finally used to train a support vector machine (SVM) classifier.

- Keshishzadeh et al. [374] investigates the Autoregressive (AR) coefficients as the feature set which is identified by an SVM classifier.

- Gui et al.[349] propose to reduce the noise level through a low-pass filter, extract frequency features using wavelet packet decomposition, and perform classification based on a deep neural network.

- Thomas and Vinod [353] combine subject-specific Alpha peak frequency, peak power, and Delta band power values to form discriminative feature vectors and templates.

- Kumari and Vaish [352] apply discrete wavelet analysis to decompose the raw EEG signals corresponding to sub-band frequency (0-59Hz). The extracted statistical measures and energy calculation of each decomposed wave are classified by a neural network structure.

Table 9.6: The accuracy comparison with baselines and the state-of-the-art methods over EID-M dataset. The result shows that our approach achieves the highest accuracy of 0.982.

Index	Method	Acc	Recall	F1-Sore	AUC
1	Jayarathne[364]	0.919	0.914	0.9165	0.946
2	Bashar et al. [351]	0.873	0.898	0.8853	0.907
3	Keshishzadeh et al. [374]	0.815	0.843	0.8288	0.859
4	Gui et al.[349]	0.833	0.811	0.8219	0.842
5	Thomas and Vinod [353]	0.859	0.869	0.8640	0.888
6	Kumari and Vaish [352]	0.875	0.872	0.8735	0.901
7	RF	0.795	0.813	0.8039	0.827
8	KNN	0.849	0.836	0.8424	0.847
9	RNN	0.815	0.803	0.8090	0.821
10	RNN+XGB	0.808	0.789	0.7984	0.803
11	PD+RNN	0.853	0.821	0.8367	0.844
12	AR+RNN	0.811	0.798	0.8044	0.831
13	XGB	0.815	0.811	0.8130	0.853
14	PD+XGB	0.965	0.959	0.9620	0.977
15	**Ours (EID-M)**	**0.982**	**0.9821**	**0.9820**	**0.999**

As noted earlier, we use the EID-M dataset for the comparison. As observed from Table 9.6, our method significantly outperforms all other methods in all metrics.

9.4.3 Efficiency Evaluation

In this section, the efficiency refers to the latency incurred to perform the identification.High latency may limit the suitability for practical deployment. We compare MindID with the same baselines and classification methods as in Section 9.4.2. We run the experiments on a GPU-accelerated machine with Nvidia Titan X Pascal GPU, 768G memory, and 145 TB PCIe based SSD.

The time required to train the identification model is illustrated in Figure 9.6(the X-axis label denotes the index of algorithms shown in Table 9.6). Observe that our approach (PD+RNN+XGB) and RNN+XGB require longer to train the model that other methods. There are two reasons behind this. First, these algorithms iterate

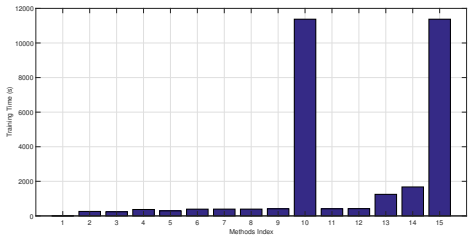


Figure 9.6: Training time. The index corresponding the index in Table 9.6.

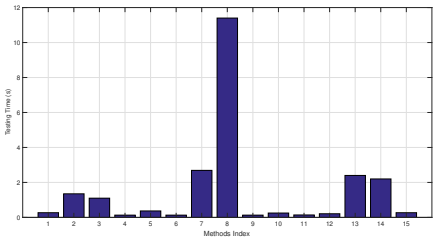


Figure 9.7: Testing time. The index corresponding the index in Table 9.6.

over a large number of rounds. RNN and XGB executes 2000 and 500 iterations, respectively. Second, the deep learning structure and the boosting trees have an inherent complex structure and require many more parameters than other classification models. Compared to the training time, however, for practical considerations, the execution time of an algorithm during testing is more important than training which is a one-time operation. Figure 9.7 shows that the testing time of our model is less than 1 second, which is shorter than most of the state-of-the-art methods and baselines. Summarily, while MindID requires longer to train, the actual execution is near real-time ($<$ 1 sec), thus making it attractive for real-world deployment.

In practice, the amount of data needed to train the model is also an important consideration as gathering training data is not so easy. We conduct a set of experiments to investigate the influence of training data size on the accuracy. We run the experiments for 5 times and report the error-bar of results in Figure 9.8. Our approach achieves an accuracy of 0.9% even when only 12.5% of the available data set is used for the training. This is rather promising and suggests that our model has a low dependency on the size of the training data.

9.4.4 Robustness Evaluation

When an EEG-based system is deployed in the real world, the typical usage would always be in a multi-trail setting. That is the data used to train the system is

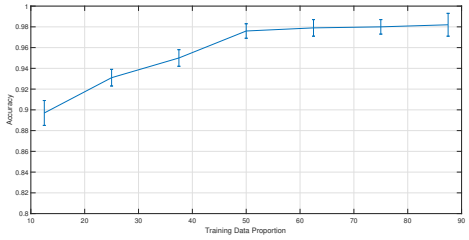


Figure 9.8: The accuracy change trend with training data size

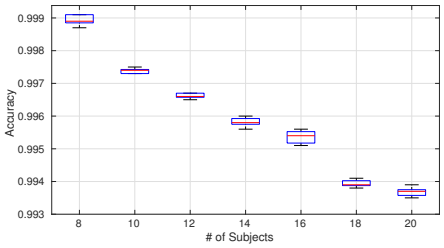


Figure 9.9: The relationship between training subject number and accuracy

collected in one trial (i.e., one set of circumstances) which would be different form the conditions in which the system is employed for user identification. Note that the placement of the EEG headset on the user's skull may vary slightly for each usage. For example, the user wears the EEG headset and collects the first trial data; then collects the second trial data after he/she removes the headset and puts it back again. There may be some difference between two trials data, which is caused by the different placement position or other internal equipment reasons. Therefore, The divergence of the training data and testing data should be considered when the identification system is designed.

In this section, we evaluate the robustness of the proposed approach by analyzing whether the trial setting (single vs multi-trial) affects the identification accuracy. Two datasets, which respectively contain single-trial identification data (EID-S) and multi-trial identification data (EID-M), are employed.

The evaluation of EID-S is shown in Table 9.7, we can observe that our approach achieves the overall accuracy of **0.9882%** and the precision for all classes is greater 0.98. To gain further insight, the confusion matrix (Table 9.4a) and ROC curves (Figure 9.4b) are provided. The performance of MindID with EID-M was reported in Section 9.4.2 (Figure 9.3a, Table 9.5, and Figure 9.3b). The multi-trial setting results in a very slight decrease (0.9882 to 0.982) in the accuracy. However, the impact of inter-trial divergence is rather minimal (0.062). This suggests that MindID has the potential to be deployed in the real-world and achieve repeatable and accurate results in diverse conditions.

Table 9.7: Evaluation report of EID-S dataset. The overall accuracy achieves 0.9882 of 7000 testing samples.

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**Average/Total**
Precision	0.9897	0.9881	0.9944	0.9837	0.9895	0.9844	0.9866	0.9897	**0.9882**
Recall	0.992	0.9924	0.9944	0.9712	0.986	0.9939	0.9789	0.9977	0.9883
F1-score	0.9908	0.9903	0.9944	0.9774	0.9878	0.9891	0.9827	0.9937	0.9883

9.4.5 Adaptability Evaluation

To examine the adaptability and consistency, our model is evaluated using another dataset (EEG-S) which is collected from a more precise EEG equipment, BCI 2000 which has 64 channels and collects signals at 160Hz. However, this headset is rather inconvenient to the subject. We selected a subset of this publicly available headset such that it matches the sample size of user population of our local dataset (EID-S), i.e., 56,000 samples from 8 subjects.

The results presented in Table 9.8 illustrate that our model achieves an accuracy of **0.9989** while all other metrics (precision, recall, and F1-score) are greater than 0.995. The confusion matrix and ROC curves are given in Figure 9.5a and Figure 9.5b, respectively. The accurate classification of EEG-S demonstrates that our approach has good adaptability and able to handle different situations (such as different types of EEG equipments).

Comparing with the results for EID-S (Figure 9.4a, Table 9.7, and Figure 9.4b), we observe a slight improvement with EEG-S of about 0.01. We attribute this to the improved precision of the EEG headset in the number of channels (64 vs 14) and a higher sampling rate (160Hz vs 128Hz).

Section 9.4.4 and 9.4.5 illustrate that our approach is robust and adaptable and thus has the potential for practical deployment in many different environments.

Table 9.8: Evaluation report of EEG-S dataset. The overall accuracy achieves 0.9989 of 7000 testing samples.

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**Average/Total**
Precision	1	0.9988	0.9988	0.9957	1	1	0.9988	0.9989	**0.9989**
Recall	1	0.9988	0.9988	0.9989	1	0.9988	0.9964	0.9989	0.9988
F1-score	1	0.9988	0.9988	0.9973	1	0.9994	0.9976	0.9989	0.9989

9.4.6 Effect of User Population Size

The user population size is an important factor that can influence the performance of the identification system. Intuitively, as the target user population size increases, there is less distinction between the EEG signals of the individual subjects, which is thus likely to impact the identification accuracy. In this section, we design extensive experiments in order to explore the influence of the user population size. The dataset EEG-S-L contains EEG data collected from 20 subjects. We vary the total number of users in the target population group from 8 to 20 (in increments of 2) and plot the accuracy results in Figure 9.9. It is evident that there is a slight decrease in the accuracy from 0.9989 for 8 subjects to 0.9937 for 20 subjects. However, the accuracy is still over 99% and thus rather competitive. Furthermore, from Figure 9.9, we can observe that the derivative of the relationship curve is negative, which suggests that the proposed approach is likely to be effective for even larger population sizes.

9.4.7 Comparison of Different EEG Frequency Patterns

This section presents experiments to validate the hypothesis proposed in Section 9.2, which claims that the Delta pattern signals contain most distinguishable information for identification. In this experiment, we use the 3 dataset (EID-M, EID-S, and EEG-S) and decompose EEG signals into 6 frequency patterns, namely: Delta, Theta, Alpha, Beta, Gamma, and Full-frequency. The last set contains the entire frequency band of the EEG signals from 0 to $128Hz$. Since the sampling rate of the EEG signals is $128Hz$, the Butterworth filter employs a frequency range of 0 - 64

Table 9.9: EEG Pattern Decomposition Analysis

Dataset	Methods	EEG Patterns						Best Result
		Delta	Theta	Alpha	Beta	Gamma	Full	
EID-M	**Jayarathne [364]**	0.919	0.701	0.725	0.598	0.602	0.785	**0.982 (Delta)**
	Bashar et al. [351]	0.873	0.716	0.425	0.393	0.412	0.571	
	Keshishzadeh et al. [374]	0.815	0.672	0.536	0.273	0.409	0.511	
	SVM	0.143	0.157	0.137	0.135	0.138	0.2745	
	RF	0.936	0.707	0.677	0.489	**0.435**	0.7935	
	KNN	0.941	**0.804**	0.618	0.35	0.313	0.819	
	AdaBoost	0.251	0.13	0.15	0.15	0.171	0.24	
	LDA	0.148	0.154	0.135	0.135	0.129	0.28	
	XGB	0.965	0.665	0.69	0.495	0.414	0.815	
	RNN	0.917	0.709	0.708	0.518	0.411	0.813	
	Ours	**0.982**	0.713	**0.73**	**0.513**	0.423	**0.822**	
EID-S	**Jayarathne [364]**	0.938	0.799	0.764	0.602	0.663	0.828	**0.9882 (Delta)**
	Bashar et al. [351]	0.884	0.760	0.437	0.413	0.452	0.597	
	Keshishzadeh et al. [374]	0.846	0.699	0.672	0.413	0.498	0.628	
	SVM	0.135	0.162	0.181	0.152	0.132	0.408	
	RF	0.947	0.771	0.719	0.587	0.377	0.863	
	KNN	0.953	**0.824**	0.714	0.472	0.495	0.853	
	AdaBoost	0.278	0.29	0.162	0.2	0.16	0.3	
	LDA	0.14	0.16	0.183	0.152	0.122	0.41	
	XGB	0.981	0.785	0.791	0.599	0.489	0.893	
	RNN	0.9425	0.7568	0.8175	**0.6331**	0.5141	0.9045	
	Ours	**0.9882**	0.821	**0.8259**	0.612	**0.517**	**0.913**	
EEG-S	**Jayarathne [364]**	0.967	0.891	0.855	0.678	0.693	0.898	**0.9989 (Delta)**
	Bashar et al. [351]	0.903	0.836	0.537	0.559	0.612	0.775	
	Keshishzadeh et al. [374]	0.928	0.832	0.732	0.611	0.589	0.801	
	SVM	0.216	0.167	0.148	0.169	0.186	0.652	
	RF	0.972	0.885	0.819	0.823	0.87	0.957	
	KNN	0.974	0.865	0.781	0.559	0.743	0.936	
	AdaBoost	0.32	0.32	0.27	0.23	0.22	0.34	
	LDA	0.186	0.17	0.28	0.168	0.162	0.6618	
	XGB	0.9972	**0.982**	0.967	0.959	0.953	0.989	
	RNN	0.9981	0.9667	0.964	0.947	0.952	0.9886	
	Ours	**0.9989**	0.972	**0.968**	**0.961**	**0.955**	**0.99**	

Hz.

We compare MindID with a subset of state-of-the-art methods and baselines as in Section 9.4.2. In particular, we select [364, 351, 374] as these methods do not rely on signals belonging to specific frequency bands but can rather use all 6 patterns under consideration. The results are shown in Table 9.9. The primary conclusions are listed as follows:

- Our approaches achieves the highest accuracy on all of the three datasets (with different trials, collection equipment, and sampling precision), which shows that our model has outstanding robustness and adaptability.

- The 11 methods including MindID achieve their best results with the Delta patterns. This result provides strong evidence to suggest that Delta pattern contain the most distinctive information for human identification and thus proves the hypothesis proposed in Section 9.2.

- Several statistical classification models (such as RF, KNN, and XGB) work well on the low-frequency patterns (Delta and Theta) but do not achieve good results with high-frequency band signals (Alpha, Beta, and Gamma).

- Deep learning methods are particularly good at extracting deep relationships between the samples which are inherently noisy and fluctuating. This conclusion can be inferred from the observations that RNN has lower accuracy than RF/KNN/XGB with Delta and Theta patterns but performs better with other patterns. These observations inspire the combination of the attention-based RNN structure and the tree-boosting classifier.

- The baselines and the state-of-the-art methods can achieve acceptable identification accuracy with high-quality EEG dataset (EEG-S) but performs poorly with the low-quality datasets (EID-M and EID-S). Consider the Full-frequency pattern as an example, RF/XGB/RNN achieves an accuracy of more than 0.95 on EEG-S but lower than 0.82 on EID-M. However, our approach consistently achieves high accuracy no matter the data quality. This suggests that MindID has the potential to deal with various real-world effects and thus a prime candidate for practical deployment.

9.5 Discussion

In this chapter, we propose an EEG-based identification approach and evaluate the robustness and adaptability over three datasets. In this section, we discuss the challenges and potential directions for future EEG-based person identification system.

- First, EEG-based identification system is less vulnerable to attacks compared to existing biometric identification systems. In order to evaluate the attack-resilience of MindID, We test our approach to dealing with the threat from unauthorized subjects among a number of attack categories [375]. We randomly select 10 subjects from EEG-S-L dataset as authorized users while the rest of users are as unauthorized subjects. During the testing, given the unauthorized users' EEG signals, MindID predicts the probability indicating how likely the samples belong to an authorized subject or not. The specific user will be regarded as unauthorized if the predict probability is under a threshold. Our experimental results demonstrate that MindID is able to precisely detect the authorized users with around 99% accuracy under an appropriate threshold setting. This suggests that our approach has the potential to distinguish the attack from unauthorized subject.

- Second, the impact of variations in the EEG signals over longitudinal scales on the performance of such identification systems needs to be studied. For a thorough investigation, it would be necessary to collect EEG data over multiple trials spread across several days. We have taken some preliminary steps in this regards by collecting EEG data from 8 subjects across 3 separate trials. However, there is scope to undertake more extensive evaluations in this regard.

- EEG signals are known to be sensitive to various factors such as the mood of the subject, intake of foods, drugs and alcohol. Knyazev [356] infers that EEG signals are affected by inherent factors such as panic, sustained pain, sexual arousal, etc. Dubbelink et al. [376] conduct experiments in obese and lean female adolescents and record the Magnetoencephalographic (MEG) signal of participants' brain. The obese adolescents had increased synchronization in delta and beta frequency bands compared to lean controls. Reid et al. [377] claim that the increase of delta power during the first 5 min following cocaine was correlated with increased ratings of cocaine craving. Reward-related decrease of delta activity has been observed after administration of legal psycho-active drugs, such as alcohol [378], tobacco [379], and caffeine

[380]. One future scope of our future work is to study how the identification system is influenced by the aforementioned factors and enhance the current approach to be more adaptive.

- The impact of population size on the performance needs further investigation. In this chapter, explore this effect to some extent but considering a corpus of 20 subjects. However, further investigations with larger groups, for example, 100 subjects, are necessary. That said, our results already demonstrate that MindID can be used in settings such as small offices which are accessed by a small group of people.

- The EEG data of an individual is known to change gradually with environmental factors such as age, mental state and lifestyle. For example, Delta patterns are known to decrease with age in older individuals [381]. This suggests that the pre-trained model used in MindID should be updated when such changes are detected. In our future work, we aim to develop an online learning system which can automatically retrain the model using the data collected during the operational phase.

- While we provide some explanations in Section 9.2 for why Delta patterns may be most informative for user identification, the underlying mechanism is still not well known. Further investigation is necessary.

- The privacy of pervasive EEG technology is not concerned in this chapter. The collected EEG data may not only contain subject ID related information but also infers other privacy of the subject (e.g., emotion and fatigue state). In our future work, we attempt to propose an algorithm to eliminate other private information in the collected EEG data.

9.6 Conclusion

Taking the advantages of EEG-based techniques for attack-resilient, we propose a biometric EEG-based identification approach to overcome the limitations of traditional biometric identification methods. We analyzed the EEG data pattern characteristics and capture the Delta pattern which takes the most distinguishable features for user identification. Based on the pattern decomposition analysis, we report the structure of the proposed approach. In the first step of identification, the preprocessed EEG data is decomposed into Delta pattern. Then an attention-based RNN structure is employed to extract deep representations of Delta wave. At last, the deep representations are used to directly identify the user' ID. The proposed approach is evaluated over 3 datasets (two local and one public dataset). The experiments results illustrate that our model achieves the accuracy of 0.982, 0.9882, and 0.9989 over three datasets, separately. The results also infer the robustness and adaptability of our model. We also outline several directions for future research in EEG-based identification.

Chapter 10

Enabling Brain Typing via Deep Feature Learning of EEG Signals

As an important pathway between human brains and the outside world [259], BCI systems allow people to communicate or interact with external devices such as wheelchairs or service robots, through their brain signals. Among the different types of brain signals, MI-EEG is especially popular and has demonstrated promising potential in discerning different brain activities in BCI systems. Motor imagery is a mental process where a subject *imagines* performing a certain action such as closing eyes or moving feet. Basically, EEG[1] is a method to analyze brain activities by measuring the voltage fluctuations of ionic current within the neurons of brains. In practice, electrodes are usually placed on the scalp for the measurement in a non-invasive and non-stationary way [382].

One of the most promising and widely discussed applications of EEG-based BCI is to enable people to type via direct brain control [383]. In this chapter, we aim at enabling a brain typing system by enhancing the decoding accuracy of EEG signals

[1]In this chapter, we will use the terms EEG and MI-EEG interchangeably.

for a wider range of brain activities (e.g., multi-class scenario). We envision a real-world implementation of such a system which can interpret the user's thoughts to infer typing commands in real-time. Motor disabled people would benefit greatly from such a system to express their thoughts and communicate with the outside world.

However, the design of a practical and effective BCI-system is faced with the following major challenges. First, EEG signals usually have very low signal-to-noise ratio [384]. As a result, EEG signals inherently lack sufficient spatial resolution and insight on activities of deep brain structures. Second, data pre-processing, parameter selection, and feature engineering are all time-consuming and highly dependent on human expertise in the domain. Third, the state-of-the-art approaches can achieve an accuracy of at most 70∼85%, which though impressive is not sufficient for widespread adoption of this technology. Fourth, existing research mainly focuses on discerning EEG signals under the binary classification situation and little work has been conducted on multi-class scenarios. Intuitively, the more scenarios an EEG-based control system can distinguish, the wider is its applicability in the real-world.

To tackle the aforementioned challenges, we propose a novel hybrid deep neural network that combines the benefits of both CNN [222] and RNN [385] for effective EEG signal decoding. Our model is capable of modeling high-level, robust and salient feature representations hidden in the raw EEG signal streams and capturing complex relationships within data via stacking multiple layers of information processing modules in a hierarchical architecture. The main contributions of this chapter are highlighted as follows:

- We design a unified deep learning framework that leverages recurrent and convolutional neural networks to capture spatial dependencies of *raw* EEG signals based on features extracted by convolutional operations and temporal correlations through RNN architecture, respectively. Moreover, an Autoencoder layer is fused to cope with possible incomplete and corrupted EEG signals to

enhance the robustness of EEG classification.

- We extensively evaluate our model using a public dataset and also a limited but easy-to-deploy dataset that we collected using an off-the-shelf EEG device. The experimental results illustrate that the proposed model achieves high levels of accuracy over both the public dataset (95.45%) and the local dataset (94.27%). This demonstrates the consistent applicability of our proposed model. We have made our local dataset and the source code used in our evaluations available to the research community to encourage further research in this area and foster reproducibility of results.

- We also present an operational prototype of a brain typing system based on our proposed model, which demonstrates the efficacy and practicality of our approach. A video demonstrating the system is made available [2].

10.1 Methodology

In this section, we first provide an overview of the proposed approach and then present the technical details.

10.1.1 Overview

The CNN and RNN have both been demonstrated to be effective for the EEG data decoding. Intuitively, we attempt to combine the advantages of them. However, the experiential experiments show that the simple concatenation of temporal and spatial features can not outperform the use of only one of them. Therefore, we design a feature adaptation method to map the stacked features to a new space

[2] `http://www.xiangzhang.info`

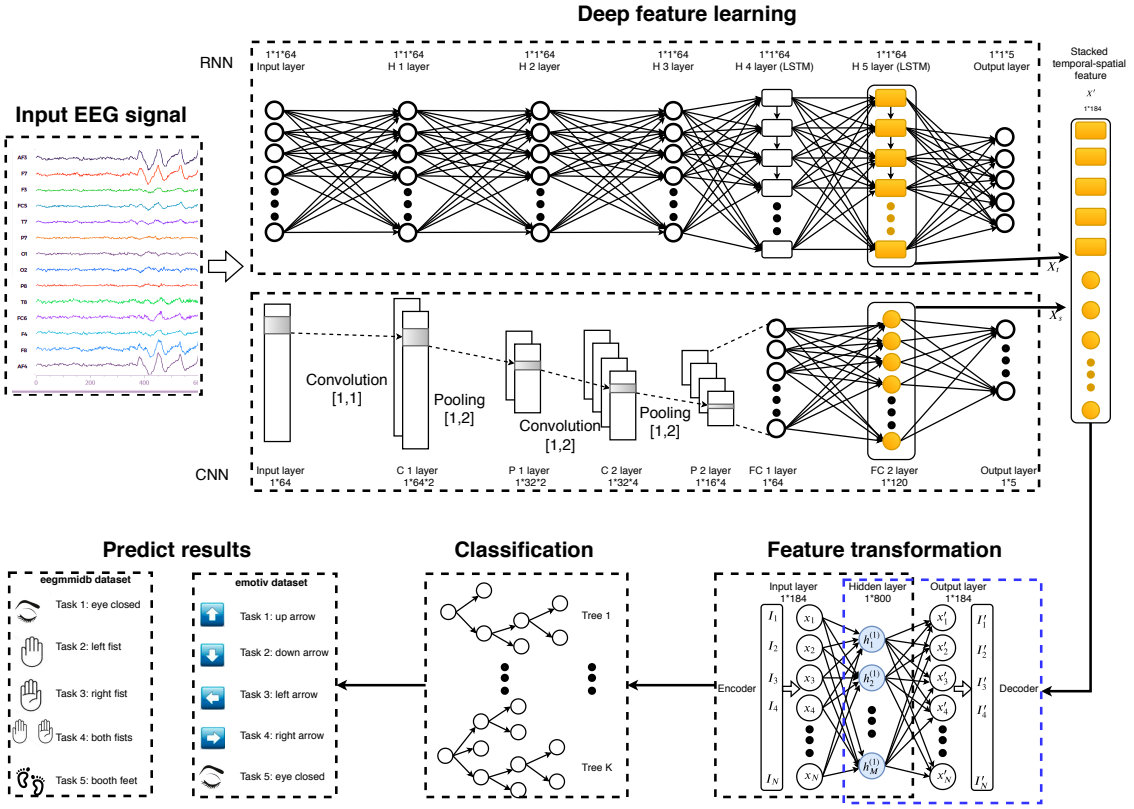


Figure 10.1: The flow chart of the proposed approach. The input raw EEG data is a single sample vector denoted by $E_{\bar{i}} \in \mathbb{R}^K$ (take $K = 64$ as an example). The C 1 layer denotes the first convolutional layer, the C 2 layer denotes the second convolutional layer, and so on. The same theory, the P 1 layer denotes the first pooling layer; the FC 1 layer denotes the first fully connected layer; the H 1 layer denotes the first hidden layer. The stacked temporal-spatial feature is generated by the FC 2 layer in the CNN and the H 5 layer in the RNN.

which can fuse the distinctive information from temporal and spatial features. Figure 10.1 illustrates the various steps involved. The essential goal of our approach is to design a deep learning model that precisely classifies the user's intents based on EEG data. In summary, we propose a hybrid approach which contains several components: deep feature learning (Section 10.1.2), feature adaption and intent recognition (Section 10.1.3).

10.1.2 Deep Feature Learning

We aim to learn the representations of the user's typing intent signal. Denote the single input EEG signal as $E_{\bar{i}} \in \mathbb{R}^K$ ($K = 64$) where K is the number of dimensions in the raw EEG signal. Next, we feed $E_{\bar{i}}$ to the RNN structure and the CNN structure for temporal and spatial feature learning in parallel. At last, the learned temporal features X_t and the spatial features X_s are combined into the stacked feature X' for the latter feature adaption (Section 10.1.3).

RNN Feature Learning

In the temporal feature processing part, the RNN structure is employed for its powerful ability for temporal feature extraction in time-series data. RNN, which is one class of deep neural network, is able to explore the feature dependencies over time through an internal state of the network, which allows it to exhibit dynamic temporal behavior. In this section, we take advantages of this trait to represent the temporal feature of the input EEG signal.

We design an RNN model consisting of three components: one input layer, 5 hidden layers, and one output layer. he number of hidden layers is optimized by Orthogonal Array Tuning method. Through the experiential experiments, the hidden layers are designed to including 3 fully connected neural networks and two layers of Long Short-Term Memory (LSTM) [230] (shown as rectangles in Figure 10.1) cells among the hidden layers. Besides, the experiential experiments also show that the activation function cannot improve the RNN performance but bring overfitting. Therefore, the non-linear transition is not employed in this chapter. Assume a batch of input EEG data contains n_{bs} (generally called batch size) EEG samples and the total input data has the 3-D shape as $[n_{bs}, 1, 64]$. Let the data in the i-th layer ($i = 1, 2, \cdots, 7$) be denoted by $X_i^r = \{X_{ijk}^r | j = 1, 2, \cdots, n_{bs}, k = 1, 2, \cdots, K_i\}, X_i^r \in \mathbb{R}^{[n_{bs}, 1, K_i]}$, where j denotes the j-th EEG sample and K_i denotes the number of dimensions in the i-th layer.

Assume that the weights between layer i and layer $i+1$ can be denoted by $W^r_{i(i+1)} \in \mathbb{R}^{[K_i, K_{i+1}]}$, for instance, W^r_{12} describes the weight between layer 1 and layer 2. $b^r_i \in \mathbb{R}^{K_i}$ denotes the biases of i-th layer. The calculation between the i-th layer data and the $i+1$-th layer data can be denoted as

$$X^r_{i+1} = sigmoid(X^r_i * W^r_{i,i+1} + b^r_i) \tag{10.1}$$

Please note the sizes of X^r_i, $W^r_{i,i+1}$ and b^r_i must match. For example, in Figure 10.1, the transformation between H1 layer and H2 layer, the sizes of X^r_3, X^r_2, $W_{[2,3]}$, and b^r_2 are separately $[1, 1, 64]$, $[1, 1, 64]$, $[64, 64]$, and $[1, 64]$.

The 5-th and 6-th layers in the designed structure are LSTM layers, so the calculation in these layers are the same as Equation 3.1 to Equation 3.6. At last, we obtain the RNN prediction results X^r_7 and employ cross-entropy as the cost function. The cost is optimized by the AdamOptimizer algorithm [231]. X^r_6 is the data in the second last layer, which has a directly linear relationship with the output layer and the prediction results. If the predicted results have high accuracy, X^r_6 is enabled to directly map to the sample label space and has the better representative of the input EEG sample. Therefore, we regard X^r_6 as the temporal feature extracted by the RNN structure and call it X_t.

CNN Feature Learning

While RNN is good in exploring the temporal (inter-sample) relevance, it is unable to appropriately decode spatial feature (intra-sample) representations. To exploit the spatial connections between different features in each specific EEG signal, we design a CNN structure. CNN is well-suited to extract the spatial relevance of the 2-D input data efficiently. In this chapter, we implement the CNN on the 1-D EEG data. As shown in Figure 10.1, the designed CNN is stacked in the following order: the input layer, the first convolutional layer, the first pooling layer, the second

convolutional layer, the second pooling layer, the first fully connected layer, the second fully connected layer, and the output layer.

The input is the same EEG data as the RNN. The input EEG single sample $E_{\bar{i}}$ has shape $[1, 64]$. Suppose the data in the i-th layer ($i = 1, 2, \cdots, 8$) is denoted by $X_i^c, X_i^c \in \mathbb{R}^{[1, K_i^c, d_i]}$, where K_i^c and d_i separately denote the dimension number and the depth in the i-th layer. The data in the first layer only has depth 1 and $X_1^c = E$. We choose the convolutional filter with size $[1, 1]$ and the stride size $[1, 1]$ in the first convolution. The stride denotes the x-movements and y-movements distance of the filter. The padding method is selected as same shape zero-padding, which results in the sample shape keeping constant in the convolution calculation. The depth of EEG sample transfers to 2 through the first convolutional layer, so the shape of X_2^c is $[1, 64, 2]$. There is a ReLU activation function designed to work on the convolutional results.

The pooling layer is a non-linear down-sampling transformation layer. There are several pooling options, with *max pooling* being the most popular [389]. The max pooling layer scans through the inputs along with a sliding window with a designed stride. Then it outputs the maximum value in every sub-region that the window is scanned. The pooling layer reduces the spatial size of the input EEG features and also prevents overfitting. In the first pooling layer (the 3rd layer of the CNN), we choose the $[1, 2]$ window and $[1, 2]$ stride. The maximum in each $[1, 2]$ window will be output to the next layer. The pooling does not change the depth and the shape of X_3^c is $[1, 32, 2]$. Similarly, the second convolutional layer chooses $[1, 2]$ filter and $[1, 1]$ stride and results in a shape of $[1, 32, 4]$. The results are made non-linear by the ReLU unit. The second pooling layer selects $[1, 2]$ window and $[1, 2]$ stride and obtains the shape as $[1, 16, 4]$.

In the full connected layer, the high-level reasoning features, extracted through previous convolutional and pooling layers, are unfolded to a flattened vector. For example, the data of the second pooling layer (X_5^c with shape $[1, 16, 4]$) is flattened to the vector with shape $[1, 64]$ (X_6^c). Then the output data can be calculated by

following the regular neural network operation:

$$X_7^c = sigmoid(T(X_6^c)) \tag{10.2}$$

$$X_8^c = softmax(T(X_8^c)) \tag{10.3}$$

At last, we have the CNN results X_8^c with shape $[1, 5]$ and employ the cross-entropy as the cost function. The cost is optimized by the AdamOptimizer algorithm. X_7^c has a directly linear relationship with the output layer and the predicted results. Therefore, we regard X_7^c as the spatial feature extracted by the CNN structure and call it X_s. The proposed approach can automatically learn the distinguishable features from 1-D EEG signals through the CNN structure. The order of the channels does not matter if the training dataset and the testing dataset have the same order. No effort is needed to transfer the 1-D data to 2-D for spatial feature learning.

In summary, the temporal features X_t and the spatial features X_s are learned through the parallel RNN and CNN structures. Both of them have the direct linear relationship with the EEG sample label, which means that they represent the temporal and spatial features of the input EEG sample if both RNN and CNN have high classification accuracy. Next, we combine the two feature vectors into a flattened stacked vector, $X' = \{X_t : X_s\}$.

10.1.3 Feature Adaptation

Next, we design a feature adaptation method to map the stacked features to a correlative new feature space which can fuse the temporal and spatial features together and highlight the useful information.

To do so, we introduce the Autoencoder layer [259] to further interpret EEG signals, which is an unsupervised approach to learning effective features. The Autoencoder is trained to learn a compressed and distributed representations for the stacked EEG

feature X'. The input of Autoencoder is the stacked temporal and spatial feature X'. Assume h, $\acute{X}'$ denote the hidden layer and output layer data, respectively.

The data transformation procedure is described as the following:

$$\acute{X}' = sigmoid(T(sigmoid(T(X')))) \tag{10.4}$$

The cost function measures the difference between X' and $\acute{X}'$ as MSE (mean squared error) which is back-propagated to the algorithm to adjust the weights and biases. The error is optimized by the RMSPropOptimizer [390]. The data in the hidden layer h is the transferred feature, which is output to the classifier. Finally, the Extreme Gradient Boosting classifier (XGBoost) is employed [260] to classify the EEG streams. It fuses a set of classification and regression trees (i.e., CART) and exploits detailed information from the input data. It builds multiple trees and each tree has its leaves and corresponding scores.

10.2 Experiments

Next, we evaluate the proposed deep learning model using a public dataset and a local dataset collected by ourselves. First, a public EEG dataset (called *eegmmidb*) is used to assess our proposed deep learning model. In addition, we evaluate our model on a local dataset for demonstrating the good adaptability of the proposed method (the collected EEG dataset is called *emotiv*) and present the corresponding results (Section 10.2.5).

10.2.1 Experimental Setting

The public dataset is described in Section 3.3.1. The motor imagery tasks and labels, along with and the corresponding typing command in the brain typing system are

Table 10.1: The motor imagery tasks and labels and the corresponding typing command in the brain typing system

Dataset	Item	Task 1	Task 2	Task 3	Task 4	Task 5
eegmmidb	intent	eye closed	left hand	right hand	both hands	both feet
	label	0	1	2	3	4
emotiv	intent	up arrow	down arrow	left arrow	right arrow	eye closed
	label	0	1	2	3	4
	Command	up	down	left	right	confirmation

shown in Table 10.1. To evaluate the performance of the classified results, we adopt several typical evaluation metrics such as accuracy, precision, recall, F1 score, ROC curve, and AUC.

10.2.2 Overall Comparison

In this section, we report the performance study and then demonstrate the efficiency of our approach by comparing with the state-of-the-art methods and other independent deep learning algorithms. Recall that the proposed approach is a hybrid model which uses RNN and CNN for feature learning, AE layer for feature adaption, and XGBoost classifier for intent recognition. In this experiment, the EEG data is randomly divided into training dataset (21,000 samples) and testing dataset (7,000 samples). The accuracy of our method is calculated as the average of 5 runs on 10 subjects.

Firstly, we report that our approach achieves the classification accuracy of **0.9545**. To take a closer look at the result, the detailed confusion matrix and classification reports are presented in Table 10.2 and Table 10.2. We can observe that for every class, our approach achieves an average precision no lower than 0.939. Figure 10.3 shows the ROC curves of the 5 classes.

Additionally, the accuracy comparison between our method and other state-of-the-art and baselines are listed in Table 10.3. Wavelet transform and independent component analysis (ICA) are state-of-the-art methods to process EEG signals.

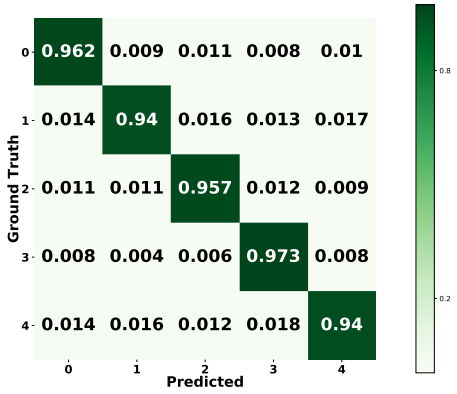


Figure 10.2: Confusion Matrix

Labels	Precision	Recall	F-1	AUC
0	0.9618	0.9380	0.9497	0.9982
1	0.9404	0.9084	0.9241	0.9977
2	0.7652	0.7930	0.7788	0.8931
3	0.9574	0.9257	0.9413	0.9990
4	0.9732	0.9028	0.9367	0.9990

Table 10.2: Evaluation over *eegmmidb*

The Deep Neural Network, Random Forest (RF) and Linear Discriminant Analysis (LDA) are applied to classify the EEG data. In addition, the key parameters of the baselines are listed here: KNN (k=3), Linear SVM ($C = 1$), RF ($n = 500$), LDA ($tol = 10^{-4}$), and AdaBoost ($n = 500, lr = 0.3$). To be fair, all the comparable methods in Table 10.3 are working on the same dataset with their suggested best setting. The results show that our method achieves the significantly higher accuracy of **0.9545** compared to all the state-of-the-art methods. Our method also performs better than other deep learning methods like RNN or CNN. Moreover, compared with most existing EEG classification research focusing on binary classification, our method works in multi-class scenario but still achieves a high-level of accuracy.

To demonstrate the advantage of our proposed hybrid model for better learning of robust features from raw EEG data, we also compare our method (joint RNN and CNN) with the independent deep feature learning method (RNN, CNN). All extracted features are classified by a XGBoost classifier. The experimental results are listed in Table 10.4, where we can see that our approach outperforms RNN and CNN in classification accuracy by 3.38% and 11.44%, respectively. Our approach also achieves the lowest standard deviation and range, implying that it is more stable and reliable. Note that the RNN on its own (RNN works as both feature extract method and classifier) without feature representation achieves a higher accuracy of 0.9325 (in Table 10.3) than the *RNN+AE+XGBoost* method (RNN works as

Table 10.3: Performance comparison with the state of the art methods. In the Binary/Multi column, B denotes Binary classification and M (N) denotes N-class classification.

	Index	**Methods**	**Binary/Multi**	**Acc**
State of the art	1	Almoari [217]	B	0.7497
	2	Sun [219]	B	0.65
	3	Mohammad [295]	B	0.845
	4	Major [218]	B	0.68
	5	Shenoy [220]	B	0.8206
	6	Tolic [235]	B	0.6821
	7	Rashid [292]	B	0.92
	8	Ward [227]	M (3)	0.8
	9	Sita [294]	M (3)	0.8724
	10	Pinheiro [236]	M (4)	0.8505
Baselines	8	KNN	M (5)	0.8769
	11	SVM	M (5)	0.5082
	12	RF	M (5)	0.7739
	13	LDA	M (5)	0.5127
	14	AdaBoost	M (5)	0.3431
	15	RNN	M (5)	0.9325
	16	CNN	M (5)	0.8409
	17	Ours	M (5)	**0.9545**

Table 10.4: The recognition accuracy of 10 subjects under different feature learning methods. The improvement represents the increase amplitude of our method over the maximum of RNN and CNN feature learning methods.

Feature learning	**S1**	**S2**	**S3**	**S4**	**S5**	**S6**
RNN	0.9005	0.8928	0.9506	0.9264	0.9487	0.9427
CNN	0.9021	0.5938	0.9395	0.9659	0.9013	0.9942
RNN+CNN	0.9390	0.9186	0.9784	0.9736	0.9967	0.9832
Improvement	0.0369	0.0258	0.0278	0.0077	0.0480	-0.0110
S7	**S8**	**S9**	**S10**	**Range**	**Average**	**STD**
0.9098	0.9293	0.9643	0.8498	0.1145	0.9215	0.0341
0.9273	0.6177	0.9310	0.6358	0.4004	0.8409	0.1580
0.9675	0.9245	0.9758	0.8954	0.1013	**0.9545**	0.0335
0.0402	-0.0048	0.0115	0.0456	0.0590	0.0228	0.0209

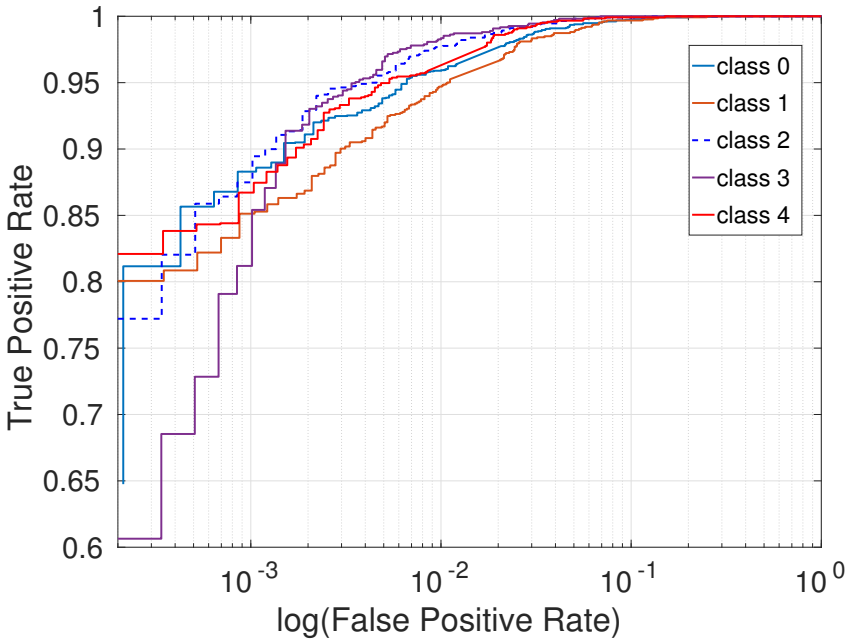
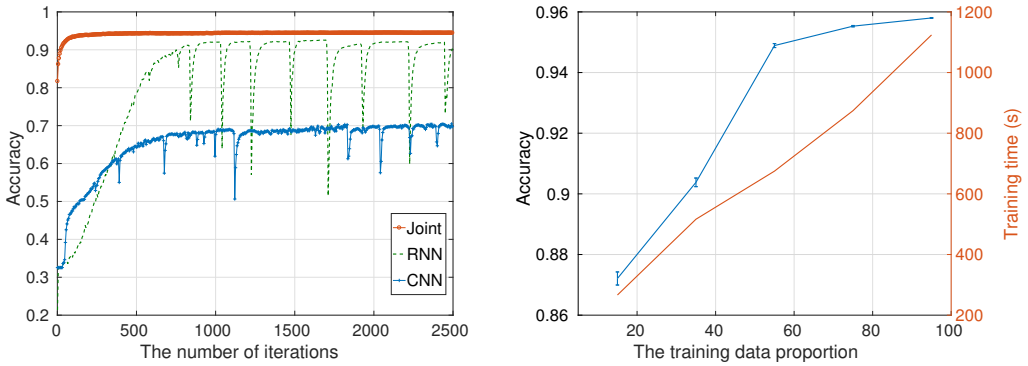


Figure 10.3: The ROC curves of the 5-class classification. Note that X-axis is the logarithmic of the False Positive Rate.

feature extract method), which exhibits an accuracy of 0.9215. This shows that RNN represented features are unsuitable for other classifiers and the inappropriate use of AE may *decrease* the signal quality. Figure 10.4a illustrates separately the accuracy changes along with the training iterations under three categories of feature learning methods. Three curves (Figure 10.4a) show that the proposed joint method converges to its high accuracy in fewer iterations than independent RNN and CNN. The learned features are fed into the AE for further processing and finally classified by the XGBoost classifier.

10.2.3 Parameter Tuning

In this section, we conduct a series of empirical studies for analyzing the impact of various parameters on the classification accuracy of the proposed approach. We extensively explore the impact of the following key factors: the training data size,

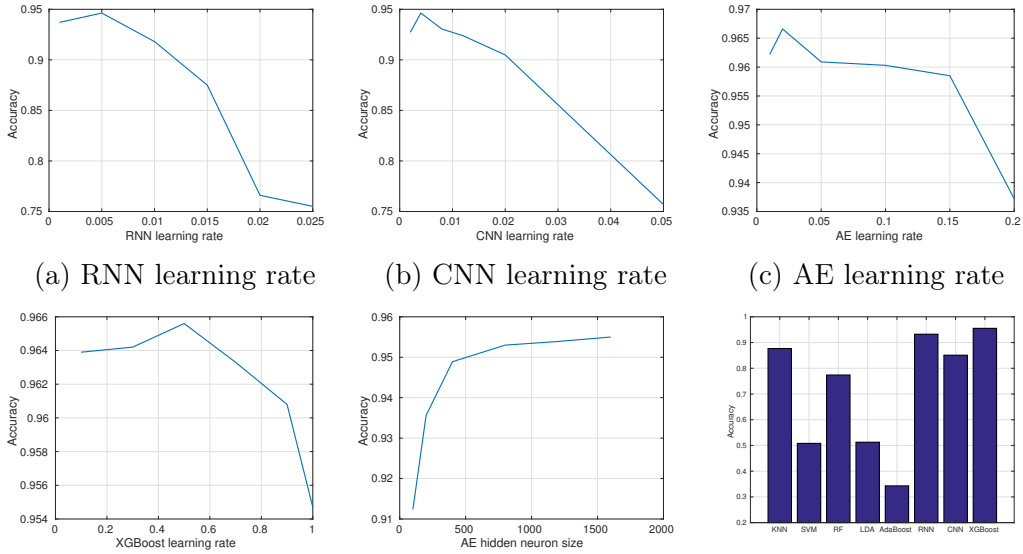


(a) The number of training iterations (b) The training data proportion

Figure 10.4: The influence of iteration numbers and training data proportion

the RNN learning rate, the CNN learning rate, the AE learning rate, the XGBoost learning rate, the AE hidden neuron size, and the classifier. We next investigate the impact of varying the data used for training on the accuracy of our model. The results are illustrated in Figure 10.4b. As expected, the accuracy increases as more data are available for training. Our method achieves an accuracy of 95% when 55% of the available data set is used for training. There is only a marginal improvement in accuracy with the inclusion of additional training data. Also, observe that we can achieve an accuracy of 87% with only 15% of training data. This indicates that our approach is less dependent on the training data size. The time required for training the model is shown on the right vertical axis in Figure 10.4b and as expected varies linearly with the size of the training data.

Figure 10.5a to Figure 10.5d show that the proposed approach performs differently under different learning rates in each component. We choose the appropriate learning rates as 0.005, 0.004, 0.002, and 0.5 for RNN, CNN, AE, and XGBoost, respectively. Figure 10.5e illustrates that the more hidden neurons in AE, the better classification results. Therefore, we choose 800 neurons as a trade-off between the accuracy and efficiency. Figure 10.5f shows that the XGBoost classifier outperforms other classifiers and achieves the highest classification accuracy over the same features refined by *RNN+CNN+AE*. It should be noted that all the not mentioned hyper-parameters are set as default value except those shown in Table 10.5. All



(a) RNN learning rate (b) CNN learning rate (c) AE learning rate

(d) XGBoost learning rate (e) AE hidden neuron size (f) Classifier

Figure 10.5: Influence of hyper-parameters

the hyper-parameters are tuned using the Orthogonal Array Tuning Method with cross-validation. [2].

10.2.4 Latency Analysis

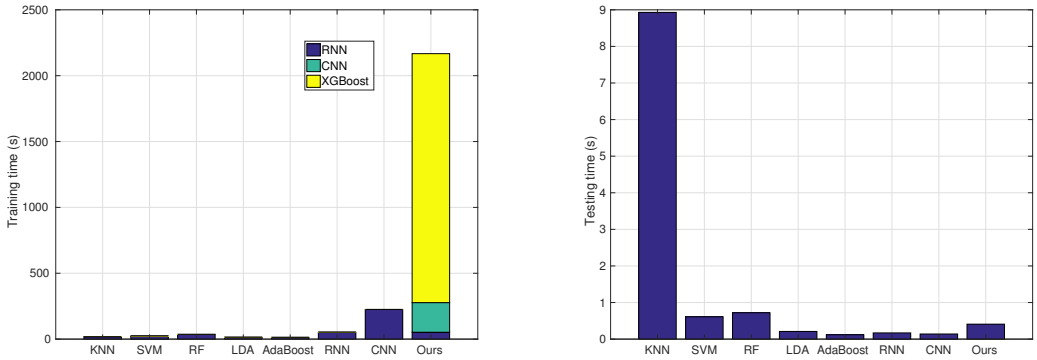
Generally, deep learning algorithms require substantial time to execute. This can limit their suitability for BCI applications (e.g., typing) which typically require close to real-time performance. For instance, the practical deployment of a BCI system could be limited by its recognition time-delay if it takes two minutes to recognize the user's intent. In this section, we will focus on the running time of our approach and compare it to the widely used baselines. The results are shown in Figure 10.6.

We first illustrate the time required to train the model in Figure 10.6a. Our model requires 2,000 seconds for training, which is significantly longer than other baseline approaches. A breakdown of the training time required for the 3 components, namely, RNN, CNN, and XGBoost is also shown. XGBoost requires the most training time as the result of its gradient boosting structure. However, training is a

Table 10.5: Hyper-Parameter Setting. For instance, RNN contains one input layer (64 neurons), 5 hidden layers (64 neurons each layer), and one output layer (5 neurons). Only the input layer neuron number is required to adjust with the dimension of the input data, all the other structures and hyper-parameters are fixed and self-adaptively.

Hyper-parameter		**Value**
RNN	Layer	7=1+5+1
	Neuron size	64*1+64*5+5*1
	Iterations	2500
	Batch size	7000
	Learning rate	0.005
	Activation function	Soft-max
	Cost function	Cross entropy
	Regularization	ℓ_2 norm ($\lambda = 0.004$)
CNN	Layer	8
	Input neuron size	64
	1st convolutional	Filter [1,1],stride [1,1], depth 2
	1st pooling	Window [1,2], stride [1,2]
	2nd convolutional	Filter [1,2],stride [1,1], depth 4
	2nd pooling	Window [1,2], stride [1,2]
	Padding method	Zero-padding
	Pooling methods	Max
	Activation function	ReLU
	1st fully connected	64
	2nd fully connected	120
	Output neuron size	5
	Iterations	2500
	Batch size	7000
	Learning rate	0.004
	Activation function	Softmax
	Cost function	Cross entropy
	Regularization	ℓ_2 norm ($\lambda = 0.001$)
AE	Layer	1+1+1
	Neuron size	184+800+184
	Iterations	400
	Learning rate	0.01
	Cost function	MSE
Classifier	Objective	Multi:softmax
	Learning rate	0.5
	max_depth	6
	Iterations	500

one-time operation. For practical considerations, the execution time of an algorithm during testing is what matters most. Figure 10.6b shows that the testing time of our approach is less than 1 second, which is similar with other baselines (except KNN which requires 9 seconds). In summary, the proposed approach takes very short testing time although it requires more time to train the model. Reducing the training time of our approach will be part of our future work.



(a) Training time (b) Testing time

Figure 10.6: The training time and testing time comparison

10.2.5 Adaptability Evaluation on Local EEG Dataset

To examine the adaptability and consistency of our model, we further evaluate our proposed model on a limited but easy-to-deploy dataset. We conduct the EEG collection by using a portable and easy-to-use commercialized EEG headset, Emotiv Epoc+ headset. The headset contains 14 channels and the sampling rate is 128 Hz. The local dataset can be accessed from this link[3]. Compared to the BCI 2000 system (64 channels) used for construct the eegmmidb dataset, our local equipment (Emotiv headset) only contains 14 channels and is much easier to be deployed in a natural environment.

Experimental Setting

This experiment is carried out using 7 subjects (4 males and 3 females) aged from 23 to 26. During the experiment, the subject wearing the *Emotiv Epoc+*[4] EEG collection headset, faces the computer screen and focuses on the corresponding *hint* which appears on the screen (shown in Figure 4.8). The brain activities and labels

[3] `https://drive.google.com/open?id=0B9MuJb6Xx2PIM0otakxuVHpkWkk`

[4] `https://www.emotiv.com/product/emotiv-epoc-14-channel-mobile-eeg/`

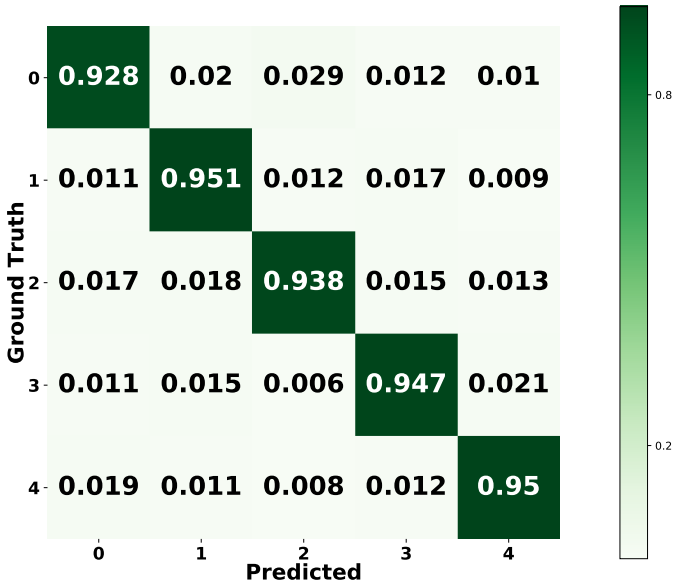
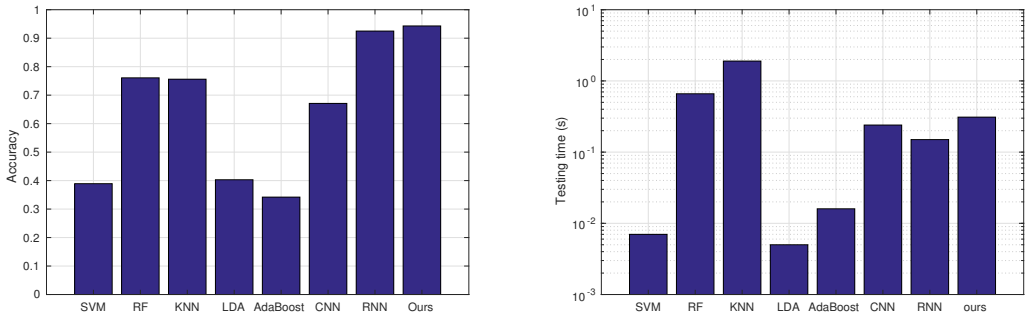


Figure 10.7: Confusion Matrix over *emotiv* dataset

used in this chapter are listed in Table 10.1. In summary, this experiment contains 241,920 samples with 34,560 samples for each subject. In order to distinguish with the aforementioned *eegmmidb* dataset, we name this dataset as *emotiv*.

Recognition Results and Comparison

For each participant, the training set contains 25,920 samples and the testing set contains 8,640 samples. The experiment parameters are the same as listed in Table 10.5. The proposed approach achieves the 5-class classification accuracy of **0.9427**. The confusion matrix is reported in Table 10.7. Subsequently, to demonstrate the efficiency of the proposed approach, we compare our method with the state-of-the-art methods and report the accuracy and testing time in Figure 10.8. To conclude, our model still achieves good performance with EEG signals collected from hardware with fewer channels and in a more natural setting.



(a) Accuracy (b) Demo Latency

Figure 10.8: The accuracy and testing time comparison over *emotiv* dataset

10.3 Application: Brain Typing System

Based on the high EEG signals classification accuracy of the proposed deep learning approach, in this section, we develop an online brain typing system to convert user's thoughts to texts. Compared with the state-of-the-art [387], the proposed application achieves a trade-off of several characteristics: non-invasive (low-cost, low-risk, and portable), complete functional (input, cancel, delete, and confirm), high speed typing, and full-dictionary[5].

The proposed system contains 5 components: EEG headset, client 1 (data collector), the server, the client 2 (typing command receiver), and the typing interface. The user wears the Emotiv EPOC+ headset (introduced in Section 10.2.5) which collects EEG signals and sends the data to client 1 through a Bluetooth connection. The raw EEG signals are transported to the server through a TCP connection. The server feeds the incoming EEG signals to the pre-trained deep learning model. The model produces a classification decision and converts it to the corresponding typing command which is sent to client 2 through a TCP connection. The typing interface receives the command and manifests the appropriate typing action.

[5]The lack-dictionary represents: after the user types in the character, for instance, 'w', s/he only has 6 choices ('A,E,H,I,O,R') for the next character instead of the overall 26 choices. The contrast situation is full-dictionary.

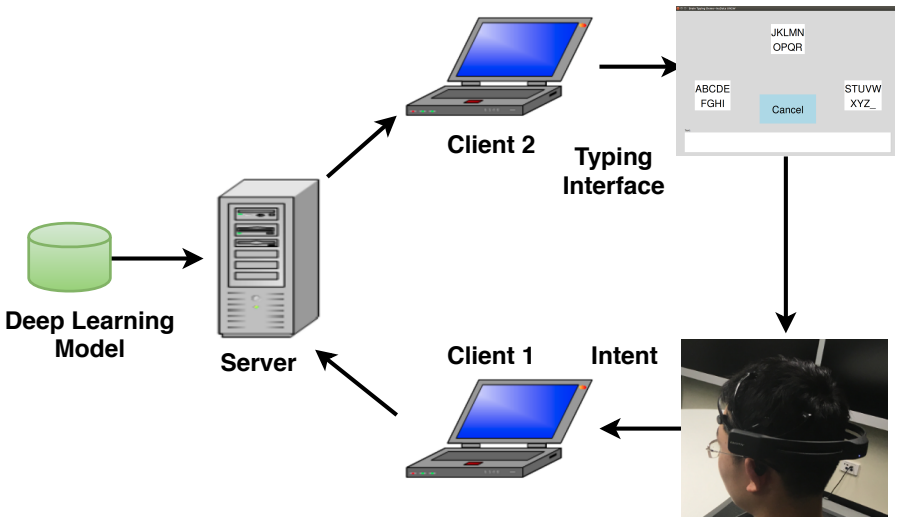


Figure 10.9: Overview of the brain typing system. The user's typing intent is collected by the headset and sent to the server through client 1. The server uses the pre-trained deep learning model to recognize the intent, which is used to control the typing interface through client 2. The server and clients are connected using TCP connections.

Specifically, the typing interface (Figure 10.10) can be divided into three levels: the initial interface, the sub-interface, and the bottom interface. All the interfaces have similar structure: three *character blocks* (separately distributed in left, up, and down directions), a *display block*, and a *cancel button*. The display block shows the typed output and the cancel button is used to cancel the last operation. The typing system in total includes $27 = 3 * 9$ characters (26 English alphabets and the space bar) and all of them are separated into 3 character blocks (each block contains 9 characters) in the initial interface. Overall, there are 3 alternative selections and each selection will lead to a specific sub-interface which contains 9 characters. Again, the $9 = 3 * 3$ characters are divided into 3 character blocks and each of them is connected to a bottom interface. In the bottom interface, each block represents only one character. As an example, Figure 10.10 shows the procedure to type the character 'I'.

In the brain typing system, there are 5 commands to control the interface: 'left', 'up', 'right', 'cancel', and 'confirm'. Each command corresponds to a specific motor imagery EEG category (as shown in Table 10.1). To type every single character,

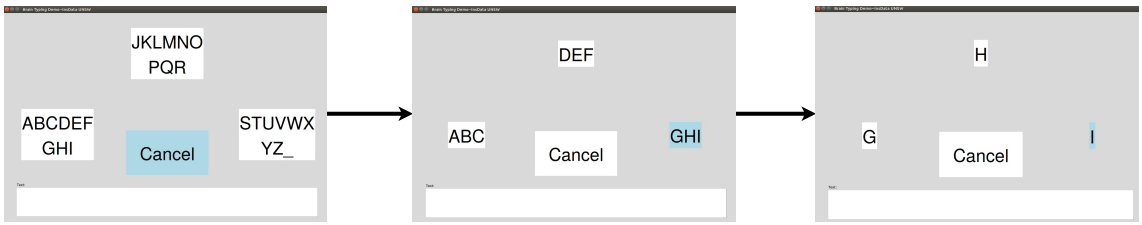


Figure 10.10: The brain typing procedure to type the character 'I'. Firstly, select the left character block (contains 'ABCDEFGHI' characters) in the *initial interface* and then confirm the selection to step in the corresponding *sub-interface*; then, select the right character block (contains 'GHI' characters) in the *sub-interface* and confirm to jump to the *bottom interface*; at last, select the right character block (only contains 'I') and the character 'I' will appear in the display block after the confirmation.

the interface is supposed to accept 6 commands. Consider typing the letter 'I' as an example (see Figure 10.10). The sequence of commands to be entered is as follows: 'left', 'confirm', 'right', 'confirm', 'right', 'confirm'. In our practical deployment, the sampling rate of Emotiv EPOC+ headset is set as 128Hz, which means the server can receive 128 EEG recordings each second. Since the brainwave signal varies rapidly and is very easy to be affected by noises, the EEG data stream is sent to server *each half second*, which means that the server receives 64 EEG samples each time. The 64 EEG samples are classified by the deep learning framework and generate 64 categories of intents. we calculate the mode of 64 intents and regard the mode as the final intent decision. Furthermore, to achieve steadiness and reliability, the server sends command to client 2 only if *three consecutive decisions remain consistent*. After the command is sent, the command list will be reset and the system will wait until 3 consistent decisions are made. Therefore, client 2 must wait for at least 1.5 seconds for a command and the entire process of typing each character takes at least 9 $(6 * 1.5)$ seconds. In other words, theoretically, the proposed brain typing system can achieve the highest typing speed of $6.67 = 60/9$ characters per minute.

10.4 Discussions

The proposed framework achieves the highest accuracy compared to the state-of-the-art EEG classification methods. The classification accuracy of the public dataset (eegmmidb) is consistently higher than the local real-world dataset (emotiv). The possible reason may be due to the different channels of two datasets (eegmmidb contains 64 channels and emotiv only takes 14 channels). In general, our framework can achieve high classification accuracy with both datasets.

The accuracy in the online mode is, however, lower than what can be achieved in an offline setting (over 95%), which could be attributed to a number of reasons. At first, the user's mental state and fluctuations in emotions may affect the quality of the EEG signals. For example, a scenario where the offline dataset used to train the deep learning model is collected when the user is in an excited emotional state but then applied in an online setting when the user is upset, would lead to low accuracy. In addition, subtle variations in the way the EEG headset is mounted on the subject's head may also impact online decision making. Specifically, the position of each of the electrodes (e.g.. the 14 electrodes in the Emotiv headset) on the scalp may vary during training and testing. Moreover, the EEG signals vary from person to person, which makes it difficult to construct a common model that applies to all individuals. Part of our future work is to identify the intra-class variabilities shared by all the activities of different subjects. Last but not least, some limitations are caused by the intrinsic attributes of the headset. For instance, the headset used in our case study is too tight for the user to wear longer than 30 minutes and the conductive quality of the wet electrodes decreases after prolonged usage.

10.5 Conclusion

In this chapter, we present a hybrid deep learning model to decode the raw EEG signal for the aim of converting the user's thoughts to texts. The model employs

the RNN and CNN to learn the temporal and spatial dependency features from the input raw EEG data and then stack them together. Our proposed approach adopts an Autoencoder to recognize the stacked feature and to eliminate the artifacts and employs the XGBoost classifier for the intent recognition. We evaluate our approach on a public MI-EEG dataset and also a real-world local dataset.

Chapter 11

Brain2Object: Printing Your Mind from Brain Signals with Spatial Correlation Embedding

In this chapter, we propose a unified approach by learning the robust structured EEG feature representations for recognizing the imagery of object seen by the individual. We first design a multi-class Common Spatial Pattern (CSP) for distilling the compact representations. CSP has proven success in extracting features using eigen decomposition based on the variance ratio between different classes [392]. Next, we propose Dynamical Graph Representation (DGR) of EEG signals to adaptively embed the spatial relationship among the channels (each channel represents one EEG electrode) and their neighbors by learning a dynamic adjacent matrix. Finally, a CNN is employed to aggregate higher-level spatial variations from the transformed graph representations.

Built on top of the aforementioned computational framework, we present a mind-controlled end-to-end system with integrated graphical interface, called *Brain2Object*.

It enables an individual to print a physical replica on an object that she is observing by interpreting visually evoked EEG signals in a real-time manner. To enable the end-to-end workflow, the proposed system gathers the user's brain activities through EEG acquisition equipment and forwards the collected EEG data to a pre-trained model which automatically recognizes the object that the user is observing. Imagine that a child observes a toy, for example Pinkie Pie (from *My Little Pony*) belonging to her friend and likes it very much and wishes that she can have one too. Brain2Object can make her wish a reality by translating her brain signals to command the 3D printer to fabricate a copy. The ability to print a replica model of any observable object could be of tremendous value to a variety of professionals including engineers, artists, construction workers, students, teachers, law enforcement, urban planners, etc. By automating the process, *Brain2Object* takes mystery of reading human mind out of the realm of experts and opens up the possibility of a wide range of BCI applications that can be useful for the masses.

To summarize, the chapter makes the following key contributions:

- We present an end-to-end digital fabrication system, *Brain2Object*, atop of the precise decoding of human brain signals that allows an individual to instantly create a real-world replica (or model) of any object in her gaze. The proposed system is able to learn an illustration of an object seen by an individual from visually-evoked EEG signals, and print a model in real-time by automatically instructing a wireless connected 3D printer.

- We propose an effective EEG decoding model by learning a dynamical graph representation, which could adaptively embed structured EEG spatial correlations during the training process. A convolutional neural network is integrated for capturing discriminative feature representations as well as the intrinsic connections among the various EEG channels.

- The proposed approach is evaluated over a large scale benchmark dataset and a limited but locally collected dataset. Our method outperforms a wide range

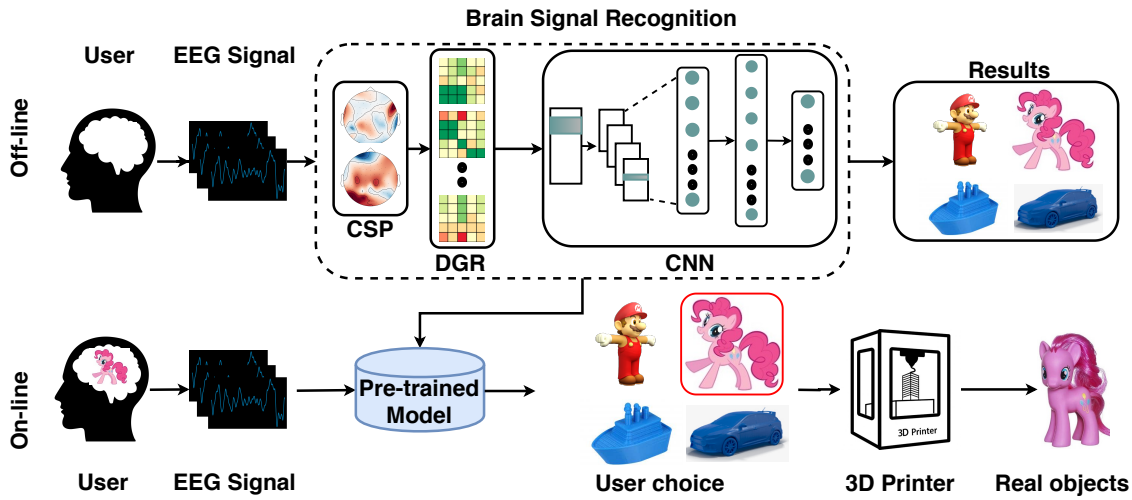
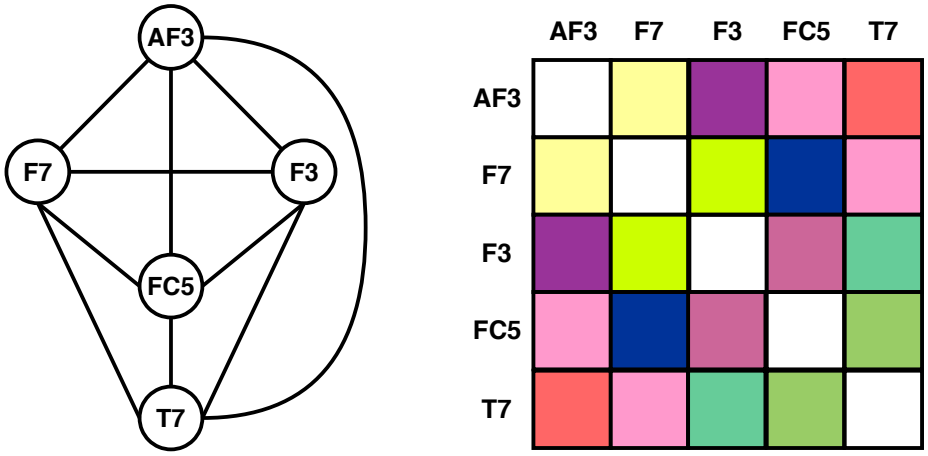


Figure 11.1: The overview of Brain2Object. The object (e.g., Pinkie Pie) observed by the user is reflected in the visually evoked EEG signals, which can be accurately recognized by the pre-trained recognition model. The recognition module employs multi-class CSP for separating the multivariate signals into additive subcomponents which have maximum differences. The spatial dependencies among processed data is extracted by DGR and then forwarded to the CNN for recognition. The schematic of the identified object is loaded from the model library of the 3D printer to fabricate a replica.

of baselines and state-of-the-art approaches in both instances, thus demonstrating the generalized nature of the approach. Finally, a prototype implementation demonstrates the practicality of *Brain2Object*.

11.1 The Proposed System

The overall aim of *Brain2Object* is to automatically recognize the object that the user desires to fabricate by analyzing her visually-evoked brain signals and actuate a 3D printer accordingly. As shown in Figure 11.1, the pre-trained recognition module employs multi-class CSP for separating the multivariate signals into additive subcomponents which have maximum differences. The spatial dependencies among processed data is extracted by DGR and then forwarded to the CNN for recognition.



(a) Graph (b) Adjacency matrix

Figure 11.2: Example of a complete weighted undirected graph with 5 vertices and the corresponding adjacency matrix. The five vertices are reading from Frontal (F) and Temporal (T) lobes of human brain. The adjacency matrix is symmetric matrices, in which the colors denote the connection weights.

The schematic of the identified object is loaded from the model library of the 3D printer to fabricate a replica.

CSP is widely used in BCI field to find spatial filters which can maximize variance between classes and achieves comparable performance [406]. In this paper, we adopt the one-vs-others strategy for multi-class CSP analysis.

11.1.1 Dynamical Graph Representation

We propose DGR to transform the CSP processed signals to a new space since graph representation has been shown to be helpful in refining and capturing spatial information [402]. In post-CSP processed EEG data $\bar{\boldsymbol{E}}$, each channel (row) separately provides the voltage amplitude of a specific electrode instead of the aggregated spatial information. The signals are discrete and discontinuous in the spatial domain. Hence, traditional spatial feature representation methods such as CNN are not well suited for further processing [402]. Instead, we invoke the knowledge of the connec-

tions of the brain neurons to map $\bar{\boldsymbol{E}}$ to a new space where each element represents not only the specific channel amplitude but also the spatial relationship with its neighboring channels.

For this purpose, we regard the brain network as a complete weighted undirected graph with M vertices where each vertex denotes a channel. The term 'complete' denotes each vertex is connected to all the residual vertices in this graph. The graph can be defined as $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{A}\}$ where $\mathcal{V} \in \mathbb{R}^M$ denotes the set of vertex with the number of $|\mathcal{V}| = M$ and $\mathcal{E}$ denotes the set of edges connecting the vertices. Suppose $\mathcal{A} \in \mathbb{R}^{M \times M}$ denotes the adjacency matrix representing the connectivity within $\mathcal{V}$. In particular, the element in the i-th row and j-th column of the adjacency matrix measures the weight or importance of the edges between the i-th and the j-th vertices.

The graph representation is dynamic, which means that the elements of the adjacency matrix are adaptively updated with the evolution of the model during training.. Hence, the name, Dynamic Graph Representation (DGR). Figure 11.2 illustrates an example of a complete weighted undirected graph which is composited by five vertices which are reading from Frontal (F) and Temporal (T) lobes of the human brain. The diagonal elements are zero since each vertex is not connected to itself. However, the proposed representation should also contain information representative of each individual vertex. To incorporate this information, we include an identity matrix I. The resulting DGR is

$$\boldsymbol{E}' = (\mathcal{A} + \boldsymbol{I})\bar{\boldsymbol{E}} \tag{11.1}$$

The represented data $\boldsymbol{E}'$ with shape $[M, L]$ can dynamically learn the intrinsic relationship between different EEG channels by training a neural network and thus benefit most from discriminative EEG feature extraction.

11.1.2 Object Recognition

The DGR representation of the EEG signals (with shape $[M, L]$) serves as input to a specified CNN structure for feature refining and classification. CNN could capture the distinctive dependencies among the patterns associated to different EEG categories. The designed CNN comprises of one convolutional layer followed by three fully-connected layers (as shown in Figure 11.1). We choose D convolutional filters with size $[2, 2]$, stride size $[1, 1]$, and zero padding. In the convolutional operation, the feature maps from the input layer are convolved with the trainable filters and fed to the activation function to generate the output feature map. For a specific convolutional area $\boldsymbol{x}$ which has the same shape as the filter, the convolutional operation can be described as

$$\boldsymbol{x}' = tanh(\sum_i \sum_j \boldsymbol{f}_{ij} * \boldsymbol{x}_{ij})$$

where $\boldsymbol{x}'$ denotes the filtered results while $\boldsymbol{f}_{ij}$ denotes the i-th row and the j-th column element in the trainable filter. We adopt the widely used Tanh activation function for nonlinearity. The output of convolutional layer has shape $[M, L, D]$ and then be flattened to $[1, M * L * D]$ and fed into the first fully-connected layer. Thus, the first fully connected layer has $M * L * D$ neurons. Afterward, the second and the third fully-connected layers have D' and K neurons, respectively. The operation between the fully-connected layers can be represented by

$$\boldsymbol{E^{h+1}} = softmax(\bar{\boldsymbol{w}}\boldsymbol{E^h} + \bar{\boldsymbol{b}}) \tag{11.2}$$

where h denotes the h-th layer and $\bar{\boldsymbol{w}}$, $\bar{\boldsymbol{b}}$ denote the corresponding weights matrix and biases. The softmax function is used for activation. For each EEG sample, the corresponding label information is presented by one-hot label $\boldsymbol{y} \in \mathbb{R}^K$. The error between the prediction and the ground truth is evaluated by cross-entropy

$$loss = -\sum_{k=1}^{K} \boldsymbol{y}_k log(p_k) \tag{11.3}$$

where p_k denotes the predicted probability of observation of an object belonging to category k. The error is optimized by the AdamOptimizer algorithm. A drop

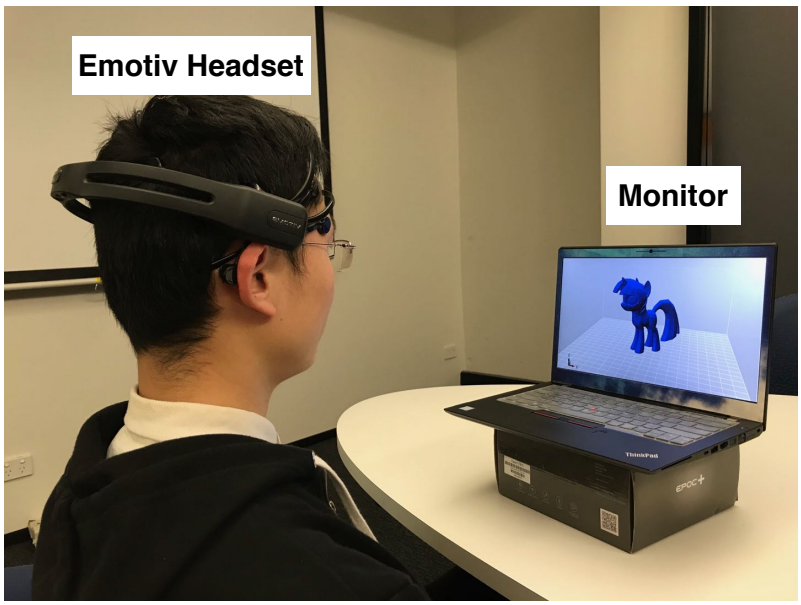


Figure 11.3: Data acquisition experiment. The participant wears the EPOC+ Emotiv headset with 14 channels siting in front of a monitor which shows the Pinkie Pie.

out layer is added after the first fully-connected layer with 0.5 drop rate in case of overfitting.

11.2 Data Acquisition

In this section, we aim to gather a local EEG dataset which reflects the user's brain voltage fluctuation under visual stimulation of a number of object images. In the ideal environment, the system is expected to recognize the EEG pattern of a random image. However, as this is a first exploration of this idea, we limit our study to include images of 4 objects: a car, a boat, Pinkie Pie Pony and Mario (from the video game).

We recruit 8 healthy participants (aged 22-27 years) including 5 males and 3 females to participate this study. The experiment is approved by the ethic committee (HC190315). The data collection is conducted in a quiet room. As shown in Fig-

ure 11.3, the subject wears the EPOC+ Emotiv EEG headset which contains 14 channels corresponding to the 10-20 system (which is an internationally recognized method to describe and apply the location of scalp electrodes). The sampling rate is set as 128 Hz and the headset can wireless connection with the computer over Bluetooth. The participants sit in a comfortable armchair, maintain a relaxed composure and gaze at a monitor placed approximately 0.5 meters in front of them. Each subject participates in 10 sessions and each session contains 4 trials.

Each trial lasts for 15 seconds and is comprised of three phases, each lasting 5 seconds. In the first phase, the monitor shows an empty slide and the subject is asked to be relaxed. In the second phase, a random object picture is presented in the middle of the screen and the subject is instructed to focus on the projected image. The final phase is identical to the first phase. Naturally, only EEF signals collected during the second phase are used in our dataset. In the second phase, the image is chosen with equal probability from the 4 aforementioned images. To keep the balance of the dataset, the final EEG data of each specific participant is composed of 40 trials where each object appears 10 times.

For each subject, there are 40 trials where each trial lasts for 5 seconds. Hence, each participant contributes 200 seconds of EEG signals. Since the sampling rate is 128 Hz, each subject contributes $25,600 = 128 \times 200$ sampling points, which means the dataset has $204,800$ sampling points in total.

11.3 Experiments

11.3.1 Datasets

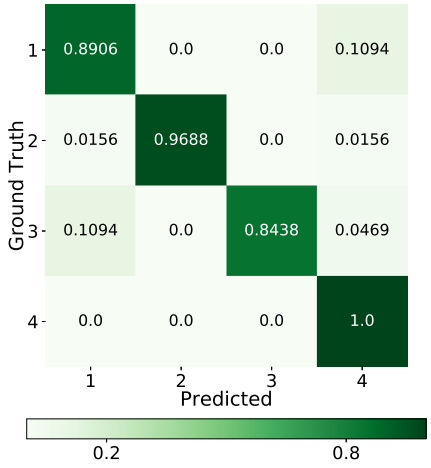
In addition to the local dataset (Section 11.2, denoted by EEG-L), our model is evaluated over a public benchmark dataset eegmmidb which is denoted as EEG-P. The EEG-P is collected by the BCI200 EEG system contains 64 channels with 160

Table 11.1: Overall comparison with state-of-the-art models and baselines over both EEG-P and EEG-L

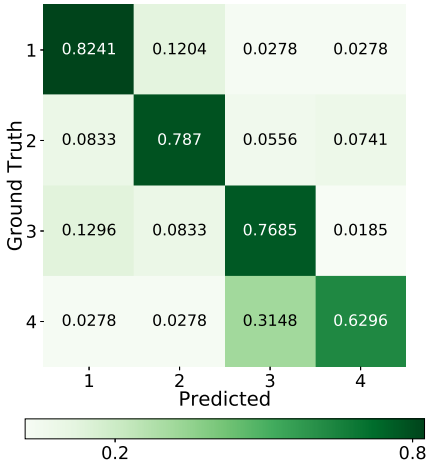
Dataset	Method	Accuracy	Precision	Recall	F-1
	KNN	0.6962	0.7325	0.7552	0.7437
	RF	0.7137	0.7536	0.7328	0.7431
	SVM	0.6692	0.7122	0.7156	0.7139
	CSP+KNN	0.9134	0.9273	0.9135	0.9203
	CNN	0.8638	0.8619	0.8722	0.8670
EEG-P	**[81]**	0.8327	0.8556	0.8559	0.8557
	[75]	0.8631	0.8725	0.8669	0.8697
	[410]	0.8915	0.9013	0.9125	0.9069
	[411]	0.7986	0.8031	0.8219	0.8124
	[9]	0.8325	0.8261	0.8433	0.8346
	Ours	**0.9258**	**0.9325**	**0.925**	**0.9287**
	KNN	0.5108	0.5212	0.5436	0.5322
	RF	0.5826	0.6258	0.6246	0.6252
	SVM	0.6538	0.6684	0.6825	0.6754
	CSP+KNN	0.5833	0.5773	0.5833	0.5803
	CNN	0.6863	0.7021	0.6038	0.6493
EEG-L	**[81]**	0.6988	0.7021	0.7086	0.7053
	[75]	0.5832	0.5968	0.6013	0.5990
	[410]	0.6892	0.6995	0.7021	0.7008
	[411]	0.6679	0.6759	0.6821	0.6790
	[9]	0.6731	0.6889	0.6921	0.6905
	Ours	**0.7523**	**0.7602**	**0.7528**	**0.7564**

Hz sampling rate. EEG data is recorded while the subjects are provided a visual stimulus (on a monitor) of certain actions and asked to imagine performing those actions. The four actions (left hand, right hand, both hands, and both feet) are labelled from 1 to 4. EEG-P has 560,000 samples belonging to 4 different labels and equally from 20 subjects.

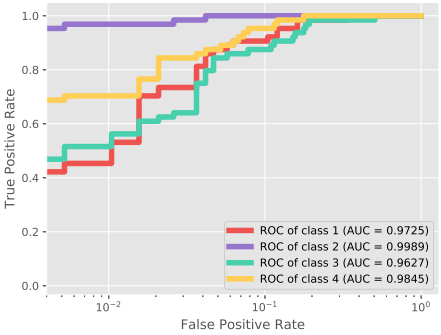
Both datasets are further sub-divided into a training set (80%) and testing set (20%). The training set is split into 4 equal mini-batches. All the features are normalized by the z-score method. The segmentation time window is set to 64 and 16 for EEG-P and EEG-L, respectively. The overlapping rates are 50%.



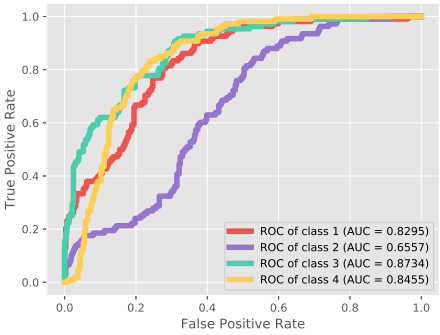
(a) EEG-P confusion matrix



(b) EEG-L confusion matrix



(c) EEG-P ROC and AUC



(d) EEG-L ROC and AUC

Figure 11.4: Confusion matrix and ROC curves with AUC score. The ROC curve of EEG-P has log scaled x-axis.

11.3.2 Overall Comparison

Next, we report the performance of Brain2Object. Recall the adopted classification method combines the multi-class CSP and the convolutional neural networks. All the experiments are run on the Titan X (Pascal) GPU and accuracy results presented are averaged over 5 runs.

First, we provide the overall comparison with several widely used baselines including KNN, Random Forest (RF), Support Vector Machine (SVM). The key parameters of the baselines are listed here: KNN with 3 nearest neighbors; SVM with RBF kernel; RF with 50 trees. The independent CNN has the identical structure of the

CNN component in our system as introduced in Section 11.1.2. The kernel and stride information have been provided above, the learning rate is set as 0.0005 and the depth of convolutional layer D equals to 10. The number of hidden neurons in the second fully-connected layer is 1000 for EEG-P and 120 for EEG-L. All the parameters are determined by empirical tuning. We also compare with a range of competitive state-of-the-art models:

- Sturm et al. [81] proposes the application of Deep Neural Networks with layer-wise relevance propagation for EEG data analysis.
- Yang et al. [75] combines augmented CSP with CNNs for motor imagery performance recognition.
- Park et al. [410] introduces an augmented complex-valued CSP based on the correlation between EEG channels.
- Thomas et al. [411] study EEG classification by selecting the subject-specific spatial and spectral features.
- Zhang et al. [9] combines Recurrent Neural Networks (RNNs) with CNN in order to extract the temporal-spatial features from brain signals.

The results are depicted in Table 11.1. Observe that our method achieves the highest accuracy (which corresponds to 0.9258 for EEG-P and 0.7523 for EEG-L) in comparison with numerous state-of-the-art approaches for both datasets. The experiments thus demonstrate the robustness, effectiveness and generality of our method. One can also readily observe that all methods achieve lower accuracy for EEG-L and as compared to EEG-P. The drawbacks of EEG-L includes low fidelity, poor spatial-temporal coverage and equipment limitations. Another reason could be the fact that our participants did not have extensive experience with the usage of EEG headsets and neither were there any specialized technicians available to assist. Finally, the emotional state of the participants may have also influenced the EEG signals.

Table 11.2: Classification report including precision, recall, and F-1 score

Dataset	Metrics	Category			
		1	**2**	**3**	**4**
EEG-P	**Precision**	0.88	1	1	0.85
	Recall	0.89	0.97	0.84	1
	F-1	0.88	0.98	0.92	0.92
EEG-L	**Precision**	0.77	0.77	0.66	0.84
	Recall	0.82	0.79	0.77	0.63
	F-1	0.8	0.78	0.71	0.72

We present a range of additional metrics for our approach. This includes the confusion matrix and ROC curves with AUC scores in Figure 11.4 and precision, recall and F-1 score for each category in Table 11.2.

11.3.3 Latency

In addition to accuracy, latency is also an important performance metric for a system such as Brain2Object.

Figure 11.5 illustrates the latency achieved by our method in comparison with a selected sub-set of baselines used in Section 11.3.2. We can observe that our approach has competitive latency compared with other methods while achieving the highest accuracy. The overall latency is less than 0.5 second. We computed the latency incurred by the different methods that are employed in our system and observed that CNN requires about 0.35 seconds for execution, while CSP and DGR together only require about 0.12 seconds. This illustrates that the use of deep learning techniques do not have a significant effect on the overall latency.

In a fully functional system, the end-to-end latency is not only comprised of the algorithmic latency but also includes the delay incurred for signal acquisition and signal transmission. The latter will be discussed in Section 11.4. In the following, we evaluate the signal acquisition latency. In the proposed system, the signal collection time is related to the acquisition equipment, in essential, the sampling rate. For

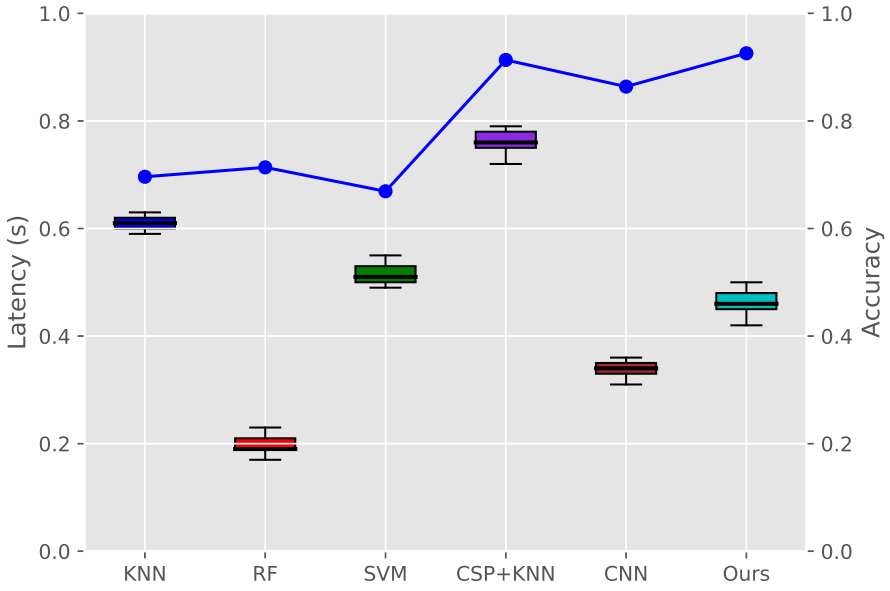
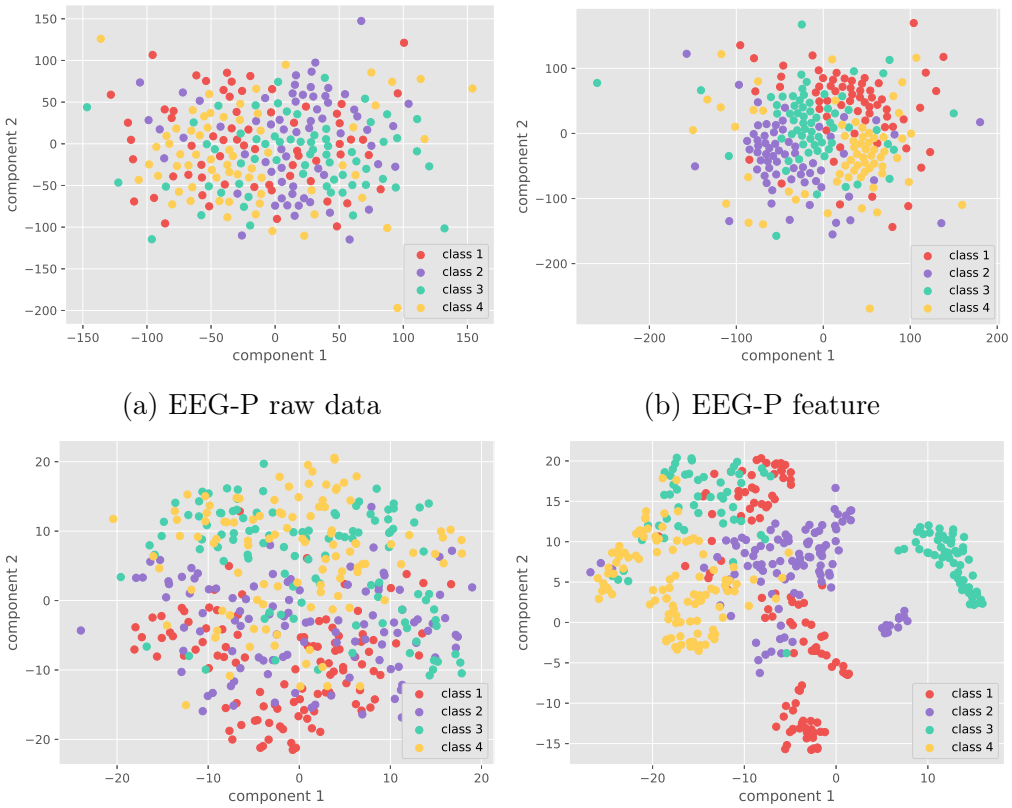


Figure 11.5: Latency comparison against the accuracy. It can be observed that our approach achieves the highest accuracy with an acceptable latency.

BCI2000, a single sample/segment is composed of 64 time points, which is gathered in 0.4 = 64/160 seconds with 160 Hz sampling frequency. On the other hand, the Epoc+ Emotiv headset only requires 0.11 = 14/128 seconds for signal collection. We can observe that the precise equipment can achieve higher accuracy but demand larger latency. In contrast, the off-the-shelf low fidelity headset has lower accuracy but also low latency. But this statement, which is similar to 'no free lunch' rule, is based on the fact that our segment length equals the channel number. Could EEG-P keep the high level accuracy with the decrease of channel amount in order to implement competitive performance with low latency at the same time? This meaningful scope deserves more attention in the future.

11.3.4 Visualization

To offer a different perspective into the performance of our system, we present a visualization of the data at two levels. At the system level, as a unified classification



(a) EEG-P raw data

(b) EEG-P feature

(c) EEG-L raw data

(d) EEG-L feature

Figure 11.6: A visualization comparing the raw data and extracted features for the two datasets. Both the raw data and the extracted feature are visualized from the corresponding testing set. This comparison demonstrates that our approach can (i) maximize the distance between the EEG data points and (ii) accurately extract the distinctive representations from the raw data.

model, we visualize the raw EEG data and the extracted distinguishable features for comparison. In Figure 11.6, the visualization of both EEG-P and EEG-L are presented. In which, Principal Component Analysis (PCA) is used for dimensionality reduction before visualization. Through the comparison, we can demonstrate that our approach maximizes the distance among EEG signals and has the ability to automatic extract the distinctive representations from raw data.

At the component level, we present the topography of various categories in each dataset. Figure 11.7 provides the EEG topographies after CSP processing. The first

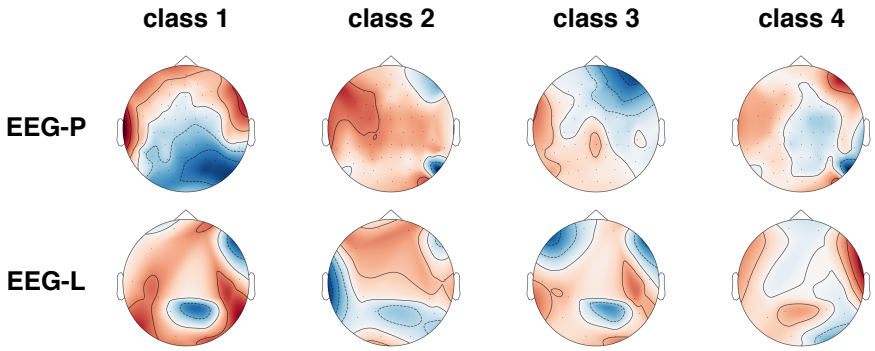


Figure 11.7: Topography after CSP processing. Each topography in the first row contains 64 channels while the second row map contains 14 channels. Through the comparison, it can be observed that the patterns belong to different categories are obviously variant, which indicates that the CSP processed features ought to be easier classified.

row represents the EEG-P dataset with 64 channels while the second row represents the EEG-L dataset with 14 channels. The channel names and positions strictly obey the international 10-20 system. Through the comparison, it can be observed that the patterns belong to different categories are obviously varying. This suggests that the CSP processed features ought to be classified easily.

11.4 Online Demonstration

In this section, we summarize our experience in developing a working prototype of Brain2Object. Figure 11.8 shows the working prototype in action. The graphical user interface is provided in Figure 11.9. The top of the interface shows the port number and baud rate of the IP printer. The IP address of the server which stores the pre-trained model and makes the object recognition decision is also shown. The main body of the interface displays object models for the four objects in our experiments, namely, Mario, car, boat and Pinkie Pie Pony.

Figure 11.10 illustrates the operational workflow of the Brain2Object demonstrator. While the user is focusing on a target object (e.g., the Pinkie Pie), the corresponding

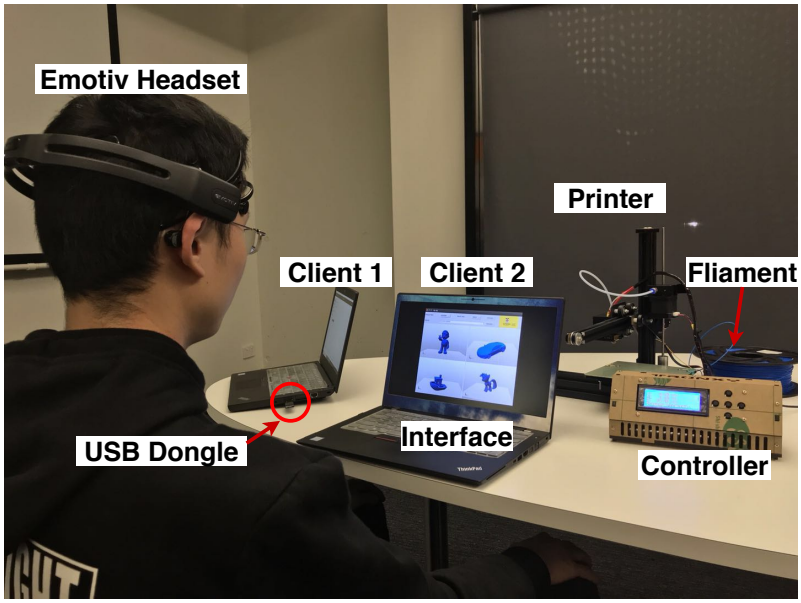


Figure 11.8: Online testing scenario. The user's EEG signals are collected by Emotiv headset for recognition. The correspond object will be printed through the 3D printer.

brain signals are collected by a properly mounted Emotiv headset and transmitted to client 1 over Bluetooth. Client 1 sends the EEG signal to the server over a TCP connection. The server loads the pre-trained EEG recognition model and classifies the EEG signal to one of the four categories. The classification result is forwarded to the interface through client 2. The interface will highlight the selected object by changing the color of the other 3 objects to gray (the selected object remains blue). Simultaneously, the selected object is dispatched to the printer driver which generates the corresponding Gcode which can be recognized by the mechanical 3D printer. Finally, the Gcode is sent to the 3D printer, which brings the object to life.

We used a Tonxy X1 desktop 3D with the following specifications. Printer size:$220 \times 220 \times 250mm$, build area: $150 \times 150 \times 150mm$, MK10 extruder diameter: 1.75mm, nozzle diameter: 0.4mm, engraving accuracy: 0.1mm, filament material: 1.75mm polylactic acid (PLA). The physical 3D model can be transmitted from a computer to the printer or directly stored in minor SD card mounted on the printer.

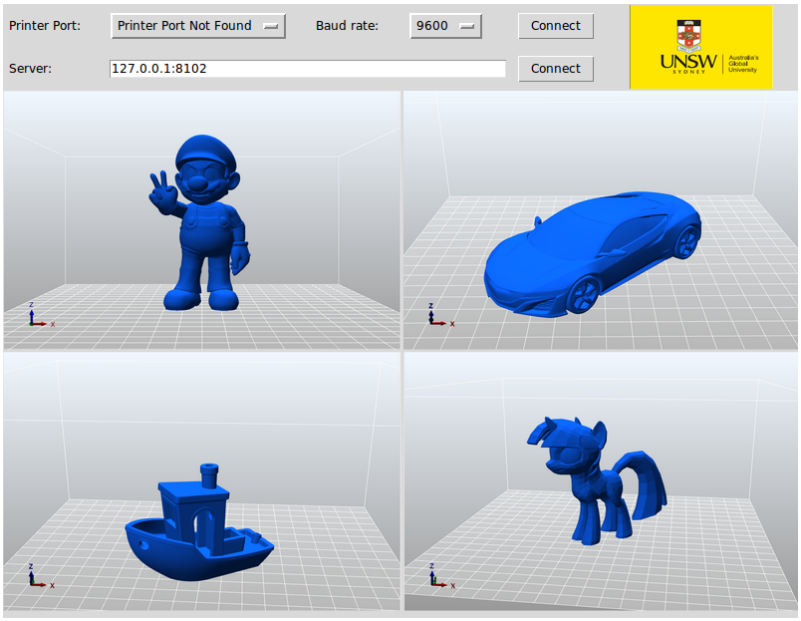


Figure 11.9: User Interface.

The sampling frequency of the Emotiv headset is 128 Hz which indicates it can collect 128 sampling point each second. The pre-trained recognition model requires each sample with 14 sampling point and each sampling point corresponds to a classification output. To achieve steadiness and reliability, the server will maintains a window of the last 10 classification output and a count of how many times each of the 4 objects has been recognized. The server will send the target to the client 2 only if one specific target appears more than 6 times in this window. In this situation, the classification is higher than 90% although the latency is increased to about 2 sec which includes data collection time (1.1 sec), recognition time (0.47 sec), transmission time, etc.

11.5 Discussion

In this section, we discuss the challenges and potential directions for future research.

First, the proposed approach is significantly influenced by the quality of the EEG

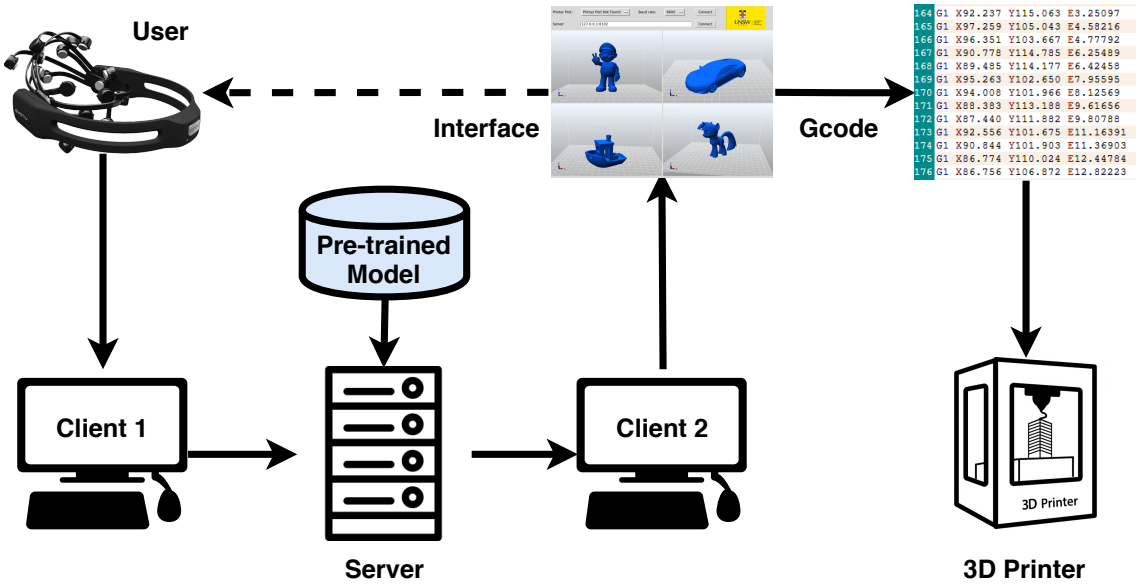


Figure 11.10: Online workflow of *Brain2Object*. The user's EEG signals are collected and send to the server through client 1. The server loads the pre-trained model to recognize the target object and send to both the interface for showing the user feedback and the 3D printer for printing. The solid line denotes signal transmission while the slash line denotes feedback.

data. The pre-trained model shows better performance on the clean and precise public dataset than on our local dataset. This suggests the need to develop novel classification methods that are robust to noisy and low resolution EEG signals. Another concern is the adaptability over different EEG acquisition equipment. Ideally, it is highly desirable that the model can consistently achieve accurate performance across a range of hardware platforms. However, the popular platforms (e.g., Emotiv, NeuroSky, OpenBCI) have different characteristics like sampling resolution, number of channels, positioning of channels on the scalp, etc. Thus, there is still innovation required to develop robust and adaptive brain signal classification algorithms.

Second, the object repository in this section is limited. An ideal instantiation of Brain2Object should recognize any object the user observed. However, in this chapter, the object repository only contains four items . The limitation of the repository scale is constrained by the learning algorithm, i.e., the ability of multi-class classification algorithms to discriminate between a large number of classes.. The

classification accuracy dramatically reduces with the increase of category numbers. In our pre-experiment which are not presented in this chapter for the space limitation, in the offline test, the proposed approach can achieve around 90% on binary classification using the Emotiv headset, however, the accuracy drops to nears 80% with three categories and about 70% with 4 objects. In our future work, we attempt to propose an algorithm to increase the multi-class classification performance.

Additionally, the ideal printing system is supposed to automatically detects the object which the subject 'thinking' (without visual stimulation) instead of 'observing' (with visual stimulation). however, the EEG SNR without visual stimulation is much lower than the SNR with visual stimulation. To enhance the SNR and help the subject to concentrate on the object, we adopt visual stimuli in our experiments. Therefore, in the local dataset and the online demonstration phase, the corresponding object images are shown on the monitor to remind the participants. However, the public dataset not only contains visual stimuli but also includes motor imagery, which is one possible reason why EEG-P is classified so accurately. Most of the existing public EEG dataset with visual stimulation focus on motor imagery (like the selected EEG-P) or evoked potentials [412]. In the latter instance, the visual images are flashed at a high frequency which results in pulsed EEG data, i.e. as objects flash by, short pulses of corresponding EEG data are generated. However, the model used in this chapter is based on stable EEG signal caused by steady stimuli. Hence, we select EEG-P instead of evoked potential-based dataset for the evaluation.

Furthermore, through the online demonstration experiment, we observed that the online performance is lower than the off-line analysis, which could be attributed to a number of reasons: 1) the user's mental state and fluctuations in emotions may affect the quality of the EEG signals. For instance, if the pre-trained model is tuned based on the EEG data which is collected when the user is relaxed, the classification performance may be affected while the user is excited in the online phase. 2) the conductive of the electrodes in headset is not exactly invariant during the off-line stage and online stage, which will have impact on the data quality; 3)

subtle variations (e.g., the position of each of the electrodes) in the way the EEG headset is mounted on the subject's head may also influent online decision making; 4) subjects often have difficulty in maintaining concentration during signal acquisition.

11.6 Conclusion

In this chapter, we propose an end-to-end 3D printing system based on the combination of multi-class CSP and graph embedded CNN. The performance of the proposed model is evaluated over two datasets in off-line and also demonstrated in the online environment. *Brain2Object* serves as a harbinger for exciting BCI applications which can help individuals with various tasks in their daily lives. The proposed system employs multi-class CSP to map the EEG data to a common space for the aim of maximizing the distance among various EEG patterns. The processed data are embedded by dynamic graph transformation and then fed into a designed convolutional neural network for automatically spatial feature learning. Extensive evaluations using a large-scale public dataset and a more relevant but limited local dataset showed that our scheme significantly outperforms a number of state-of-the-art approaches. The system latency is shown to be acceptable and a visualization of the signals is presented to offer additional perspectives into the performance. The online demonstration is presented to show the applicability of the proposed system.

Chapter 12

Multi-task Generative Adversarial Learning on Geometrical Shape Reconstruction from EEG Brain Signals

In this chapter, we present a novel and interesting application that reconstructing the geometrical shape shown in the user's mind through accurate EEG decoding.

Since the advent of neuroscience and BCI, numerous studies tried to recover the visual stimuli based on the informative human brain activities [413, 340]. The development of the decoding technologies of chaotic brain signals is supposed to reveal the mechanism of brain neurons and may implement some fantastic ambitions such as mind reading [9]. Most of the existing work focused on fMRI monitoring brain activities by detecting changes associated with blood flow in brain areas. However, fMRI-based image reconstruction faces several major challenges [414, 413]. The temporal resolution of fMRI is low constrained by the blood flow speed; the acquisition

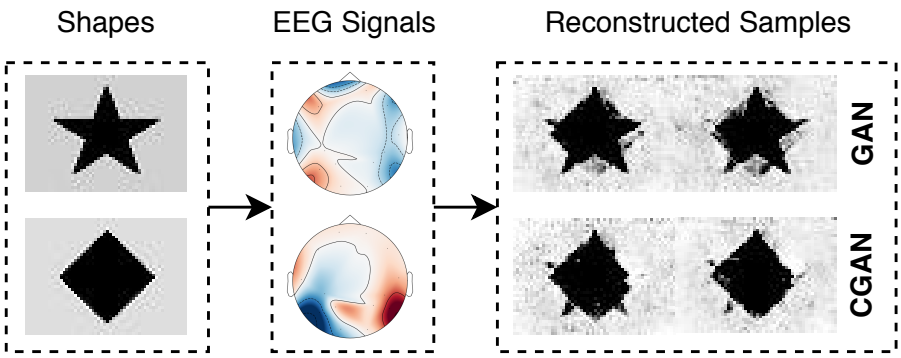


Figure 12.1: Generated samples based on EEG signals evoked by geometric shapes. It is observed that the samples synthesized by traditional methods (e.g., GAN and CGAN) are blur and lack of realistic details.

of fMRI requires a scanner which is expensive and hard to afford; the scanner is heavy and has poor portability [340].

Thus, EEG recently has drawn much attention as its high temporal resolution, low price, and high portability. EEG is a non-invasive signal measuring the voltage fluctuations generated by an electrical current within human neurons. Researchers have tried to exploit EEG signals to reconstruct visual stimuli [142, 141] through Generative Adversarial Networks (GANs). Nevertheless, the previous studies suffer from the low realism problem of the generated samples, which means that the model can not generate images with high realism based on the input brain signals. In other words, the current EEG-based synthesis methods can roughly present the visual stimuli but fail to contain necessary details. For example, as shown in Figure 12.1, the clear geometric shapes are present to the individual and reconstruct the shapes based on the collected EEG data. It is demonstrated that the geometric shapes generated by traditional GAN and CGAN are blurry and lack of realistic details.

Aiming at the aforementioned issues, in this chapter, we conduct experiments to measure the individual's EEG oscillation evoked by various geometrical shapes and propose a novel framework in order to precisely decode the EEG signals and synthesize the geometric shapes. Moreover, we employ a CNN to explore the latent representation form the raw EEG signals since CNN is much efficient than the RNN

with a similar EEG representation learning ability based on our empirical experiments. In addition, we adopted a multi-task discriminator with a task-specific classifier which assigns the geometric shape into the correct class for the aim of improving the quality of the recovered shapes. Furthermore, we propose a semantic alignment method involving the semantic information of the real shape to enhance the realism level of the reconstructed shape. The previous works are mainly paid attention to brain signal based images (e.g., bird and plane) reconstruction which contain too many attributes (e.g., color, shape, size, background, and semantic information), as a result, it is difficult to figure out which attribute the human brain is more sensitive to and which one contributes more to the object reconstruction. Thus, in this section, we focus on the EEG-based geometric shape reconstruction and attempt to illustrate that EEG signals are sensitive to geometries.

In detail, the contributions of this chapter are listed here:

- We present a novel deep generative model to recover the geometrical shape seen by human beings from the EEG signals. To our best knowledge, we are the first work investigating the brain signal based geometric shape reconstruction. The reproducible codes are publicly available here[1].
- We propose an effective semantic alignment method to harness the semantic information of the original geometric shape in order to force the approach to produce more realistic shapes.
- We conducted a local EEG dataset stimulated by various geometric shapes and evaluate the proposed approach over the collected dataset. The experimental results demonstrated that our model outperforms all the competitive state-of-the-art baselines.

[1]https://github.com/xiangzhang1015/EEG_Shape_Reconstruction

12.1 Related Work

Recent years' research in neuroscience and neuroimaging [415] indicated that human perception of visual stimuli can be decoded through some techniques in neuroimaging. To be specific, a few works gave evidence about decoding the brain signals to human activity by using the Functional Magnetic Resonance Imaging (fMRI) and EEG. There are some works use the fMRI signals to reconstruct the image which is seen by the individual and get an acceptable performance [414, 416]. The studies show the potential of fMRI-based image reconstruction in the brain signals decoding area, however, fMRI faces a number of crucial issues such as expensive acquisition equipment and low portability. Apart from the fMRI based method, there are a few EEG based methods in image reconstruction as EEG signals are less expensive [142, 141]. As a typical investigation, Brain2image [142] encoded the raw EEG signals into a latent space which contains the distinctive information, and then sent them to a Conditional Generative Adversarial Networks (CGAN) for image reconstruction. Palazzo et al. [141] applied a very similar algorithm framework.

Most of the visual object reconstruction methods are based on GAN and the variations. GANs [56], as the typical deep learning frameworks, was used widely in image generation. The standard GANs are composed of a generator network which generates images from the random sampled noise and a discriminator network which tried to distinguish the generated image correctly. Normally, original GANs had to suffer from the uncontrollable issue of the generation process. In order to retard it, the conditional GAN (CGAN) was proposed [319] which involves the conditional information (e.g., labels) in order to control the generating process. Auxiliary Classifier GAN (ACGAN)[417] improve the performance of GAN for image synthesis. ACGAN demonstrated that adding more structure to the GAN latent space along with a specialized cost function results in higher quality samples. A task-specific branch in the discriminator is empowered to enhance the discriminability.

Summary. Most brain signal based image reconstruction work is based on fMRI.

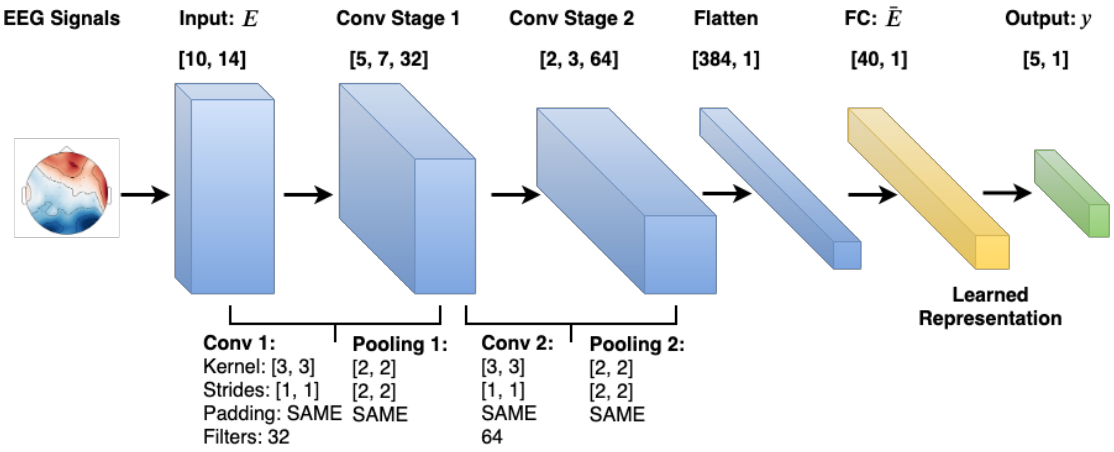


Figure 12.2: Demonstration of discriminative EEG representation learning. The last second layer $\bar{\boldsymbol{E}}$ with discriminative information is selected as learned representation. Each Conv stage contains a convolutional layer followed by a pooling layer. The basic hyper-parameters are presented.

Due to the drawbacks of fMRI (e.g., low time resolution, expensive, and low portability), we focus on EEG based geometric shape reconstruction. Compare to the typical EEG-based work like brain2image [142], we have several technical advantages: 1) we concentrate on the influence to the EEG signals brought by geometric attribute while [142] focus on images with a large number of attributes; 2) we adopt CNN instead of RNN to learn the latent EEG features which cost less training time with a similar accuracy; 3) we add an auxiliary task-specific classifier to improve the discriminability of the discriminator; 4) we propose a semantic alignment method to generate more realistic images.

12.2 Methodology

In this study, we aim to propose a method to convert the individual's mental geometry into physical shape. In particular, we first decode the non-invasive EEG signals into an implicit representation (Section 12.2.1) and then propose a modified GAN framework to generate the real shape which evoked the EEG signals (Section 12.2.2.

12.2.1 EEG Feature Learning

In the EEG feature learning, we adopt a CNN structure to capture the latent distinguishable features from the collected EEG signals. Some research had demonstrated that CNN is empowered to learn informative features from noisy EEG data[418, 120]. Suppose the EEG sample pairs can be denoted by $\boldsymbol{E} = \{(\boldsymbol{E}_h, \boldsymbol{y}_h), h = 0, 1, \cdots H\}$ where $\boldsymbol{E}_h \in \mathbb{R}^{M \times N}$ and $\boldsymbol{y}_h \in \mathbb{R}^5$ represent the EEG observations and the corresponding one-hot label. In this chapter, we focused on the decoding of five different visual-stimuli evoked imagination, thus the number of labels is five. The H denotes the number of EEG segments and M, N denotes the time- and spatial- resolution of each segment.

Figure 12.2 shows the workflow of the learning procedure of the discriminative representation. The visual-stimuli evoked EEG signals, reflecting the imagination in the user's mind, are feed into a CNN model with seven layers. The first convolutional layer contains 32 filters with the kernel size of $[3, 3]$ and stride of $[1, 1]$. The padding method is 'SAME' while the activation function is ReLU. The first pooling layer adopts max pooling and both the pooling size and strides are $[2, 2]$. The second convolutional and pooling layers are identical to the first layers, respectively, except the Conv 2 has 64 filters. The followed fully-connected layer has d nodes, which is regarded as the learned representation, denoted by $\bar{\boldsymbol{E}}$, and contains enough information to reconstruct the visual shape. The learning algorithm iterates for 1,000 epochs with Adam optimizer has a learning rate of $5e^{-4}$.

Compared to Brain2Image [142] which employed LSTM for feature learning, CNN is able to achieve a similar performance but spend much less training time. In particular, LSTM obtained the classification accuracy of 74% with 5,935s while CNN achieved 72% but with only $1,222$s.

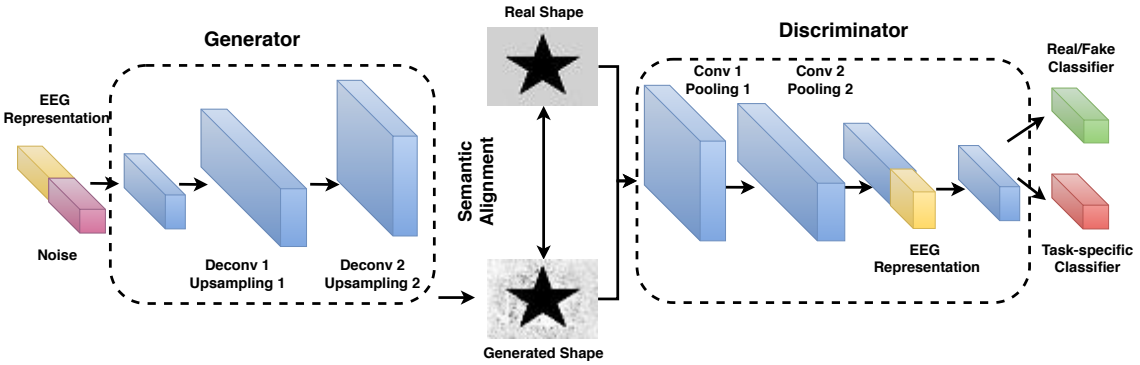


Figure 12.3: Workflow of the proposed visual stimuli reconstruction framework. We adopted a semantic classifier apart from the real/fake classifier in order to exploit the semantic information of the EEG samples. Moreover, a semantic regularization constraint is proposed to force the generated visual stimuli has similar semantic information with the real visual stimuli.

12.2.2 Multi-task Generation Model

Overview

In this part, we will describe the framework which is used to reconstruct the shapes that human seeing. As shown in Figure 12.3, the proposed geometrical shape generation framework contains two components: a generator and a discriminator.

The generator receives the learned discriminative EEG representation $\bar{\boldsymbol{E}} \in \mathbb{R}^d$ along with a random sampled Gaussian noise $\boldsymbol{z} \in \mathbb{R}^{d'}$ and produces generated shape. The EEG representation is evolved to guarantee the compelling of the generated shapes while the Gaussian noise is adopted to keep the diversity. On the other hand, the discriminator receives the real shape which evoked the brain signals (the imagination which presented in the human brain) and the generated fake shape. Inspired by ACGAN [417], we design a multi-task discriminator containing two branches while the first branch, like the standard GAN, aims at the recognition of the fake shapes and the second branch, an auxiliary task-specific classifier, attempts to classify what class the shape belongs to. The first branch is called real/fake classifier whilst the second one is called task-specific classifier. By adding the task-specific classifier, the

designed discriminator not only is able to distinguish whether the shape is real or not but also can recognize the category of the shape. As a consequence, the discriminator drives the distribution of the synthesized shapes not only approximate to the general distribution of the overall real shapes but also approximate to the distribution of a specific category. In addition, the learned EEG representation is also input to the discriminator, as proposed in [319], in order to make the discriminator under the same conditional situation with the generator.

Architecture

Next we report the details of the architecture. The generator receives the input vector which concatenates $\bar{\boldsymbol{E}}$ and $\boldsymbol{z}$, represented by $\boldsymbol{h_0} = \{\bar{\boldsymbol{E}} : \boldsymbol{z}\} \in \mathbb{R}^{d+d'}$, and attempts to map it to a meaningful shape. The generator is composed of a fully-connected and two deconvolutional layers each followed by a unsampling layer. The $\boldsymbol{h_0}$ is first fed into the fully-connected layer with $64(M+N)$ nodes:

$$\boldsymbol{h_1} = \sigma(\boldsymbol{w}\boldsymbol{h_0} + \boldsymbol{b}) \tag{12.1}$$

where $\boldsymbol{w}, \boldsymbol{b}$ and σ denote the weight, bias vector, and the sigmoid function, respectively. Then $\boldsymbol{h_1}$ is reshaped into $[M, N, 64]$ where 64 denotes the depth. To this end, $\boldsymbol{h_1}$ has a similar form, but deeper depth, with the raw EEG segment $\boldsymbol{E_h}$ which is supposed to contain enough information to reconstruct the user's imagination. Afterward, $\boldsymbol{h_1}$ is sent to the the first deconvolutional layer with 32 filters, kernel size $[5, 5]$, stride $[2, 2]$, and 'SAME' padding method. The upsampling operation is the invert operation of pooling and shares the same parameters with pooling layer. The second deconvolutional with one filter and upsampling layers. We choose the Tanh as activation function since it transforms the signals into the range $[-1, 1]$ which is the same range the real shape falls into. The synthesized shape $\boldsymbol{F}$ has shape $[4M, 4N]$. According to empirical experiments, we set the shape size 4 times of the EEG raw segment in both width and height in order to have a better generation quality. The real geometric shape $\boldsymbol{R}$ is in gray scale with format $[4M, 4N]$. All

the pixels are normalized into the range $[0, 1]$ by max-min normalization and then transformed to $[-1, 1]$ by:

$$\bar{\boldsymbol{R}} = 2\boldsymbol{R} - 1 \tag{12.2}$$

In the discriminator, as shown in Figure 12.3, both $\bar{\boldsymbol{R}}$ and $\boldsymbol{F}$ are fed into the discriminator which has almost the same structure and hyper-parameters with the discriminative representation learning model (Section 12.2.1). The input shape is flattened to a vector and then concatenates with the learned representation $\bar{\boldsymbol{E}}$. The fully-connected layer has 100 nodes. This designed discriminator has two branches corresponding two output layers. The output layer of the real/fake classifier only has one node which represents the fake probability. As for the task-specific classifier, the output layer has five nodes corresponding to five different geometrical shape categories.

Loss Function

We present the loss functions in the proposed framework. For the generator, since we add a task-specific classifier, the loss function contains two components where one component forces the discriminator cannot recognize the shape is generated while another component forces the discriminator to recognize which shape category the shape belongs to. Thus, the log-likelihood loss function for the generator can be defined as [417]:

$$\mathcal{L}_g = E[\log P(C = \boldsymbol{y}|X = \boldsymbol{F})] + E[\log(1 - D(G(\boldsymbol{y}, \bar{\boldsymbol{E}}, \boldsymbol{z})))] \tag{12.3}$$

in which,

$$\boldsymbol{F} = G(\boldsymbol{y}, \bar{\boldsymbol{E}}, \boldsymbol{z}) \tag{12.4}$$

describes the generator G, and

$$P(S|X), P(C|X) = D(X) \tag{12.5}$$

describes the real/fake classifier and task-specific classifier of the discriminator D, respectively. As for the discriminator, the loss function also contains two components

separately coming from the two classifiers. The discriminator is supposed to filter out which shape is generated, meanwhile, to assign the shape into the correct class. The log-likelihood loss function $\mathcal{L}_d$ for the discriminator is:

$$\mathcal{L}_d = E[\log P(S = \bar{\boldsymbol{R}}|X = \bar{\boldsymbol{R}})] + E[\log P(S = \boldsymbol{F}|X = \boldsymbol{F})] + E[\log P(C = \boldsymbol{y}|X = \bar{\boldsymbol{R}})] \tag{12.6}$$

In the above formula, the $\boldsymbol{y}$ represents the class label. The C, S denote the predicted class and and source, which are the classification results of the multi-task generator. X denotes the shape fed into the discriminator. The $P(S|X)$ denotes the probability distribution over the source S while the $P(C|X)$ denotes the probability distribution over the class label $\boldsymbol{y}$.

12.2.3 Semantic Alignment

To this end, the geometrical shape reconstruction model is able to generate a batch of samples which have enough diversity but still less discriminability. Furthermore, in order to increase the discriminability of the generated samples and make the samples more realistic, we propose a semantic alignment method to adopt the semantic information to make the synthesized shape more realistic and sharper. In particular, we add an additional constraint on the generator loss function aiming at reducing the distance between the real and the generated geometric shapes.

The semantic distance can be measured by S_r:

$$S_r = \frac{1}{\sqrt{\bar{N}}} \sqrt{\sum_{i=0}^{\bar{N}} \sum_{j=0}^{\bar{N}} (\bar{R}_{i,j} - F_{i,j})^2} \tag{12.7}$$

where $\bar{N}$ denotes the number of pixels in the geometric sample and $\bar{N} = 4M \times 4N$. The $\bar{R}_{i,j}$ and $F_{i,j}$ denote the pixels in the real and generated samples. In order to improve the performance of the generator, the S_r is considered as a regularization of the generator loss. Thus, we update the Equation 12.3 as:

$$\mathcal{L}_g = E[\log P(C = \boldsymbol{y}|X = \boldsymbol{F})] + E[\log(1 - D(G(\boldsymbol{y}, \bar{\boldsymbol{E}}, \boldsymbol{z})))] + \lambda S_r \tag{12.8}$$

where λ is a constant coefficient to adjust the weight of semantic regularization. If the alignment constraint too strong, the generated shapes may have less diversity. In this section, we set $\lambda = 0.01$ to make a trade-off between the diversity and discriminability of the generated samples.

During the training, both $\mathcal{L}_g$ and $\mathcal{L}_d$ are optimized by the Adam optimizer. The learning rate is set as 0.0002 with the exponential decay rate of 0.5. In each epoch, the $\mathcal{L}_g$ and $\mathcal{L}_d$ are separately trained in turn. The proposed framework converges after 120 epochs and trend to overfitting after 160 epochs, thus, we adopt the early stopping strategy by breaking the iteration at the 150-th epoch.

12.3 Experiments

In this section, we will describe the experiments and the performance analysis containing qualitative and quantitative aspects in detail. The qualitative comparison will conduct the analysis in the quality of the generated shapes, and the quantitative comparison will be based on the inception score [317] and inception accuracy.

12.3.1 EEG Signal Acquisition

We conducted a local experiment with 8 healthy participants (6 males and 2 females) aged 25 $\pm$ 3, which is approved by UNSW ethic abroad (HC190315). During the experiments, the participant is required to sit in an armed comfortable chair in front of a computer monitor. We select five representative and widely-seen geometrical shapes (circle, star, triangle, rhombus, and rectangle) to present to the subject. The whole experiments contain two sessions and each session has five trials. In each trial, the five geometrical shapes are presented in random order and each shape lasts for five seconds. There are five seconds relax period among two adjacent shapes. The relaxing time among trials and sessions are 10s and 30s, respectively. The EEG

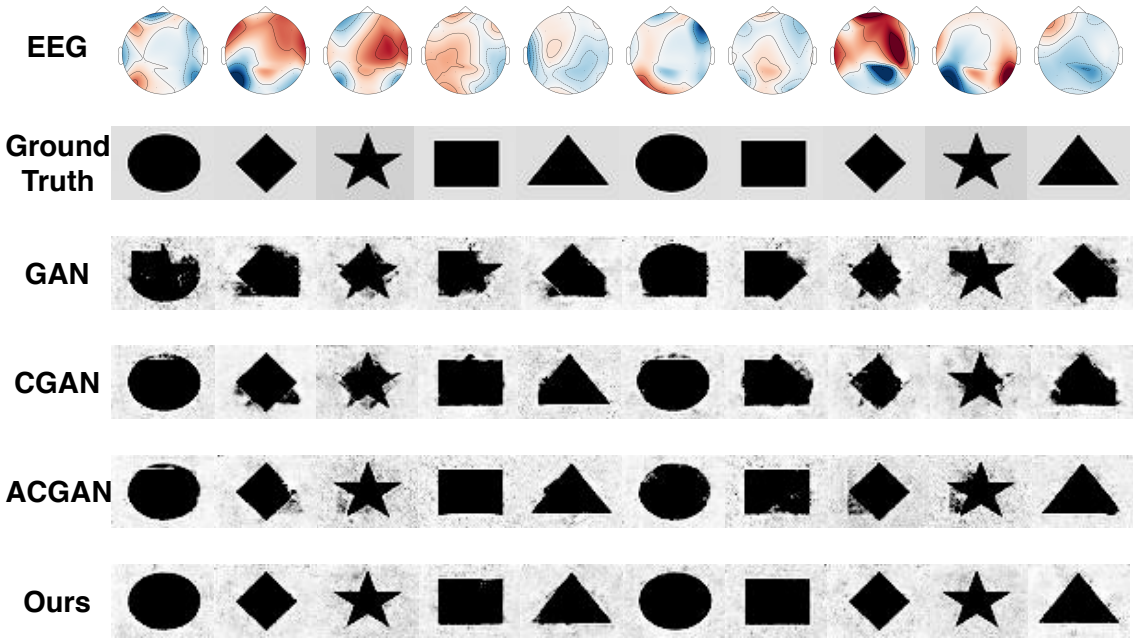


Figure 12.4: Demonstration of the qualitative comparison. Our model can reconstruct all the shapes correctly which have the highest similarity with the ground truth.

signals are collected through a portable Emotiv EPOC+ headset with 14 electrodes and the sampling frequency is set as 128 Hz. Each EEG segment contains ten continuous instances with 50% overlapping. The dataset is randomly divided into a training set (80% proportion) and testing set (20% proportion).

Based on the collected EEG data, we report the hyper-parameters settings. The single EEG segment $\boldsymbol{E}$ ($M = 10$ and $N = 14$) is compressed into a latent discriminative representation $\bar{\boldsymbol{E}}$ with dimension $d = 40$. In the generator, the stochastically sampled noise $\boldsymbol{z}$ has dimension $d' = 20$. The coefficient of semantic regularization λ is set as 0.001.

12.3.2 Qualitative Comparison

In this section, we compare the quality of the generated shapes among the proposed method and the state-of-the-art models. As shown in Figure 12.4, we choose the

most widely used generative models including GAN, CGAN and ACGAN as the baseline.

GAN achieve a promising result in many areas, especially in shape field [56]. On the top of basic GAN, CGAN [319] is proposed to add the conditional information as a constraint, which is adopted in [142]. Furthermore, ACGAN attempt to deeply exploit the informative sample labels to enhance the discriminability of D [417]. Our work, compared to ACGAN, proposed a semantic alignment method to constrain the distance among the synthesized shapes and the visual geometrical shapes in order to further emphasize the reality.

It's easy to find that, from Figure 12.4, our approach have the best shape quality. To be specific, the samples which generated by GAN are lack of clear edge, which is a typical mode collapse problem, meanwhile, it's not hard to figure out that most of the synthesized shapes have miscellaneous features. The CGAN has a better performance than normal GAN as the shapes have a higher integrity. However, we still can find that some shapes generated by CGAN have combined features such as a star have the feature from rhombus. The ACGAN have the best result among the baseline models, which it can learn most of the shapes' feature and correctly reconstruct the shapes with a trivial acceptable flaw. Our model can reconstruct all the shapes correctly which have the highest similarity with the ground truth.

12.3.3 Quantitative Comparison

The qualitative comparison is relatively easy as the shape quality is the assessment criteria. The quantitative analyses are hard to conduct as the comparison between reconstructed and real shape is not obvious and clearly defined. The common way we used to do that is using the inception score and the inception accuracy [142]. We build an inception network used the generated shapes as input in order to calculate the inception score which measures how realistic the generated shapes are. In detail, we generate 1,000 images for each geometric shape and calculate the overall inception

Table 12.1: The quantitative comparison of inception score and inception accuracy

Models	**GAN**	**C-GAN [142]**	**ACGAN**	**Ours**
Inception Score	1.931	1.986	2.061	**2.178**
Inception Accuracy	0.43	0.67	0.79	**0.83**

score. Moreover, our work is supposed to convert the specific EEG signals into the corresponding geometrical shape belonging to the specific label. Thus, we adopt the performance of the task-specific classifier when the input data is $\boldsymbol{F}$ as inception accuracy in order to measure how precise can our model generates shapes.

We conduct the quantitative analyses for the baselines and our proposed model. The results are presented in Table 12.1, in which, it is easy to observe that our model achieves the highest inception score and inception accuracy of 2.178 and 0.83, respectively. The inception score is not good as the public datasets like CIFAR-10 and the most possible reason is that our generated shapes are conditioned by EEG signals which is chaotic and has a low signal-to-noise ratio. Even though, the proposed approach outperforms all the competitive baselines.

12.4 Discussion

Next, we discuss the opening challenges and potential future work.

First of all, one major issue faced by brain signal based reconstruction is the recovery of unseen geometrical shapes. For instance, one future scope is to decode the EEG signals evoked by star while the star never is trained in the reconstruction model. One possible solution is train a common generative model by a large classes of basic geometrical shapes (e.g., circle, ellipse, straight line, triangle, rectangle, and rhombus) in order to learn the latent features of each different shape and then approximate the unseen shape (e.g., star) based on the learned features.

Second, we only focused on the simple geometrical shapes in this section, as a pre-

liminary study, however, the real world application demands more complex shapes like a bow. One of our future works is to consider more complicated geometric shapes in the experiments. In addition, another potential research direction is to increase the number of geometrical categories since this section only evaluated five basic classes.

Last but not least, more participants should be involved in the experiments in order to provide a general generative model which is robust for different individuals. The influence of inter-subject divergence should be taken into account in future research.

12.5 Conclusion

In this chapter, we propose a novel approach to reconstruct the geometrical shape based on the brain signals. We first develop a framework learning the latent discriminative representation of the raw EEG signals, and then, based on the learned representation, we propose an adversarial reconstruction framework to recover the geometric shapes which are visualizing by the human. In particular, we propose a semantic alignment method to improve the realism of the generated samples and force the framework to generate more realistic geometric shapes. The proposed approach is evaluated over a local dataset and the experimental results show that our model outperforms the competitive state-of-the-art methods both quantitatively and qualitatively.

Part VI

Conclusion

12. Geometrical Shape Reconstruction from EEG Brain Signals

This part contains work published in:

[1] **Zhang, X.**, Yao L., Wang X., Monaghan J., Mcalpine D., and Zhang Y. (2018). A Survey on Deep Learning based Brain Computer Interface: Recent Advances and New Frontiers. *Journal of Neural Engineering.* (Under Review, SJR Q1)

Chapter 13

Conclusion and Future Works

13.1 Conclusion

In this dissertation, we first present the overview of recent advanced studies which adopt deep representation learning to deal with BCI challenges. In particular, we provide a classification of deep learning models based on the functions in BCI procedures. Moreover, we provide a systematical summary of the state-of-the-art methods in each brain signal paradigms. Then, we discuss the current widely used real-world BCI applications and gather a set of open source BCI benchmark datasets.

Additionally, we propose the hybrid deep representation learning frameworks to automatically extract distinctive features from the raw EEG signals, addressing the drawbacks of brain signals (such as low SNR, vulnerable to environmental and artificial factors, and time-consuming feature engineering). Focusing on the multi-class EEG classification in the context of cross-subject, the proposed approach aims at discovering the patterns in the discrepancy among various EEG classes with robustness over the difference between various subjects.

In order to solve the challenge of domain knowledge-dependency, we develop a robust representation learning framework with better generalization. The proposed framework is enabled to adaptively learn the representative features from various scenarios. In detail, the proposed reinforced selective attention framework combines the advantages of reinforcement learning, deep learning, and attention mechanism. We explore the effectiveness of the modified WAS-LSTM and CNN classifier in different sensory signals and various EEG application scenarios. The experimental results demonstrate that our approach not only outperforms several state-of-the-art baselines by a large margin but also shows low latency and high resilience in coping with multiple EEG signal channels and incomplete EEG signals. Our approach shows excellent discriminative ability over all the tested situations.

Furthermore, to efficiently exploit the latent information in EEG data, we develop an effective and robust semi-supervised latent representation framework by combining the benefits of a modified VAE model (which is called VAE++) and a semi-supervised generative adversarial networks. The proposed framework is empowered to discover the distribution of unlabeled signals to enhance the convergence of the training of the labeled samples. Then, since the complex framework contains a large number of hyper-parameters, we adopt an orthogonal array based method, OATM, to search the optimal settings of hyper-parameter in order to reduce tuning time and computational cost.

At last, we developed several preliminary BCI applications based on the proposed data-efficient deep representation learning frameworks. To begin with, we proposed a biometric EEG-based identification approach called MindID and argue that its inherent resilience against attacks makes it an attractive approach compared to traditional biometric identification methods. Then, we present a hybrid deep learning model to decode the raw EEG signal for the aim of converting the user's thoughts to texts. The model employs the RNN and CNN to learn the temporal and spatial dependency features from the input raw EEG data and then stack them together, and then fed into an autoencoder to eliminate the artifacts and employs the XG-

Boost classifier for the intent recognition. In addition, we develop an end-to-end Brain2Object system serves as a harbinger for exciting BCI applications which can help individuals with various tasks in their daily lives. This system collects the user's brain waves, decodes them into recognizable object, and print a replica of the recognized object through a 3D printer. Finally, we develop an adversarial reconstruction framework to recover the geometric shapes which are visualizing by the human. In particular, we propose a semantic alignment method to improve the realism of the generated samples and force the framework to generate more realistic geometric shapes.

According to our research experience, we would like to share some insights in the field of deep learning-based BCI. BCI has a bright future both in academia and industrial. By bridging human mind with the outer world, BCI could be widely used in a wide range of real-world scenarios (e.g., healthcare, public security, and entertainment) for not only disabilities but also health individuals. Second, deep learning technicals have already and will constantly play an crucial role in the brain signal decoding which is the core component of BCI. More detailed opening challenges will be discussed in Section 13.2.

13.2 Future Works

Although deep learning has increased the performance of BCI systems, technical and usability challenges remains. The technical challenges concern the classification ability in complex BCI scenarios; and the usability challenges refer to limitations in large scale real-world deployment. In this section, we introduce these challenges and point out the possible solutions.

13.2.1 General Framework

Until now, we have introduce several types of BCI signals (e.g., spontaneous EEG, ERP, fMRI) and deep learning models that have been applied for each type. One promising research direction for deep learning based BCI is to develop a general framework that can handle various BCI signals regardless of the number of channels used for signal collection, the sample dimensions (e.g., 1-D or 2-D sample), and stimulation types (e.g., visual or audio stimuli), etc. The general framework would requires two key capabilities: the attention mechanism and the ability to capture latent feature. The former guarantees the framework can focus on the most valuable parts of input signals and the latter enables the framework to capture the distinctive and informative features.

The attention mechanism can be implemented based on attention scores or by various machine learning algorithms such as reinforcement learning. The attention scores can be inferred from the input data and work as a weight to help the framework to pay attention to the parts with high attention scores. Reinforcement learning has been shown to be able to find the most valuable part through a policy search [4]. CNN is the most suitable structure for capturing features in various levels and ranges. In the future, CNN could be used as a fundamental feature learning tool and be integrated with suitable attention mechanisms to form a general classification framework.

13.2.2 Person-independent Classification

Until now, most BCI classification tasks focus on person-dependent scenarios, where the training and the testing sets come from the same person. The future direction is to realize person-independent classification so that the testing data will never appear in the training set. High-performance person-independent classification is necessary for the wide application of BCI Systems in the real-world.

One possible solution to achieving this goal is to build a personalized model with transfer learning. A personalized effective model can adopt a transductive parameter transfer approach to construct individual classifiers and to learn a regression function that maps the relationship between data distribution and classifier parameters [419]. Another potential solution is mining the subject-independent components from the input data. The input data can be decomposed into two parts: a subject-dependent component, which depends on the subject and a subject-independent component, which is common for all subjects. A hybrid multi-task model can work on two tasks simultaneously, one focusing on person identification and the other on class recognition. A well-trained and converged model ought to extract the subject-independent features in a class recognition task.

13.2.3 Semi-supervised and Unsupervised Classification

The performance of deep learning models highly depends on the size of training data, which requires expensive and time-consuming manual labeling to collect abundant class labels in a wide range of scenarios such as sleeping EEG. While supervised learning requires both observations and labels for the training, unsupervised learning requires no labels and semi-supervised learning only requires partial labels [111]. Therefore, they are more suitable for problems with little ground truth data available.

Zhang et al. proposed an Adversarial Variational Embedding (AVAE) framework that combines a VAE++ model (as a high-quality generative model) and semi-supervised GAN (as a posterior distribution learner) [6] for robust and effective semi-supervised learning. Jia et al. [111] proposed a semi-supervised framework by leveraging label information in feature extraction and integrating unlabeled information to regularize the supervised training.

Two methods may enhance unsupervised learning: one is to employ crowd-sourcing to label the unlabeled observations; the other is to leverage unsupervised domain

adaption learning to align the distribution of source BCI signals and the distribution of target signals with a linear transformation.

13.2.4 Hardware Portability

Poor portability of hardware has prevented the wide application of BCI systems in the real world. In most scenarios, users would like to use small, comfortable, or even wearable BCI hardware to collect brain signals and to control appliances and assistant robots.

Currently, there are three types of EEG collection equipment: the unportable, the portable headset, and ear-EEG sensors. The unportable equipment (e.g., Neuroscan, Biosemi) has high sampling frequency, channel numbers, and signal quality but is expensive. It is suitable for physical examination in hospital. The portable headsets (e.g., Neurosky, Emotiv EPOC) have $1 \sim 14$ channels and $128 \sim 256$ sampling rate but may cause discomfort for users after a long-time use. The ear-EEG sensors, which are attached to the outer ear, have gained increasing attention recently but remain mostly at the laboratory stage [420]. The ear-EEG platform comprises a set of electrodes placed inside each ear canal, together with additional electrodes in the concha of each ear [421]. The EEGrids, to the best of our knowledge, is the only commercial ear-EEG. It has multi-channel sensor arrays placed around the ear using an adhesive [1] and is even more expensive. An promising future direction is to improve the usability by developing a cheaper (e.g., lower than 200$) and more comfortable (e.g., can last longer than 3 hours without feeling uncomfortable) wireless ear-EEG equipment.

[1] http://ceegrid.com/home/concept/

Bibliography

[25] F. Lotte, L. Bougrain, A. Cichocki, M. Clerc, M. Congedo, A. Rakotomamonjy, and F. Yger, "A review of classification algorithms for eeg-based brain–computer interfaces: a 10 year update," *Journal of neural engineering*, vol. 15, no. 3, p. 031005, 2018.

[26] S. N. Abdulkader, A. Atia, and M.-S. M. Mostafa, "Brain computer interfacing: Applications and challenges," *Egyptian Informatics Journal*, vol. 16, no. 2, pp. 213–230, 2015.

[27] T. Ball, M. Kern, I. Mutschler, A. Aertsen, and A. Schulze-Bonhage, "Signal quality of simultaneously recorded invasive and non-invasive eeg," *Neuroimage*, vol. 46, no. 3, pp. 708–716, 2009.

[28] A. Bashashati, M. Fatourechi, R. K. Ward, and G. E. Birch, "A survey of signal processing algorithms in brain–computer interfaces based on electrical brain signals," *Journal of Neural engineering*, vol. 4, no. 2, p. R32, 2007.

[29] W. Samek, K.-R. Müller, M. Kawanabe, and C. Vidaurre, "Brain-computer interfacing in discriminative and stationary subspaces," in *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*, pp. 2873–2876, IEEE, 2012.

[30] H. Cecotti, M. P. Eckstein, and B. Giesbrecht, "Single-trial classification of event-related potentials in rapid serial visual presentation tasks using supervised spatial filtering," *IEEE transactions on neural networks and learning systems*, vol. 25, no. 11, pp. 2030–2042, 2014.

[31] M. Mahmud, M. S. Kaiser, A. Hussain, and S. Vassanelli, "Applications of deep learning and reinforcement learning to biological data," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 6, pp. 2063–2079, 2018.

[32] D. Wen, Z. Wei, Y. Zhou, G. Li, X. Zhang, and W. Han, "Deep learning methods to process fmri data and their application in the diagnosis of cognitive

impairment: A brief overview and our opinion," *Frontiers in neuroinformatics*, vol. 12, p. 23, 2018.

[33] M. Fatourechi, A. Bashashati, R. K. Ward, and G. E. Birch, "Emg and eog artifacts in brain computer interface systems: A survey," *Clinical neurophysiology*, vol. 118, no. 3, pp. 480–494, 2007.

[34] J. Liu, Y. Pan, M. Li, Z. Chen, L. Tang, C. Lu, and J. Wang, "Applications of deep learning to mri images: a survey," *Big Data Mining and Analytics*, vol. 1, no. 1, pp. 1–18, 2018.

[35] F. Lotte, M. Congedo, A. Lécuyer, F. Lamarche, and B. Arnaldi, "A review of classification algorithms for eeg-based brain–computer interfaces," *Journal of neural engineering*, vol. 4, no. 2, p. R1, 2007.

[36] S. Mason, A. Bashashati, M. Fatourechi, K. Navarro, and G. Birch, "A comprehensive survey of brain interface technology designs," *Annals of biomedical engineering*, vol. 35, no. 2, pp. 137–169, 2007.

[37] X. Wang, G. Gong, N. Li, and Y. Ma, "A survey of the bci and its application prospect," in *Theory, Methodology, Tools and Applications for Modeling and Simulation of Complex Systems*, pp. 102–111, Springer, 2016.

[38] H. Cecotti and A. J. Ries, "Best practice for single-trial detection of event-related potentials: Application to brain-computer interfaces," *International Journal of Psychophysiology*, vol. 111, pp. 156–169, 2017.

[39] N. Naseer and K.-S. Hong, "fnirs-based brain-computer interfaces: a review," *Frontiers in human neuroscience*, vol. 9, p. 3, 2015.

[40] R. Yannick, B. Hubert, A. Isabela, G. Alexandre, F. Jocelyn, *et al.*, "Deep learning-based electroencephalography analysis: a systematic review," *arXiv preprint arXiv:1901.05498*, 2019.

[41] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[42] L. Deng, "A tutorial survey of architectures, algorithms, and applications for deep learning," *APSIPA Transactions on Signal and Information Processing*, vol. 3, 2014.

[43] L. Deng, D. Yu, *et al.*, "Deep learning: methods and applications," *Foundations and Trends® in Signal Processing*, vol. 7, no. 3–4, pp. 197–387, 2014.

[44] G. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. Van Der Laak, B. Van Ginneken, and C. I. Sánchez, "A survey on deep learning in medical image analysis," *Medical image analysis*, vol. 42, pp. 60–88, 2017.

[45] S. R. Soekadar, N. Birbaumer, M. W. Slutzky, and L. G. Cohen, "Brain–machine interfaces in neurorehabilitation of stroke," *Neurobiology of disease*, vol. 83, pp. 172–179, 2015.

[46] S. Vieira, W. H. Pinaya, and A. Mechelli, "Using deep learning to investigate the neuroimaging correlates of psychiatric and neurological disorders: Methods and applications," *Neuroscience & Biobehavioral Reviews*, vol. 74, pp. 58–75, 2017.

[47] F. Movahedi, J. L. Coyle, and E. Sejdić, "Deep belief networks for electroencephalography: A review of recent contributions and future outlooks," *IEEE journal of biomedical and health informatics*, vol. 22, no. 3, pp. 642–652, 2018.

[48] M. Ahn and S. C. Jun, "Performance variation in motor imagery brain–computer interface: a brief review," *Journal of neuroscience methods*, vol. 243, pp. 103–110, 2015.

[49] S. Ruiz, K. Buyukturkoglu, M. Rana, N. Birbaumer, and R. Sitaram, "Real-time fmri brain computer interfaces: self-regulation of single brain regions to networks," *Biological psychology*, vol. 95, pp. 4–20, 2014.

[50] A. Haider and R. Fazel-Rezai, "Application of p300 event-related potential in brain-computer interface," in *Event-Related Potentials and Evoked Potentials*, InTech, 2017.

[51] O. Tsinalis, P. M. Matthews, Y. Guo, and S. Zafeiriou, "Automatic sleep stage scoring with single-channel eeg using convolutional neural networks," *arXiv preprint arXiv:1610.01683*, 2016.

[52] Q. Gui, M. Ruiz-blondet, S. Laszlo, and Z. Jin, "A survey on brain biometrics," *ACM Computing Surveys*, vol. 51, no. 112, 2019.

[53] P. O. Glauner, "Comparison of training methods for deep neural networks," *arXiv preprint arXiv:1504.06825*, 2015.

[54] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[55] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[56] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, pp. 2672–2680, 2014.

[57] G. St-Yves and T. Naselaris, "Generative adversarial networks conditioned on brain activity reconstruct seen images," in *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 1054–1061, IEEE, 2018.

[58] S. Chambon, M. N. Galtier, P. J. Arnal, G. Wainrib, and A. Gramfort, "A deep learning architecture for temporal sleep stage classification using multivariate and multimodal time series," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 2018.

[59] J. Zhang, Y. Wu, J. Bai, and F. Chen, "Automatic sleep stage classification based on sparse deep belief net and combination of multiple classifiers," *Transactions of the Institute of Measurement and Control*, vol. 38, no. 4, pp. 435–451, 2016.

[60] A. Sors, S. Bonnet, S. Mirek, L. Vercueil, and J.-F. Payen, "A convolutional neural network for sleep stage scoring from raw single-channel eeg," *Biomedical Signal Processing and Control*, vol. 42, pp. 107–114, 2018.

[61] A. Vilamala, K. H. Madsen, and L. K. Hansen, "Neural networks for interpretable analysis of eeg sleep stage scoring," in *INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING 2017*, 2017.

[62] M. Shahin, B. Ahmed, S. T.-B. Hamida, F. L. Mulaffer, M. Glos, and T. Penzel, "Deep learning and insomnia: Assisting clinicians with their diagnosis," *IEEE journal of biomedical and health informatics*, vol. 21, no. 6, pp. 1546–1553, 2017.

[63] I. Fernández-Varela, D. Athanasakis, S. Parsons, E. Hernández-Pereira, and V. Moret-Bonillo, "Sleep staging with deep learning: a convolutional model," in *Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2018)*.

[64] S. Biswal, J. Kulas, H. Sun, B. Goparaju, M. B. Westover, M. T. Bianchi, and J. Sun, "Sleepnet: automated sleep staging system via deep learning," *arXiv preprint arXiv:1707.08262*, 2017.

[65] K. M. Tsiouris, V. C. Pezoulas, M. Zervakis, S. Konitsiotis, D. D. Koutsouris, and D. I. Fotiadis, "A long short-term memory deep learning network for the prediction of epileptic seizures using eeg signals," *Computers in biology and medicine*, vol. 99, pp. 24–37, 2018.

[66] D. Tan, R. Zhao, J. Sun, and W. Qin, "Sleep spindle detection using deep learning: a validation study based on crowdsourcing," in *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*, pp. 2828–2831, IEEE, 2015.

[67] M. Manzano, A. Guillén, I. Rojas, and L. J. Herrera, "Combination of eeg data time and frequency representations in deep networks for sleep stage classification," in *International Conference on Intelligent Computing*, pp. 219–229, Springer, 2017.

[68] M. Manzano, A. Guillén, I. Rojas, and L. J. Herrera, "Deep learning using eeg data in time and frequency domains for sleep stage classification," in *International Work-Conference on Artificial Neural Networks*, pp. 132–141, Springer, 2017.

[69] A. Supratak, H. Dong, C. Wu, and Y. Guo, "Deepsleepnet: a model for automatic sleep stage scoring based on raw single-channel eeg," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 25, no. 11, pp. 1998–2008, 2017.

[70] H. Dong, A. Supratak, W. Pan, C. Wu, P. M. Matthews, and Y. Guo, "Mixed neural network approach for temporal sleep stage classification," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 26, no. 2, pp. 324–333, 2018.

[71] L. Duan, M. Bao, J. Miao, Y. Xu, and J. Chen, "Classification based on multilayer extreme learning machine for motor imagery task from eeg signals," *Procedia Computer Science*, vol. 88, pp. 176–184, 2016.

[72] K. G. Hartmann, R. T. Schirrmeister, and T. Ball, "Hierarchical internal representation of spectral features in deep convolutional networks trained for eeg decoding," in *Brain-Computer Interface (BCI), 2018 6th International Conference on*, pp. 1–6, IEEE, 2018.

[73] E. Nurse, B. S. Mashford, A. J. Yepes, I. Kiral-Kornek, S. Harrer, and D. R. Freestone, "Decoding eeg and lfp signals using deep learning: heading truenorth," in *Proceedings of the ACM International Conference on Computing Frontiers*, pp. 259–266, ACM, 2016.

[74] X. Zhang, L. Yao, K. Chen, X. Wang, Q. Sheng, and T. Gu, "Deepkey: An eeg and gait based dual-authentication system," *arXiv preprint arXiv:1706.01606*, 2017.

[75] H. Yang, S. Sakhavi, K. K. Ang, and C. Guan, "On the use of convolutional neural networks and augmented csp features for multi-class motor imagery

of eeg signals classification," in *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*, pp. 2620–2623, IEEE, 2015.

[76] L. Jingwei, C. Yin, and Z. Weidong, "Deep learning eeg response representation for brain computer interface," in *Control Conference (CCC), 2015 34th Chinese*, pp. 3518–3523, IEEE, 2015.

[77] T. Uktveris and V. Jusas, "Application of convolutional neural networks to four-class motor imagery classification problem," *Information Technology And Control*, vol. 46, no. 2, pp. 260–273, 2017.

[78] H. K. Lee and Y.-S. Choi, "A convolution neural networks scheme for classification of motor imagery eeg based on wavelet time-frequecy image," in *Information Networking (ICOIN), 2018 International Conference on*, pp. 906–909, IEEE, 2018.

[79] Z. Tang, C. Li, and S. Sun, "Single-trial eeg classification of motor imagery using deep convolutional neural networks," *Optik-International Journal for Light and Electron Optics*, vol. 130, pp. 11–18, 2017.

[80] Q. Wang, Y. Hu, and H. Chen, "Multi-channel eeg classification based on fast convolutional feature extraction," in *International Symposium on Neural Networks*, pp. 533–540, Springer, 2017.

[81] I. Sturm, S. Lapuschkin, W. Samek, and K.-R. Müller, "Interpretable deep neural networks for single-trial eeg classification," *Journal of neuroscience methods*, vol. 274, pp. 141–145, 2016.

[82] N. Lu, T. Li, X. Ren, and H. Miao, "A deep learning scheme for motor imagery classification based on restricted boltzmann machines," *IEEE transactions on neural systems and rehabilitation engineering*, vol. 25, no. 6, pp. 566–576, 2017.

[83] S. Kumar, A. Sharma, K. Mamun, and T. Tsunoda, "A deep learning approach for motor imagery eeg signal classification," in *Computer Science and Engineering (APWC on CSE), 2016 3rd Asia-Pacific World Congress on*, pp. 34–39, IEEE, 2016.

[84] Y. Ren and Y. Wu, "Convolutional deep belief networks for feature extraction of eeg signal," in *Neural Networks (IJCNN), 2014 International Joint Conference on*, pp. 2850–2853, IEEE, 2014.

[85] J. Li and A. Cichocki, "Deep learning of multifractal attributes from motor imagery induced eeg," in *International Conference on Neural Information Processing*, pp. 503–510, Springer, 2014.

[86] S. Redkar, "Using deep learning for human computer interface via electroencephalography," *IAES International Journal of Robotics and Automation*, vol. 4, no. 4, 2015.

[87] E. S. Nurse, P. J. Karoly, D. B. Grayden, and D. R. Freestone, "A generalizable brain-computer interface (bci) using machine learning for feature discovery," *PloS one*, vol. 10, no. 6, p. e0131328, 2015.

[88] X. Zhang, L. Yao, S. Zhang, S. Kanhere, M. Sheng, and Y. Liu, "Internet of things meets brain-computer interface: A unified deep learning framework for enabling human-thing cognitive interactivity," *IEEE Internet of Things Journal*, 2018.

[89] M. Dai, D. Zheng, R. Na, S. Wang, and S. Zhang, "Eeg classification of motor imagery using a novel deep learning framework," *Sensors*, vol. 19, no. 3, p. 551, 2019.

[90] L. Fraiwan and K. Lweesy, "Neonatal sleep state identification using deep learning autoencoders," in *Signal Processing & its Applications (CSPA), 2017 IEEE 13th International Colloquium on*, pp. 228–231, IEEE, 2017.

[91] Y. R. Tabar and U. Halici, "A novel deep learning approach for classification of eeg motor imagery signals," *Journal of neural engineering*, vol. 14, no. 1, p. 016003, 2016.

[92] C. Tan, F. Sun, W. Zhang, J. Chen, and C. Liu, "Multimodal classification with deep convolutional-recurrent neural networks for electroencephalography," in *International Conference on Neural Information Processing*, pp. 767–776, Springer, 2017.

[93] A. J. Yepes, J. Tang, and B. S. Mashford, "Improving classification accuracy of feedforward neural networks for spiking neuromorphic chips," *arXiv preprint arXiv:1705.07755*, 2017.

[94] A. Frydenlund and F. Rudzicz, "Emotional affect estimation using video and eeg data in deep neural networks," in *Canadian Conference on Artificial Intelligence*, pp. 273–280, Springer, 2015.

[95] J. Li, Z. Zhang, and H. He, "Implementation of eeg emotion recognition system based on hierarchical convolutional neural networks," in *International Conference on Brain Inspired Cognitive Systems*, pp. 22–33, Springer, 2016.

[96] W. Liu, H. Jiang, and Y. Lu, "Analyze eeg signals with convolutional neural network based on power spectrum feature selection," *Proceedings of Science*, 2017.

[97] M.-P. Hosseini, D. Pompili, K. Elisevich, and H. Soltanian-Zadeh, "Optimized deep learning for eeg big data and seizure prediction bci via internet of things," *IEEE Transactions on Big Data*, vol. 3, no. 4, pp. 392–404, 2017.

[98] F. Wang, S.-h. Zhong, J. Peng, J. Jiang, and Y. Liu, "Data augmentation for eeg-based emotion recognition with deep convolutional neural networks," in *International Conference on Multimedia Modeling*, pp. 82–93, Springer, 2018.

[99] J. Li, Z. Zhang, and H. He, "Hierarchical convolutional neural networks for eeg-based emotion recognition," *Cognitive Computation*, pp. 1–13, 2017.

[100] S. S. Talathi, "Deep recurrent neural networks for seizure detection and early seizure detection systems," *arXiv preprint arXiv:1706.03283*, 2017.

[101] T. Zhang, W. Zheng, Z. Cui, Y. Zong, and Y. Li, "Spatial-temporal recurrent neural network for emotion recognition," *IEEE transactions on cybernetics*, no. 99, pp. 1–9, 2018.

[102] X. Li, P. Zhang, D. Song, G. Yu, Y. Hou, and B. Hu, "Eeg based emotion identification using unsupervised deep feature learning," 2015.

[103] Y. Gao, H. J. Lee, and R. M. Mehmood, "Deep learninig of eeg signals for emotion recognition," in *Multimedia & Expo Workshops (ICMEW), 2015 IEEE International Conference on*, pp. 1–5, IEEE, 2015.

[104] K. Li, X. Li, Y. Zhang, and A. Zhang, "Affective state recognition from eeg with deep belief networks," in *2013 IEEE International Conference on Bioinformatics and Biomedicine*, pp. 305–310, IEEE, 2013.

[105] H. Xu and K. N. Plataniotis, "Affective states classification using eeg and semi-supervised deep learning approaches," in *Multimedia Signal Processing (MMSP), 2016 IEEE 18th International Workshop on*, pp. 1–6, IEEE, 2016.

[106] H. Xu and K. N. Plataniotis, "Eeg-based affect states classification using deep belief networks," in *Digital Media Industry & Academic Forum (DMIAF)*, pp. 148–153, IEEE, 2016.

[107] Y. Zhao and L. He, "Deep learning in the eeg diagnosis of alzheimerâĂŹs disease," in *Asian Conference on Computer Vision*, pp. 340–353, Springer, 2014.

[108] W.-L. Zheng, J.-Y. Zhu, Y. Peng, and B.-L. Lu, "Eeg-based emotion classification using deep belief networks," in *Multimedia and Expo (ICME), 2014 IEEE International Conference on*, pp. 1–6, IEEE, 2014.

[109] W.-L. Zheng and B.-L. Lu, "Investigating critical frequency bands and channels for eeg-based emotion recognition with deep neural networks," *IEEE Transactions on Autonomous Mental Development*, vol. 7, no. 3, pp. 162–175, 2015.

[110] W.-L. Zheng, H.-T. Guo, and B.-L. Lu, "Revealing critical channels and frequency bands for emotion recognition from eeg with deep belief network," in *Neural Engineering (NER), 2015 7th International IEEE/EMBS Conference on*, pp. 154–157, IEEE, 2015.

[111] X. Jia, K. Li, X. Li, and A. Zhang, "A novel semi-supervised deep learning framework for affective state recognition on eeg signals," in *Bioinformatics and Bioengineering (BIBE), 2014 IEEE International Conference on*, pp. 30–37, IEEE, 2014.

[112] X. Yan, W.-L. Zheng, W. Liu, and B.-L. Lu, "Identifying gender differences in multimodal emotion recognition using bimodal deep autoencoder," in *International Conference on Neural Information Processing*, pp. 533–542, Springer, 2017.

[113] W. Liu, W.-L. Zheng, and B.-L. Lu, "Emotion recognition using multimodal deep learning," in *International Conference on Neural Information Processing*, pp. 521–529, Springer, 2016.

[114] X. Chai, Q. Wang, Y. Zhao, X. Liu, O. Bai, and Y. Li, "Unsupervised domain adaptation techniques based on auto-encoder for non-stationary eeg-based emotion recognition," *Computers in biology and medicine*, vol. 79, pp. 205–214, 2016.

[115] S. Alhagry, A. A. Fahmy, and R. A. El-Khoribi, "Emotion recognition based on eeg using lstm recurrent neural network," *Emotion*, vol. 8, no. 10, 2017.

[116] Z. Yin, M. Zhao, Y. Wang, J. Yang, and J. Zhang, "Recognition of emotions using multimodal physiological signals and an ensemble deep learning model," *Computer methods and programs in biomedicine*, vol. 140, pp. 93–110, 2017.

[117] P. Kawde and G. K. Verma, "Deep belief network based affect recognition from physiological signals," in *Electrical, Computer and Electronics (UPCON), 2017 4th IEEE Uttar Pradesh Section International Conference on*, pp. 587–592, IEEE, 2017.

[118] Y. Yuan, G. Xun, F. Ma, Q. Suo, H. Xue, K. Jia, and A. Zhang, "A novel channel-aware attention framework for multi-channel eeg seizure detection via multi-view deep learning," in *Biomedical & Health Informatics (BHI), 2018 IEEE EMBS International Conference on*, pp. 206–209, IEEE, 2018.

[119] I. Ullah, M. Hussain, H. Aboalsamh, *et al.*, "An automated system for epilepsy detection using eeg brain signals based on deep learning approach," *Expert Systems with Applications*, vol. 107, pp. 61–71, 2018.

[120] U. R. Acharya, S. L. Oh, Y. Hagiwara, J. H. Tan, and H. Adeli, "Deep convolutional neural network for the automated detection and diagnosis of seizure using eeg signals," *Computers in biology and medicine*, vol. 100, pp. 270–278, 2018.

[121] R. Schirrmeister, L. Gemein, K. Eggensperger, F. Hutter, and T. Ball, "Deep learning with convolutional neural networks for decoding and visualization of eeg pathology," in *Signal Processing in Medicine and Biology Symposium (SPMB), 2017 IEEE*, pp. 1–7, IEEE, 2017.

[122] K. Wang, Y. Zhao, Q. Xiong, M. Fan, G. Sun, L. Ma, and T. Liu, "Research on healthy anomaly detection model based on deep learning from multiple time-series physiological signals," *Scientific Programming*, vol. 2016, 2016.

[123] A. R. Johansen, J. Jin, T. Maszczyk, J. Dauwels, S. S. Cash, and M. B. Westover, "Epileptiform spike detection via convolutional neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pp. 754–758, IEEE, 2016.

[124] U. R. Acharya, S. L. Oh, Y. Hagiwara, J. H. Tan, H. Adeli, and D. P. Subha, "Automated eeg-based screening of depression using deep convolutional neural network," *Computer methods and programs in biomedicine*, vol. 161, pp. 103–113, 2018.

[125] F. C. Morabito, M. Campolo, C. Ieracitano, J. M. Ebadi, L. Bonanno, A. Bramanti, S. Desalvo, N. Mammone, and P. Bramanti, "Deep convolutional neural networks for classification of mild cognitive impaired and alzheimer's disease patients from scalp eeg recordings," in *Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI), 2016 IEEE 2nd International Forum on*, pp. 1–6, IEEE, 2016.

[126] A. H. Ansari, P. J. Cherian, A. Caicedo, G. Naulaers, M. De Vos, and S. Van Huffel, "Neonatal seizure detection using deep convolutional neural networks," *International journal of neural systems*, p. 1850011, 2018.

[127] L. Chu, R. Qiu, H. Liu, Z. Ling, T. Zhang, and J. Wang, "Individual recognition in schizophrenia using deep learning methods with random forest and voting classifiers: Insights from resting state eeg streams," *arXiv preprint arXiv:1707.03467*, 2017.

[128] G. Ruffini, D. Ibañez, M. Castellano, S. Dunne, and A. Soria-Frisch, "Eeg-driven rnn classification for prognosis of neurodegeneration in at-risk patients," in *International Conference on Artificial Neural Networks*, pp. 306–313, Springer, 2016.

[129] B. Yan, Y. Wang, Y. Li, Y. Gong, L. Guan, and S. Yu, "An eeg signal classification method based on sparse auto-encoders and support vector machine," in *Communications in China (ICCC), 2016 IEEE/CIC International Conference on*, pp. 1–6, IEEE, 2016.

[130] J. Turner, A. Page, T. Mohsenin, and T. Oates, "Deep belief networks used on high resolution multichannel electroencephalography data for seizure detection," in *2014 AAAI Spring Symposium Series*, 2014.

[131] A. Page, J. Turner, T. Mohsenin, and T. Oates, "Comparing raw data and feature extraction for seizure detection with deep learning methods.," in *FLAIRS Conference*, 2014.

[132] A. M. Al-kaysi, A. Al-Ani, and T. W. Boonstra, "A multichannel deep belief network for the classification of eeg data," in *International Conference on Neural Information Processing*, pp. 38–45, Springer, 2015.

[133] Y. Yuan, G. Xun, K. Jia, and A. Zhang, "A novel wavelet-based model for eeg epileptic seizure detection using multi-context learning," in *Bioinformatics and Biomedicine (BIBM), 2017 IEEE International Conference on*, pp. 694–699, IEEE, 2017.

[134] M.-P. Hosseini, H. Soltanian-Zadeh, K. Elisevich, and D. Pompili, "Cloud-based deep learning of big eeg data for epileptic seizure prediction," *arXiv preprint arXiv:1702.05192*, 2017.

[135] Q. Lin, S.-q. Ye, X.-m. Huang, S.-y. Li, M.-z. Zhang, Y. Xue, and W.-S. Chen, "Classification of epileptic eeg signals with stacked sparse autoencoder based on deep learning," in *International Conference on Intelligent Computing*, pp. 802–810, Springer, 2016.

[136] V. Shah, M. Golmohammadi, S. Ziyabari, E. Von Weltin, I. Obeid, and J. Picone, "Optimizing channel selection for seizure detection," in *Signal Processing in Medicine and Biology Symposium (SPMB), 2017 IEEE*, pp. 1–5, IEEE, 2017.

[137] M. Golmohammadi, S. Ziyabari, V. Shah, S. L. de Diego, I. Obeid, and J. Picone, "Deep architectures for automated seizure detection in scalp eegs," *arXiv preprint arXiv:1712.09776*, 2017.

[138] F. C. Morabito, M. Campolo, N. Mammone, M. Versaci, S. Franceschetti, F. Tagliavini, V. Sofia, D. Fatuzzo, A. Gambardella, A. Labate, *et al.*, "Deep learning representation from electroencephalography of early-stage creutzfeldt-jakob disease and features for differentiation from rapidly progressive dementia," *International journal of neural systems*, vol. 27, no. 02, p. 1650039, 2017.

[139] T. Wen and Z. Zhang, "Deep convolution neural network and autoencoders-based unsupervised feature learning of eeg signals," *IEEE Access*, vol. 6, pp. 25399–25410, 2018.

[140] S. M. Abdelfattah, G. M. Abdelrahman, and M. Wang, "Augmenting the size of eeg datasets using generative adversarial networks," in *2018 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–6, IEEE, 2018.

[141] S. Palazzo, C. Spampinato, I. Kavasidis, D. Giordano, and M. Shah, "Generative adversarial networks conditioned by brain signals," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3410–3418, 2017.

[142] I. Kavasidis, S. Palazzo, C. Spampinato, D. Giordano, and M. Shah, "Brain2image: Converting brain signals into images," in *Proceedings of the 25th ACM international conference on Multimedia*, pp. 1809–1817, ACM, 2017.

[143] S. Stober, D. J. Cameron, and J. A. Grahn, "Using convolutional neural networks to recognize rhythmï£ij stimuli from electroencephalography recordings," in *Advances in neural information processing systems*, pp. 1449–1457, 2014.

[144] S. Stober, D. J. Cameron, and J. A. Grahn, "Classifying eeg recordings of rhythm perception.," in *ISMIR*, pp. 649–654, 2014.

[145] S. Stober, A. Sternin, A. M. Owen, and J. A. Grahn, "Deep feature learning for eeg recordings," *arXiv preprint arXiv:1511.04306*, 2015.

[146] A. Sternin, S. Stober, J. Grahn, and A. Owen, "Tempo estimation from the eeg signal during perception and imagination of music," in *1st International Workshop on Brain-Computer Music Interfacing/11th International Symposium on Computer Music Multidisciplinary Research (BCMI/CMMRâĂŹ15)(Plymouth)*, 2015.

[147] S. Sarkar, K. Reddy, A. Dorgan, C. Fidopiastis, and M. Giering, "Wearable eeg-based activity recognition in phm-related service environment via deep learning," *Int. J. Progn. Health Manag*, vol. 7, pp. 1–10, 2016.

[148] P. Bashivan, I. Rish, and S. Heisig, "Mental state recognition via wearable eeg," *arXiv preprint arXiv:1602.00985*, 2016.

[149] J. Shang, W. Zhang, J. Xiong, and Q. Liu, "Cognitive load recognition using multi-channel complex network method," in *International Symposium on Neural Networks*, pp. 466–474, Springer, 2017.

[150] Y. Gordienko, S. Stirenko, Y. Kochura, O. Alienin, M. Novotarskiy, and N. Gordienko, "Deep learning for fatigue estimation on the basis of multimodal human-machine interactions," *arXiv preprint arXiv:1801.06048*, 2017.

[151] P. Bashivan, M. Yeasin, and G. M. Bidelman, "Single trial prediction of normal and excessive cognitive load through eeg feature fusion," in *Signal Processing in Medicine and Biology Symposium (SPMB), 2015 IEEE*, pp. 1–5, IEEE, 2015.

[152] P. Bashivan, I. Rish, M. Yeasin, and N. Codella, "Learning representations from eeg with deep recurrent-convolutional neural networks," *ICLR*, 2016.

[153] Z. Yin and J. Zhang, "Cross-session classification of mental workload levels using eeg and an adaptive deep learning model," *Biomedical Signal Processing and Control*, vol. 33, pp. 30–47, 2017.

[154] P. Li, W. Jiang, and F. Su, "Single-channel eeg-based mental fatigue detection based on deep belief network," in *Engineering in Medicine and Biology Society (EMBC), 2016 IEEE 38th Annual International Conference of the*, pp. 367–370, IEEE, 2016.

[155] R. Chai, S. H. Ling, P. P. San, G. R. Naik, T. N. Nguyen, Y. Tran, A. Craig, and H. T. Nguyen, "Improving eeg-based driver fatigue classification using sparse-deep belief networks," *Frontiers in neuroscience*, vol. 11, p. 103, 2017.

[156] L.-H. Du, W. Liu, W.-L. Zheng, and B.-L. Lu, "Detecting driving fatigue with multimodal deep learning," in *Neural Engineering (NER), 2017 8th International IEEE/EMBS Conference on*, pp. 74–77, IEEE, 2017.

[157] M. Hajinoroozi, Z. Mao, and Y. Huang, "Prediction of driver's drowsy and alert states from eeg signals with deep learning," in *Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2015 IEEE 6th International Workshop on*, pp. 493–496, IEEE, 2015.

[158] M. Hajinoroozi, J. Zhang, and Y. Huang, "Prediction of fatigue-related driver performance from eeg data by deep riemannian model," in *Engineering in Medicine and Biology Society (EMBC), 2017 39th Annual International Conference of the IEEE*, pp. 4167–4170, IEEE, 2017.

[159] Y.-C. Hung, Y.-K. Wang, M. Prasad, and C.-T. Lin, "Brain dynamic states analysis based on 3d convolutional neural network," in *Systems, Man, and*

Cybernetics (SMC), 2017 IEEE International Conference on, pp. 222–227, IEEE, 2017.

[160] M. Hajinoroozi, T.-P. Jung, C.-T. Lin, and Y. Huang, "Feature extraction with deep belief networks for driver's cognitive states prediction from eeg data," in *Signal and Information Processing (ChinaSIP), 2015 IEEE China Summit and International Conference on*, pp. 812–815, IEEE, 2015.

[161] P. P. San, S. H. Ling, R. Chai, Y. Tran, A. Craig, and H. Nguyen, "Eeg-based driver fatigue detection using hybrid deep generic model," in *Engineering in Medicine and Biology Society (EMBC), 2016 IEEE 38th Annual International Conference of the*, pp. 800–803, IEEE, 2016.

[162] M. A. Almogbel, A. H. Dang, and W. Kameyama, "Eeg-signals based cognitive workload detection of vehicle driver using deep learning," in *Advanced Communication Technology (ICACT), 2018 20th International Conference on*, pp. 256–259, IEEE, 2018.

[163] S. Narejo, E. Pasero, and F. Kulsoom, "Eeg based eye state classification using deep belief network and stacked autoencoder," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 6, no. 6, pp. 3131–3141, 2016.

[164] T. K. Reddy and L. Behera, "Online eye state recognition from eeg data using deep architectures," in *Systems, Man, and Cybernetics (SMC), 2016 IEEE International Conference on*, pp. 000712–000717, IEEE, 2016.

[165] V. Baltatzis, K.-M. Bintsi, G. K. Apostolidis, and L. J. Hadjileontiadis, "Bullying incidences identification within an immersive environment using hd eeg-based analysis: A swarm decomposition and deep learning approach," *Scientific reports*, vol. 7, no. 1, p. 17292, 2017.

[166] P. Khurana, A. Majumdar, and R. Ward, "Class-wise deep dictionaries for eeg classification," in *Neural Networks (IJCNN), 2016 International Joint Conference on*, pp. 3556–3563, IEEE, 2016.

[167] M. Völker, R. T. Schirrmeister, L. D. Fiederer, W. Burgard, and T. Ball, "Deep transfer learning for error decoding from non-invasive eeg," in *Brain-Computer Interface (BCI), 2018 6th International Conference on*, pp. 1–6, IEEE, 2018.

[168] J. A. Mioranda-Correa and I. Patras, "A multi-task cascaded network for prediction of affect, personality, mood and social context using eeg signals," in *Automatic Face & Gesture Recognition (FG 2018), 2018 13th IEEE International Conference on*, pp. 373–380, IEEE, 2018.

[169] M. J. Putten, S. Olbrich, and M. Arns, "Predicting sex from brain rhythms with deep learning," *Scientific reports*, vol. 8, no. 1, p. 3069, 2018.

[170] L. G. Hernández, O. M. Mozos, J. M. Ferrández, and J. M. Antelis, "Eeg-based detection of braking intention under different car driving conditions," *Frontiers in neuroinformatics*, vol. 12, 2018.

[171] J. Behncke, R. T. Schirrmeister, W. Burgard, and T. Ball, "The signature of robot action success in eeg signals of a human observer: Decoding and visualization using deep convolutional neural networks," in *Brain-Computer Interface (BCI), 2018 6th International Conference on*, pp. 1–6, IEEE, 2018.

[172] J. Teo, C. L. Hou, and J. Mountstephens, "Deep learning for eeg-based preference classification," in *AIP Conference Proceedings*, vol. 1891, p. 020141, AIP Publishing, 2017.

[173] J. Teo, C. L. Hou, and J. Mountstephens, "Preference classification using electroencephalography (eeg) and deep learning," *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, vol. 10, no. 1-11, pp. 87–91, 2018.

[174] V. Lawhern, A. Solon, N. Waytowich, S. M. Gordon, C. Hung, and B. J. Lance, "Eegnet: a compact convolutional neural network for eeg-based brain–computer interfaces," *Journal of neural engineering*, 2018.

[175] M.-P. Hosseini, T. X. Tran, D. Pompili, K. Elisevich, and H. Soltanian-Zadeh, "Deep learning with edge computing for localization of epileptogenicity using multimodal rs-fmri and eeg big data," in *Autonomic Computing (ICAC), 2017 IEEE International Conference on*, pp. 83–92, IEEE, 2017.

[176] P. Garg, E. Davenport, G. Murugesan, B. Wagner, C. Whitlow, J. Maldjian, and A. Montillo, "Automatic 1d convolutional neural network-based detection of artifacts in meg acquired without electrooculography or electrocardiography," in *Pattern Recognition in Neuroimaging (PRNI), 2017 International Workshop on*, pp. 1–4, IEEE, 2017.

[177] A. Hasasneh, N. Kampel, P. Sripad, N. J. Shah, and J. Dammers, "Deep learning approach for automatic classification of ocular and cardiac artifacts in meg data," *Journal of Engineering*, vol. 2018, 2018.

[178] A. Antoniades, L. Spyrou, D. Martin-Lopez, A. Valentin, G. Alarcon, S. Sanei, and C. C. Took, "Deep neural architectures for mapping scalp to intracranial eeg," *International journal of neural systems*, p. 1850009, 2018.

[179] N.-S. Kwak, K.-R. Müller, and S.-W. Lee, "A convolutional neural network for steady state visual evoked potential classification under ambulatory environment," *PloS one*, vol. 12, no. 2, p. e0172578, 2017.

[180] K. Kawasaki, T. Yoshikawa, and T. Furuhashi, "Visualizing extracted feature by deep learning in p300 discrimination task," in *Soft Computing and Pattern Recognition (SoCPaR), 2015 7th International Conference of*, pp. 149–154, IEEE, 2015.

[181] H. Cecotti and A. Graser, "Convolutional neural networks for p300 detection with application to brain-computer interfaces," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 3, pp. 433–445, 2011.

[182] R. Maddula, J. Stivers, M. Mousavi, S. Ravindran, and V. de Sa, "Deep recurrent convolutional neural networks for classifying p300 bci signals," in *Proceedings of the 7th Graz Brain-Computer Interface Conference, Graz, Austria*, pp. 18–22, 2017.

[183] M. Liu, W. Wu, Z. Gu, Z. Yu, F. Qi, and Y. Li, "Deep learning based on batch normalization for p300 signal detection," *Neurocomputing*, vol. 275, pp. 288–297, 2018.

[184] T. Koike-Akino, R. Mahajan, T. K. Marks, Y. Wang, S. Watanabe, O. Tuzel, and P. Orlik, "High-accuracy user identification using eeg biometrics," in *2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 854–858, IEEE, 2016.

[185] Z. Mao, W. X. Yao, and Y. Huang, "Eeg-based biometric identification with deep learning," in *Neural Engineering (NER), 2017 8th International IEEE/EMBS Conference on*, pp. 609–612, IEEE, 2017.

[186] T. Hiroyasu, K. Hanawa, and U. Yamamoto, "Gender classification of subjects from cerebral blood flow changes using deep learning," in *Computational Intelligence and Data Mining (CIDM), 2014 IEEE Symposium on*, pp. 229–233, IEEE, 2014.

[187] A. HACHEM, M. M. B. Khelifa, A. M. Alimi, P. Gorce, S. V. ARASU, S. BAULKANI, S. K. BISOY, P. K. PATTNAIK, S. RAVINDRAN, N. PALANISAMY, *et al.*, "Effect of fatigue on ssvep during virtual wheelchair navigation," *Journal of Theoretical and Applied Information Technology*, vol. 65, no. 1, 2014.

[188] N. R. Waytowich, V. Lawhern, J. O. Garcia, J. Cummings, J. Faller, P. Sajda, and J. M. Vettel, "Compact convolutional neural networks for classification of asynchronous steady-state visual evoked potentials," *arXiv preprint arXiv:1803.04566*, 2018.

[189] R. Manor and A. B. Geva, "Convolutional neural network for multi-category rapid serial visual presentation bci," *Frontiers in computational neuroscience*, vol. 9, p. 146, 2015.

[190] H. Cecotti, "Convolutional neural networks for event-related potential detection: impact of the architecture," in *Engineering in Medicine and Biology Society (EMBC), 2017 39th Annual International Conference of the IEEE*, pp. 2031–2034, IEEE, 2017.

[191] M. Hajinoroozi, Z. Mao, Y.-P. Lin, and Y. Huang, "Deep transfer learning for cross-subject and cross-experiment prediction of image rapid serial visual presentation events from eeg data," in *International Conference on Augmented Cognition*, pp. 45–55, Springer, 2017.

[192] J. Pérez-Benítez, J. Pérez-Benítez, and J. Espina-Hernández, "Development of a brain computer interface interface using multi-frequency visual stimulation and deep neural networks," in *Electronics, Communications and Computers (CONIELECOMP), 2018 International Conference on*, pp. 18–24, IEEE, 2018.

[193] J. Shamwell, H. Lee, H. Kwon, A. R. Marathe, V. Lawhern, and W. Nothwang, "Single-trial eeg rsvp classification using convolutional neural networks," in *Micro-and Nanotechnology Sensors, Systems, and Applications VIII*, vol. 9836, p. 983622, International Society for Optics and Photonics, 2016.

[194] J. Kulasingham, V. Vibujithan, and A. De Silva, "Deep belief networks and stacked autoencoders for the p300 guilty knowledge test," in *Biomedical Engineering and Sciences (IECBES), 2016 IEEE EMBS Conference on*, pp. 127–132, IEEE, 2016.

[195] Q. Liu, X.-G. Zhao, Z.-G. Hou, and H.-G. Liu, "Deep belief networks for eeg-based concealed information test," in *International Symposium on Neural Networks*, pp. 498–506, Springer, 2017.

[196] L. Marc Moreno, "Deep learning for brain tumor segmentation," *Master diss. University of Colorado Colorado Springs.*, 2017.

[197] M. Havaei, A. Davy, D. Warde-Farley, A. Biard, A. Courville, Y. Bengio, C. Pal, P.-M. Jodoin, and H. Larochelle, "Brain tumor segmentation with deep neural networks," *Medical image analysis*, vol. 35, pp. 18–31, 2017.

[198] V. Shreyas and V. Pankajakshan, "A deep learning architecture for brain tumor segmentation in mri images," in *Multimedia Signal Processing (MMSP), 2017 IEEE 19th International Workshop on*, pp. 1–6, IEEE, 2017.

[199] A. Antoniades, L. Spyrou, C. C. Took, and S. Sanei, "Deep learning for epileptic intracranial eeg data," in *Machine Learning for Signal Processing (MLSP), 2016 IEEE 26th International Workshop on*, pp. 1–6, IEEE, 2016.

[200] S. M. Plis, D. R. Hjelm, R. Salakhutdinov, E. A. Allen, H. J. Bockholt, J. D. Long, H. J. Johnson, J. S. Paulsen, J. A. Turner, and V. D. Calhoun, "Deep learning for neuroimaging: a validation study," *Frontiers in neuroscience*, vol. 8, p. 229, 2014.

[201] H.-I. Suk, C.-Y. Wee, S.-W. Lee, and D. Shen, "State-space model with deep learning for functional dynamics estimation in resting-state fmri," *NeuroImage*, vol. 129, pp. 292–307, 2016.

[202] G. Huve, K. Takahashi, and M. Hashimoto, "Brain-computer interface using deep neural network and its application to mobile robot control," in *Advanced Motion Control (AMC), 2018 IEEE 15th International Workshop on*, pp. 169–174, IEEE, 2018.

[203] R. Parasuraman and Y. Jiang, "Individual differences in cognition, affect, and performance: Behavioral, neuroimaging, and molecular genetic approaches," *Neuroimage*, vol. 59, no. 1, pp. 70–82, 2012.

[204] N. Naseer, N. K. Qureshi, F. M. Noori, and K.-S. Hong, "Analysis of different classification techniques for two-class functional near-infrared spectroscopy-based brain-computer interface," *Computational intelligence and neuroscience*, vol. 2016, 2016.

[205] J. J. M. Monaghan, T. Goehring, X. Yang, F. Bolner, S. Wang, M. C. M. Wright, and S. Bleeck, "Auditory inspired machine learning techniques can improve speech intelligibility and quality for hearing-impaired listeners," *The Journal of the Acoustical Society of America*, vol. 141, no. 3, pp. 1985–1998, 2017.

[206] T. Goehring, F. Bolner, J. J. Monaghan, B. van Dijk, A. Zarowski, and S. Bleeck, "Speech enhancement based on neural networks improves speech intelligibility in noise for cochlear implant users," *Hearing Research*, vol. 344, pp. 183 – 194, 2017.

[207] N. Ding and J. Z. Simon, "Emergence of neural encoding of auditory objects while listening to competing speakers," *Proceedings of the National Academy of Sciences*, vol. 109, no. 29, pp. 11854–11859, 2012.

[208] J. OÃŢSullivan, Z. Chen, J. Herrero, G. M. McKhann, S. A. Sheth, A. D. Mehta, and N. Mesgarani, "Neural decoding of attentional selection in multi-speaker environments without access to clean sources," *Journal of neural engineering*, vol. 14, no. 5, p. 056001, 2017.

[209] J. A. O'Sullivan, A. J. Power, N. Mesgarani, S. Rajaram, J. J. Foxe, B. G. Shinn-Cunningham, M. Slaney, S. A. Shamma, and E. C. Lalor, "Attentional Selection in a Cocktail Party Environment Can Be Decoded from Single-Trial EEG," *Cerebral Cortex*, vol. 25, pp. 1697–1706, 01 2014.

[210] B. Mirkovic, M. G. Bleichner, M. De Vos, and S. Debener, "Target speaker detection with concealed eeg around the ear," *Frontiers in neuroscience*, vol. 10, p. 349, 2016.

[211] C. Han, J. O'Sullivan, Y. Luo, J. Herrero, A. D. Mehta, and N. Mesgarani, "Speaker-independent auditory attention decoding without access to clean speech sources," *Science Advances*, vol. 5, no. 5, 2019.

[212] G. Ciccarelli, M. Nolan, J. Perricone, P. Calamia, S. Haro, J. O'Sullivan, N. Mesgarani, T. Quatieri, and C. J. Smalt, "Comparison of two-talker attention decoding from eeg with nonlinear neural networks and linear methods," *bioRxiv*, 2018.

[213] L. Deckers, N. Das, A. H. Ansari, A. Bertrand, and T. Francart, "Eeg-based detection of the attended speaker and the locus of auditory attention with convolutional neural networks," *bioRxiv*, 2018.

[214] G. Muhammad, M. Alhamid, M. Hossain, A. Almogren, and A. Vasilakos, "Enhanced living by assessing voice pathology using a co-occurrence matrix," *Sensors*, vol. 17, no. 2, p. 267, 2017.

[215] M. Rashid and X. Han, "Gesture control of zigbee connected smart home internet of things," in *Informatics, Electronics and Vision (ICIEV), 2016 5th International Conference on*.

[216] S. Kumar, "Ubiquitous smart home system using android application," *arXiv preprint arXiv:1402.2114*, 2014.

[217] M. H. Alomari, A. Abubaker, A. Turani, A. M. Baniyounes, and A. Manasreh, "Eeg mouse: A machine learning-based brain computer interface," *International Journal of Advanced Research in Computer Science and Applications*, 2014.

[218] T. C. Major and J. M. Conrad, "The effects of pre-filtering and individualizing components for electroencephalography neural network classification," in *SoutheastCon 2017*, pp. 1–6, IEEE, 2017.

[219] L. Sun *et al.*, "Classification of imagery motor eeg data with wavelet denoising and features selection," in *2016 International Conference on Wavelet Analysis and Pattern Recognition (ICWAPR)*, pp. 184–188, IEEE, 2016.

[220] H. V. Shenoy, A. P. Vinod, and C. Guan, "Shrinkage estimator based regularization for eeg motor imagery classification," in *2015 10th International Conference on Information, Communications and Signal Processing (ICICS)*, pp. 1–5, IEEE, 2015.

[221] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[222] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Fifteenth annual conference of the international speech communication association*, 2014.

[223] A. Page, C. Sagedy, E. Smith, N. Attaran, T. Oates, and T. Mohsenin, "A flexible multichannel eeg feature extractor and classifier for seizure detection," *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2015.

[224] B. Albert, J. Zhang, A. Noyvirt, R. Setchi, H. Sjaaheim, S. Velikova, and F. Strisland, "Automatic eeg processing for the early diagnosis of traumatic brain injury," *Procedia Computer Science*, vol. 96, pp. 703–712, 2016.

[225] A. M. Al-Kaysi, A. Al-Ani, C. K. Loo, T. Y. Powell, D. M. Martin, M. Breakspear, and T. W. Boonstra, "Predicting tdcs treatment outcomes of patients with major depressive disorder using automated eeg classification," *Journal of affective disorders*, vol. 208, pp. 597–603, 2017.

[226] X. An, D. P. Kuang, X. J. Guo, Y. L. Zhao, and L. H. He *Intelligent Computing in Bioinformatics*.

[227] C. Ward, J. Picone, and I. Obeid, "Applications of ubms and i-vectors in eeg subject verification," in *Engineering in Medicine and Biology Society (EMBC), 2016 IEEE 38th Annual International Conference of the*, 2016.

[228] Y. R. Tabar and U. Halici, "A novel deep learning approach for classification of EEG motor imagery signals," *Journal of Neural Engineering*, 2017.

[229] Z. Mu, J. Yin, and J. Hu, "Design of smart home system using eeg signal," *Metallurgical & Mining Industry*, no. 6, 2015.

[230] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," *arXiv preprint arXiv:1409.2329*, 2014.

[231] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[232] G. Taguchi, *System of experimental design: engineering methods to optimize quality and minimize costs*. UNIPUB/Kraus International Publications, 1987.

[233] G. Schalk, D. J. McFarland, T. Hinterberger, N. Birbaumer, and J. R. Wolpaw, "Bci2000: a general-purpose brain-computer interface (bci) system," *IEEE Transactions on Biomedical Engineering*, vol. 51, no. 6, pp. 1034–1043, 2004.

[234] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000.

[235] M. Tolić and F. Jović, "Classification of wavelet transformed eeg signals with neural network for imagined mental and motor tasks," *Kineziologija*, 2013.

[236] O. R. Pinheiro, L. R. Alves, M. Romero, and J. R. de Souza, "Wheelchair simulator game for training people with severe disabilities," in *Technology and Innovation in Sports, Health and Wellbeing (TISHW), International Conference on*, IEEE, 2016.

[237] R. Djemal, A. G. Bazyed, K. Belwafi, S. Gannouni, and W. Kaaniche, "Three-class eeg-based motor imagery classification using phase-space reconstruction technique," *Brain Sciences*, vol. 6, no. 3, p. 36, 2016.

[238] D. Wang, D. Miao, and G. Blohm, "Multi-class motor imagery eeg decoding for brain-computer interfaces," *Frontiers in Neuroscience*, vol. 6, p. 151, 2012.

[239] M. Wairagkar, I. Zoulias, V. Oguntosin, Y. Hayashi, and S. Nasuto, "Movement intention based brain computer interface for virtual reality and soft robotics rehabilitation using novel autocorrelation analysis of eeg," in *Biomedical Robotics and Biomechatronics (BioRob), 2016 6th IEEE International Conference on*, pp. 685–685, IEEE, 2016.

[240] L. Rossini, D. Izzo, and L. Summerer, "Brain-machine interfaces for space applications," in *Engineering in Medicine and Biology Society, 2009. EMBC 2009. Annual International Conference of the IEEE*, pp. 520–523, IEEE, 2009.

[241] Y. Zhang, G. Zhou, J. Jin, Q. Zhao, X. Wang, and A. Cichocki, "Sparse bayesian classification of eeg for brain–computer interface," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 11, pp. 2256–2267, 2016.

[242] Y. Zhang, G. Zhou, J. Jin, X. Wang, and A. Cichocki, "Frequency recognition in ssvep-based bci using multiset canonical correlation analysis," *International Journal of Neural Systems*, vol. 24, no. 04, p. 1450013, 2014.

[243] M. Arvaneh, C. Guan, K. K. Ang, and C. Quek, "Optimizing spatial filters by minimizing within-class dissimilarities in electroencephalogram-based brain–computer interface," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 24, no. 4, pp. 610–619, 2013.

[244] B. Yahya-Zoubir, M. Bentlemsan, E.-T. Zemouri, and K. Ferroudji, "Adaptive time window for eeg-based motor imagery classification," in *Proceedings of the International Conference on Intelligent Information Processing, Security and Advanced Communication*, p. 83, ACM, 2015.

[245] R. Santana, L. Bonnet, J. Legény, and A. Lécuyer, "Introducing the use of model-based evolutionary algorithms for eeg-based motor imagery classification," in *Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation*, pp. 1159–1166, ACM, 2012.

[246] S. Bhattacharyya, A. Sengupta, T. Chakraborti, A. Konar, and D. Tibarewala, "Automatic feature selection of motor imagery eeg signals using differential evolution and learning automata," *Medical & biological engineering & computing*, vol. 52, no. 2, pp. 131–139, 2014.

[247] G. R. Müller-Putz, P. Ofner, A. Schwarz, J. Pereira, A. Pinegger, C. L. Dias, L. Hehenberger, R. Kobler, and A. I. Sburlea, "Towards non-invasive eeg-based arm/hand-control in users with spinal cord injury," in *Brain-Computer Interface (BCI), 2017 5th International Winter Conference on*, pp. 63–65, IEEE, 2017.

[248] L. Vézard, P. Legrand, M. Chavent, F. Faïta-Aïnseba, and L. Trujillo, "Eeg classification for the detection of mental states," *Applied Soft Computing*, vol. 32, pp. 113–131, 2015.

[249] H. Meisheri, N. Ramrao, and S. K. Mitra, "Multiclass common spatial pattern with artifacts removal methodology for eeg signals," in *Computational and Business Intelligence (ISCBI), 2016 4th International Symposium on*, pp. 90–93, IEEE, 2016.

[250] T. Shiratori, H. Tsubakida, A. Ishiyama, and Y. Ono, "Three-class classification of motor imagery eeg data including âĂIJrest stateâĂİ using filter-bank multi-class common spatial pattern," in *Brain-Computer Interface (BCI), 2015 3rd International Winter Conference on*, pp. 1–4, IEEE, 2015.

[251] M. Rahman, W. Ma, D. Tran, and J. Campbell, "A comprehensive survey of the feature extraction methods in the eeg research," *Algorithms and Architectures for Parallel Processing*, pp. 274–283, 2012.

[252] Y. Zhang, B. Liu, X. Ji, and D. Huang, "Classification of eeg signals based on autoregressive model and wavelet packet decomposition," *Neural Processing Letters*, pp. 1–14, 2016.

[253] L. Duan, Y. Xu, S. Cui, J. Chen, and M. Bao, "Feature extraction of motor imagery eeg based on extreme learning machine auto-encoder," in *Proceedings of ELM-2015 Volume 1*, pp. 361–370, Springer, 2016.

[254] H. Kang and S. Choi, "Bayesian common spatial patterns for multi-subject eeg classification," *Neural Networks*, vol. 57, pp. 39–50, 2014.

[255] M. J. Eugster, T. Ruotsalo, M. M. Spapé, I. Kosunen, O. Barral, N. Ravaja, G. Jacucci, and S. Kaski, "Predicting term-relevance from brain signals," in *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*, pp. 425–434, 2014.

[256] H. Ji, J. Li, R. Lu, R. Gu, L. Cao, and X. Gong, "Eeg classification for hybrid brain-computer interface using a tensor based multiclass multimodal analysis scheme," *Computational Intelligence and Neuroscience*, vol. 2016, p. 51, 2016.

[257] N. T. H. Anh, T. H. Hoang, V. T. Thang, T. Q. Bui, *et al.*, "An artificial neural network approach for electroencephalographic signal classification towards brain-computer interface implementation," in *Computing & Communication Technologies, Research, Innovation, and Vision for the Future (RIVF), 2016 IEEE RIVF International Conference on*, pp. 205–210, IEEE, 2016.

[258] T. E. Olivier, S. Du, B. J. van Wyk, and Y. Hamam, "Independent components for eeg signal classification," in *Proceedings of the 2016 International Conference on Intelligent Information Processing*, p. 33, ACM, 2016.

[259] T. Nguyen, S. Nahavandi, A. Khosravi, D. Creighton, and I. Hettiarachchi, "Eeg signal analysis for bci application using fuzzy system," in *Neural Networks (IJCNN), 2015 International Joint Conference on*, pp. 1–8, IEEE, 2015.

[260] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, ACM, 2016.

[261] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of Statistics*, pp. 1189–1232, 2001.

[262] S. Kamburugamuve, L. Christiansen, and G. Fox, "A framework for real time processing of sensor data in the cloud," *Journal of Sensors*, vol. 2015, 2015.

[263] N. Bigdely-Shamlo, T. Mullen, C. Kothe, K.-M. Su, and K. A. Robbins, "The prep pipeline: standardized preprocessing for large-scale eeg analysis," *Frontiers in neuroinformatics*, vol. 9, p. 16, 2015.

[264] P. Cavanagh *et al.*, "Attention-based motion perception," *Science*, vol. 257, no. 5076, pp. 1563–1565, 1992.

[265] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, p. 529, 2015.

[266] B. Markham and J. Townshend, "Land cover classification accuracy as a function of sensor spatial resolution," 1981.

[267] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with lstm," 1999.

[268] D. Zhang, L. Yao, X. Zhang, S. Wang, W. Chen, and R. Boots, "Eeg-based intention recognition from spatio-temporal representations via cascade and parallel convolutional recurrent neural networks," in *AAAI*, 2018.

[269] M. Tan, C. d. Santos, B. Xiang, and B. Zhou, "Lstm-based deep learning models for non-factoid answer selection," *arXiv preprint arXiv:1511.04108*, 2015.

[270] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *NeurIPS*, pp. 577–585, 2015.

[271] Z. Wang, T. Schaul, M. Hessel, H. Van Hasselt, M. Lanctot, and N. De Freitas, "Dueling network architectures for deep reinforcement learning," *arXiv preprint arXiv:1511.06581*, 2015.

[272] M. Tokic, "Adaptive ε-greedy exploration in reinforcement learning based on value differences," in *Annual Conference on Artificial Intelligence*, pp. 203–210, Springer, 2010.

[273] F. A. Gers and E. Schmidhuber, "Lstm recurrent networks learn simple context-free and context-sensitive languages," *IEEE Transactions on Neural Networks*, vol. 12, no. 6, pp. 1333–1340, 2001.

[274] M. Sundermeyer, R. Schlüter, and H. Ney, "Lstm neural networks for language modeling," in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[275] Z. C. Lipton, D. C. Kale, C. Elkan, and R. Wetzel, "Learning to diagnose with lstm recurrent neural networks," *arXiv preprint arXiv:1511.03677*, 2015.

[276] L. Yao, Q. Z. Sheng, X. Li, T. Gu, M. Tan, X. Wang, S. Wang, and W. Ruan, "Compressive representation for device-free activity recognition with passive rfid signal strength," *IEEE Transactions on Mobile Computing*, vol. 17, no. 2, pp. 293–306, 2018.

[277] B. Fida, D. Bibbo, I. Bernabucci, A. Proto, S. Conforto, and M. Schmid, "Real time event-based segmentation to classify locomotion activities through a single inertial sensor," in *MOBIHEALTH*, pp. 104–107, 2015.

[278] A. Chowdhury, D. Tjondronegoro, V. Chandran, and S. Trost, "Physical activity recognition using posterior-adapted class-based fusion of multiaccelerometers data," *IEEE Journal of Biomedical and Health Informatics*, 2017.

[279] S. M. Erfani, M. Baktashmotlagh, M. Moshtaghi, V. Nguyen, C. Leckie, J. Bailey, and K. Ramamohanarao, "From shared subspaces to shared landmarks: A robust multi-source classification approach.," in *AAAI*, pp. 1854–1860, 2017.

[280] Y. Zheng, Q. Liu, E. Chen, Y. Ge, and J. L. Zhao, "Time series classification using multi-channels deep convolutional neural networks," in *International Conference on Web-Age Information Management*, pp. 298–310, Springer, 2014.

[281] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[282] A. Vallabhaneni, T. Wang, and B. He, "Brain computer interface," in *Neural engineering*, pp. 85–121, Springer, 2005.

[283] C. V. Russoniello, K. OâĂŹBrien, and J. M. Parks, "The effectiveness of casual video games in improving mood and decreasing stress," *Journal of CyberTherapy & Rehabilitation*, vol. 2, no. 1, pp. 53–66, 2009.

[284] V. Schetinin, L. Jakaite, N. Nyah, D. Novakovic, and W. Krzanowski, "Feature extraction with gmdh-type neural networks for eeg-based person identification," *IJNS*, p. 1750064, 2017.

[285] V. Veeriah, R. Durvasula, and G.-J. Qi, "Deep learning architecture with dynamically programmed layers for brain connectome prediction," in *SIGKDD*, pp. 1205–1214, ACM, 2015.

[286] E. Acar, C. A. Bingol, H. Bingol, R. Bro, and B. Yener, "Seizure recognition on epilepsy feature tensor," in *EMBS*, pp. 4273–4276, IEEE, 2007.

[287] H. Adeli, S. Ghosh-Dastidar, and N. Dadmehr, "A wavelet-chaos methodology for analysis of eegs and eeg subbands to detect seizure and epilepsy," *IEEE Transactions on Biomedical Engineering*, vol. 54, no. 2, pp. 205–211, 2007.

[288] M. Fraschini, A. Hillebrand, M. Demuru, L. Didaci, and G. L. Marcialis, "An eeg-based biometric system using eigenvector centrality in resting state brain networks," *IEEE Signal Processing Letters*, vol. 22, no. 6, pp. 666–670, 2015.

[289] X. Zhang, L. Yao, C. Huang, S. Wang, M. Tan, G. Long, and C. Wang, "Multi-modality sensor data classification with selective attention," in *IJCAI-18*, pp. 3111–3117, 2018.

[290] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NeurIPS*, pp. 1097–1105, 2012.

[291] M. Golmohammadi, V. Shah, S. Lopez, S. Ziyabari, S. Yang, J. Camaratta, I. Obeid, and J. Picone, "The tuh eeg seizure corpus," in *ACNS Annual Meeting*, p. 1, 2017.

[292] M. M. or Rashid and M. Ahmad, "Classification of motor imagery hands movement using levenberg-marquardt algorithm based on statistical features of eeg signal," in *ICEEICT*, pp. 1–6, IEEE, 2016.

[293] X. Ma, S. Qiu, C. Du, J. Xing, and H. He, "Improving eeg-based motor imagery classification via spatial and temporal recurrent neural networks," in *EMBC*, pp. 1903–1906, IEEE, 2018.

[294] J. Sita and G. Nair, "Feature extraction and classification of eeg signals for mapping motor area of the brain," in *ICCC*, pp. 463–468, IEEE, 2013.

[295] M. H. Alomari, A. AbuBaker, A. Turani, A. M. Baniyounes, and A. Manasreh, "Eeg mouse: A machine learning-based brain computer interface," *Int. J. Adv. Comput. Sci. Appl*, vol. 5, pp. 193–198, 2014.

[296] P. Szczuko, "Real and imaginary motion classification based on rough set analysis of eeg signals for multimedia applications," *Multimedia Tools and Applications*, vol. 76, no. 24, pp. 25697–25711, 2017.

[297] C. A. Stefano Filho, R. Attux, and G. Castellano, "Eeg sensorimotor rhythmsâĂŹ variation and functional connectivity measures during motor imagery: linear relations and classification approaches," *PeerJ*, vol. 5, p. e3983, 2017.

[298] Y. Kim, J. Ryu, K. K. Kim, C. C. Took, D. P. Mandic, and C. Park, "Motor imagery classification using mu and beta rhythms of eeg with strong uncorrelating transform based complex common spatial patterns," *Computational intelligence and neuroscience*, vol. 2016, pp. 1–14, 2016.

[299] L. Ma, J. W. Minett, T. Blu, and W. S. Wang, "Resting state eeg-based biometrics for individual identification using convolutional neural networks," in *EMBC*, pp. 2848–2851, IEEE, 2015.

[300] S. Yang and F. Deravi, "Novel hht-based features for biometric identification using eeg signals," in *ICPR*, pp. 1922–1927, IEEE, 2014.

[301] D. Rodrigues, G. F. Silva, J. P. Papa, A. N. Marana, and X.-S. Yang, "Eeg-based person identification through binary flower pollination algorithm," *Expert Systems with Applications*, vol. 62, pp. 81–90, 2016.

[302] K. P. Thomas and A. P. Vinod, "Biometric identification of persons using sample entropy features of eeg during rest state," in *SMC*, pp. 003487–003492, IEEE, 2016.

[303] S. Ziyabari, V. Shah, M. Golmohammadi, I. Obeid, and J. Picone, "Objective evaluation metrics for automatic classification of eeg events," *arXiv preprint arXiv:1712.10107*, 2017.

[304] A. Harati, M. Golmohammadi, S. Lopez, I. Obeid, and J. Picone, "Improved eeg event classification using differential energy," in *Signal Processing in Medicine and Biology Symposium (SPMB), 2015 IEEE*, pp. 1–4, IEEE, 2015.

[305] Y. Zhang, S. Yang, Y. Liu, Y. Zhang, B. Han, and F. Zhou, "Integration of 24 feature types to accurately detect and predict seizures using scalp eeg signals," *Sensors (Basel, Switzerland)*, vol. 18, no. 5, 2018.

[306] T. R. Goodwin and S. M. Harabagiu, "Deep learning from eeg reports for inferring underspecified information," *AMIA Summits on Translational Science Proceedings*, vol. 2017, pp. 112–121, 2017.

[307] M. Golmohammadi, A. H. H. N. Torbati, S. L. de Diego, I. Obeid, and J. Picone, "Automatic analysis of eegs using big data and hybrid deep learning architectures," *arXiv preprint arXiv:1712.09771*, 2017.

[308] K. Ghasedi Dizaji, X. Wang, and H. Huang, "Semi-supervised generative adversarial network for gene expression inference," in *The 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ACM, 2018.

[309] L. Yao, F. Nie, Q. Z. Sheng, T. Gu, X. Li, and S. Wang, "Learning from less for better: semi-supervised activity recognition via shared structure discovery," in *UbiComp*, ACM, 2016.

[310] L. Peng, W. Chen, W. Zhou, F. Li, J. Yang, and J. Zhang, "An immune-inspired semi-supervised algorithm for breast cancer diagnosis," *Computer methods and programs in biomedicine*, vol. 134, 2016.

[311] C. Gong, D. Tao, S. J. Maybank, W. Liu, G. Kang, and J. Yang, "Multi-modal curriculum learning for semi-supervised image classification," *IEEE Transactions on Image Processing*, vol. 25, no. 7, 2016.

[312] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling, "Semi-supervised learning with deep generative models," in *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[313] C. K. Sønderby, T. Raiko, L. Maaløe, S. K. Sønderby, and O. Winther, "Ladder variational autoencoders," in *Advances in neural information processing systems (NIPS)*, 2016.

[314] S. Narayanaswamy, T. B. Paige, J.-W. Van de Meent, A. Desmaison, N. Goodman, P. Kohli, F. Wood, and P. Torr, "Learning disentangled representations with semi-supervised deep generative models," in *NIPS*, 2017.

[315] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, "Adversarial autoencoders," *arXiv preprint arXiv:1511.05644*, 2015.

[316] A. Odena, "Semi-supervised learning with generative adversarial networks," *arXiv preprint arXiv:1606.01583*, 2016.

[317] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *Advances in Neural Information Processing Systems (NIPS)*, 2016.

[318] J. Cao, Y. Guo, Q. Wu, C. Shen, and M. Tan, "Adversarial learning with local coordinate coding," *The International Conference of Machine Learning (ICML)*, 2018.

[319] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[320] L. Maaløe, C. K. Sønderby, S. K. Sønderby, and O. Winther, "Auxiliary deep generative models," *arXiv preprint:1602.05473*, 2016.

[321] J. T. Springenberg, "Unsupervised and semi-supervised learning with categorical generative adversarial networks," *The International Conference on Learning Representations (ICLR)*, 2016.

[322] J. Walker, C. Doersch, A. Gupta, and M. Hebert, "An uncertain future: Forecasting from static images using variational autoencoders," in *European Conference on Computer Vision*, Springer, 2016.

[323] K. Chen, L. Yao, X. Wang, D. Zhang, T. Gu, Z. Yu, and Z. Yang, "Interpretable parallel recurrent neural networks with convolutional attentions for multi-modality activity modeling," *International Joint Conference on Neural Networks (IJCNN)*, 2018.

[324] O. D. Lara, A. J. Pérez, M. A. Labrador, and J. D. Posada, "Centinela: A human activity recognition system based on acceleration and vital sign data," *Pervasive and mobile computing*, vol. 8, no. 5, 2012.

[325] H. Guo, L. Chen, L. Peng, and G. Chen, "Wearable sensor based multimodal human activity recognition exploiting the diversity of classifier ensemble," in *UbiComp*, ACM, 2016.

[326] B. Fida, D. Bibbo, I. Bernabucci, A. Proto, S. Conforto, and M. Schmid, "Real time event-based segmentation to classify locomotion activities through a single inertial sensor," in *MobiHealth*, 2015.

[327] J. Weston, F. Ratle, H. Mobahi, and R. Collobert, "Deep learning via semi-supervised embedding," in *Neural Networks: Tricks of the Trade*, Springer, 2012.

[328] T. Miyato, S.-i. Maeda, S. Ishii, and M. Koyama, "Virtual adversarial training: a regularization method for supervised and semi-supervised learning," *IEEE transactions on pattern analysis and machine intelligence*, 2018.

[329] M. J. Pazzani and D. Billsus, "Content-based recommendation systems," in *The adaptive web*, Springer, 2007.

[330] S. Rendle, "Factorization machines with libfm," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 3, no. 3, 2012.

[331] X. He and T.-S. Chua, "Neural factorization machines for sparse predictive analytics," in *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, ACM, 2017.

[332] J. Chen, H. Zhang, X. He, L. Nie, W. Liu, and T.-S. Chua, "Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention," in *SIGIR*, ACM, 2017.

[333] I. Obeid and J. Picone, "The temple university hospital eeg data corpus," *Frontiers in neuroscience*, vol. 10, 2016.

[334] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in *NeurIPS 24*, pp. 2546–2554, 2011.

[335] S. Andradóttir, "A review of random search methods," in *Handbook of Simulation Optimization*, pp. 277–292, Springer, 2015.

[336] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," in *NeurIPS 25*, pp. 2951–2959, Curran Associates, Inc., 2012.

[337] R. Calandra, N. Gopalan, A. Seyfarth, J. Peters, and M. P. Deisenroth, "Bayesian gait optimization for bipedal locomotion," in *Learning and Intelligent Optimization*, pp. 274–290, 2014.

[338] M. Nalbant, H. Gökkaya, and G. Sur, "Application of taguchi method in the optimization of cutting parameters for surface roughness in turning," *Materials & design*, vol. 28, no. 4, pp. 1379–1385, 2007.

[339] S. Mahapatra and A. Patnaik, "Optimization of wire electrical discharge machining (wedm) process parameters using taguchi method," *IJAMT*, vol. 34, no. 9, pp. 911–925, 2007.

[340] X. Zhang, L. Yao, X. Wang, J. Monaghan, D. Mcalpine, and Y. Zhang, "A survey on deep learning based brain computer interface: Recent advances and new frontiers," *arXiv preprint arXiv:1905.04149*, 2019.

[341] L. Yao, Q. Z. Sheng, X. Li, T. Gu, M. Tan, X. Wang, S. Wang, and W. Ruan, "Compressive representation for device-free activity recognition with passive rfid signal strength," *IEEE Transactions on Mobile Computing*, vol. 17, no. 2, pp. 293–306, 2017.

[342] G. H. Givens, J. R. Beveridge, Y. M. Lui, D. S. Bolme, B. A. Draper, and P. J. Phillips, "Biometric face recognition: from classical statistics to future challenges," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 5, no. 4, pp. 288–308, 2013.

[343] N. S. Latman and E. Herb, "A field study of the accuracy and reliability of a biometric iris recognition system," *Science & Justice*, vol. 53, no. 2, pp. 98–102, 2013.

[344] F. Sadikoglu and S. Uzelaltinbulat, "Biometric retina identification based on neural network," *Procedia Computer Science*, vol. 102, pp. 26–33, 2016.

[345] D. Ormerod, "Sounding out expert voice identification," *Expert Evidence and Scientific Proof in Criminal Trials*, 2017.

[346] J. Unar, W. C. Seng, and A. Abbasi, "A review of biometric technology along with trends and prospects," *Pattern recognition*, vol. 47, no. 8, pp. 2673–2688, 2014.

[347] D. H. Kerem and A. B. Geva, "Brain state identification and forecasting of acute pathology using unsupervised fuzzy clustering of eeg temporal patterns," in *Fuzzy and neuro-fuzzy systems in medicine*, pp. 19–68, CRC Press, 2017.

[348] J. Sohankar, K. Sadeghi, A. Banerjee, and S. K. Gupta, "E-bias: A pervasive eeg-based identification and authentication system," in *Proceedings of the 11th ACM Symposium on QoS and Security for Wireless and Mobile Networks*, pp. 165–172, ACM, 2015.

[349] Q. Gui, Z. Jin, and W. Xu, "Exploring eeg-based biometrics for user identification and authentication," in *Signal Processing in Medicine and Biology Symposium (SPMB), 2014 IEEE*, pp. 1–6, IEEE, 2014.

[350] P. Kidmose, D. Looney, M. Ungstrup, M. L. Rank, and D. P. Mandic, "A study of evoked potentials from ear-eeg," *IEEE Transactions on Biomedical Engineering*, vol. 60, no. 10, pp. 2824–2830, 2013.

[351] M. K. Bashar, I. Chiaki, and H. Yoshida, "Human identification from brain eeg signals using advanced machine learning method eeg-based biometrics," in *Biomedical Engineering and Sciences (IECBES), 2016 IEEE EMBS Conference on*, pp. 475–479, IEEE, 2016.

[352] P. Kumari and A. Vaish, "Brainwave based user identification system: A pilot study in robotics environment," *Robotics and Autonomous Systems*, vol. 65, pp. 15–23, 2015.

[353] K. P. Thomas and A. P. Vinod, "Utilizing individual alpha frequency and delta band power in eeg based biometric recognition," in *Systems, Man, and Cybernetics (SMC), 2016 IEEE International Conference on*, pp. 004787–004791, IEEE, 2016.

[354] H. Wang, C. Zhang, T. Shi, F. Wang, and S. Ma, "Real-time eeg-based detection of fatigue driving danger for accident prediction," *International journal of neural systems*, vol. 25, no. 02, p. 1550002, 2015.

[355] L. J. Trejo, K. Kubitz, R. Rosipal, R. L. Kochavi, and L. D. Montgomery, "Eeg-based estimation and classification of mental fatigue," *Psychology*, vol. 6, no. 05, p. 572, 2015.

[356] G. G. Knyazev, "Eeg delta oscillations as a correlate of basic homeostatic and motivational processes," *Neuroscience & Biobehavioral Reviews*, vol. 36, no. 1, pp. 677–695, 2012.

[357] J. Müller-Gerking, G. Pfurtscheller, and H. Flyvbjerg, "Designing optimal spatial filters for single-trial eeg classification in a movement task," *Clinical neurophysiology*, vol. 110, no. 5, pp. 787–798, 1999.

[358] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, pp. 4945–4949, IEEE, 2016.

[359] W. Chan, N. Jaitly, Q. V. Le, O. Vinyals, and N. M. Shazeer, "Speech recognition with attention-based recurrent neural networks," 2017. US Patent 9,799,327.

[360] B. Wang, K. Liu, and J. Zhao, "Inner attention based recurrent neural networks for answer selection," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, pp. 1288–1297, 2016.

[361] P. Li, L. Peng, J. Cai, X. Ding, and S. Ge, "Attention based rnn model for document image quality assessment," in *Document Analysis and Recognition (ICDAR), 2017 14th IAPR International Conference on*, vol. 1, pp. 819–825, IEEE, 2017.

[362] J. Ba, V. Mnih, and K. Kavukcuoglu, "Multiple object recognition with visual attention," *arXiv preprint arXiv:1412.7755*, 2014.

[363] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *arXiv preprint arXiv:1508.04025*, 2015.

[364] I. Jayarathne, M. Cohen, and S. Amarakeerthi, "Brainid: Development of an eeg-based biometric authentication system," in *Information Technology, Electronics and Mobile Communication Conference (IEMCON), 2016 IEEE 7th Annual*, pp. 1–6, IEEE, 2016.

[365] x. Li, "Signal processing in neuroscience," pp. 8–12, Springer, 2016.

[366] E. Başar, *EEG-brain dynamics: relation between EEG and brain evoked potentials*. Elsevier-North-Holland Biomedical Press, 1980.

[367] M. Steriade, "Alertness, quiet sleep, dreaming," in *Normal and Altered States of Function*, pp. 279–357, Springer, 1991.

[368] R. N. Sachdev, N. Gaspard, J. L. Gerrard, L. J. Hirsch, D. D. Spencer, and H. P. Zaveri, "Delta rhythm in wakefulness: evidence from intracranial recordings in human beings," *Journal of neurophysiology*, vol. 114, no. 2, pp. 1248–1254, 2015.

[369] T. Harmony, "The functional significance of delta oscillations in cognitive processing," *Frontiers in integrative neuroscience*, vol. 7, p. 83, 2013.

[370] T. T. Dang-Vu, M. Schabus, M. Desseilles, G. Albouy, M. Boly, A. Darsaud, S. Gais, G. Rauchs, V. Sterpenich, G. Vandewalle, *et al.*, "Spontaneous neural activity during human slow wave sleep," *Proceedings of the National Academy of Sciences*, vol. 105, no. 39, pp. 15160–15165, 2008.

[371] D. McGinty, R. Szymusiak, and D. Thomson, "Preoptic/anterior hypothalamic warming increases eeg delta frequency activity within non-rapid eye movement sleep," *Brain research*, vol. 667, no. 2, pp. 273–277, 1994.

[372] K. Yoshida, X. Li, G. Cano, M. Lazarus, and C. B. Saper, "Parallel preoptic pathways for thermoregulation," *Journal of Neuroscience*, vol. 29, no. 38, pp. 11954–11964, 2009.

[373] X. Zhang, L. Yao, C. Huang, S. Wang, M. Tan, G. Long, and C. Wang, "Multi-modality sensor data classification with selective attention," in *The 27th International Joint Conference on Artificial Intelligence, IJCAI-18*, pp. 3111–3117, 2018.

[374] S. Keshishzadeh, A. Fallah, and S. Rashidi, "Improved eeg based human authentication system on large dataset," in *Electrical Engineering (ICEE), 2016 24th Iranian Conference on*, pp. 1165–1169, IEEE, 2016.

[375] F. Pasqualetti, F. Dörfler, and F. Bullo, "Attack detection and identification in cyber-physical systems," *IEEE Transactions on Automatic Control*, vol. 58, no. 11, pp. 2715–2729, 2013.

[376] K. T. O. Dubbelink, A. Felius, J. P. Verbunt, B. W. Van Dijk, H. W. Berendse, C. J. Stam, and H. A. Delemarre-van de Waal, "Increased resting-state functional connectivity in obese adolescents; a magnetoencephalographic pilot study," *PLoS One*, vol. 3, no. 7, p. e2827, 2008.

[377] M. S. Reid, F. Flammino, B. Howard, D. Nilsen, and L. S. Prichep, "Topographic imaging of quantitative eeg in response to smoked cocaine self-administration in humans," *Neuropsychopharmacology*, vol. 31, no. 4, p. 872, 2006.

[378] A. Sanz-Martin, M. Á. Guevara, C. Amezcua, G. Santana, and M. Hernández-González, "Effects of red wine on the electrical activity and functional coupling between prefrontal–parietal cortices in young men," *Appetite*, vol. 57, no. 1, pp. 84–93, 2011.

[379] V. Knott, M. Cosgrove, C. Villeneuve, D. Fisher, A. Millar, and J. McIntosh, "Eeg correlates of imagery-induced cigarette craving in male and female smokers," *Addictive behaviors*, vol. 33, no. 4, pp. 616–621, 2008.

[380] D. C. Hammond, "The effects of caffeine on the brain: A review," *Journal of Neurotherapy*, vol. 7, no. 2, pp. 79–89, 2003.

[381] D. D. Emek-Savaş, B. Güntekin, G. G. Yener, and E. Başar, "Decrease of delta oscillatory responses is associated with increased age in healthy elderly," *International Journal of Psychophysiology*, vol. 103, pp. 103–109, 2016.

[382] E. Niedermeyer and F. L. da Silva, *Electroencephalography: basic principles, clinical applications, and related fields*. 2005.

[383] F. Akram, S. M. Han, and T.-S. Kim, "An efficient word typing p300-bci system using a modified t9 interface and random forest classifier," *Computers in biology and medicine*, vol. 56, pp. 30–36, 2015.

[384] G. Repovs, "Dealing with noise in eeg recording and data analysis," in *Informatica Medica Slovenica*, vol. 15, pp. 18–25, 2010.

[385] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur, "Recurrent neural network based language model.," in *Interspeech*, vol. 2, p. 3, 2010.

[386] S. Jirayucharoensak, S. Pan-Ngum, and P. Israsena, "Eeg-based emotion recognition using deep learning network with principal component based covariate shift adaptation," *The Scientific World Journal*, vol. 2014, 2014.

[387] W. Speier, N. Chandravadia, D. Roberts, S. Pendekanti, and N. Pouratian, "Online bci typing using language model classifiers by als patients in their homes," *Brain-Computer Interfaces*, vol. 4, no. 1-2, pp. 114–121, 2017.

[388] M. Moghadamfalahi, U. Orhan, M. Akcakaya, H. Nezamfar, M. Fried-Oken, and D. Erdogmus, "Language-model assisted brain computer interface for typing: A comparison of matrix and rapid serial visual presentation," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 23, no. 5, pp. 910–920, 2015.

[389] J. Nagi, F. Ducatelle, G. A. Di Caro, D. Cireşan, U. Meier, A. Giusti, F. Nagi, J. Schmidhuber, and L. M. Gambardella, "Max-pooling convolutional neural networks for vision-based hand gesture recognition," in *Signal and Image Processing Applications (ICSIPA)*.

[390] G. Hinton, N. Srivastava, and K. Swersky, "Rmsprop: Divide the gradient by a running average of its recent magnitude," *Neural networks for machine learning, Coursera lecture 6e*, 2012.

[391] C. Donos, M. Dümpelmann, and A. Schulze-Bonhage, "Early seizure detection algorithm based on intracranial eeg and random forest classification," *International journal of neural systems*, vol. 25, no. 05, p. 1550023, 2015.

[392] A. E. Elisha, L. Garg, O. Falzon, and G. Di Giovanni, "Eeg feature extraction using common spatial pattern with spectral graph decomposition," in *Computing Networking and Informatics (ICCNI), 2017 International Conference on*, pp. 1–8, IEEE, 2017.

[393] S. Michelmann, M. S. Treder, B. Griffiths, C. Kerrén, F. Roux, M. Wimber, D. Rollings, V. Sawlani, R. Chelvarajah, S. Gollwitzer, *et al.*, "Data-driven

re-referencing of intracranial eeg based on independent component analysis (ica)," *Journal of neuroscience methods*, vol. 307, pp. 125–137, 2018.

[394] P. Li, X. Wang, F. Li, R. Zhang, T. Ma, Y. Peng, X. Lei, Y. Tian, D. Guo, T. Liu, *et al.*, "Autoregressive model in the lp norm space for eeg analysis," *Journal of neuroscience methods*, vol. 240, pp. 170–178, 2015.

[395] Y. Zhang, B. Liu, X. Ji, and D. Huang, "Classification of eeg signals based on autoregressive model and wavelet packet decomposition," *Neural Processing Letters*, vol. 45, no. 2, pp. 365–378, 2017.

[396] J. Liu, Y. Su, and Y. Liu, "Multi-modal emotion recognition with temporal-band attention based on lstm-rnn," in *Pacific Rim Conference on Multimedia*, pp. 194–204, Springer, 2017.

[397] J. Li, Z. Struzik, L. Zhang, and A. Cichocki, "Feature learning from incomplete eeg with denoising autoencoder," *Neurocomputing*, vol. 165, pp. 23–31, 2015.

[398] V. J. Lawhern, A. J. Solon, N. R. Waytowich, S. M. Gordon, C. P. Hung, and B. J. Lance, "Eegnet: A compact convolutional network for eeg-based brain-computer interfaces," *arXiv preprint arXiv:1611.08024*, 2016.

[399] Z. Li, X. Tian, L. Shu, X. Xu, and B. Hu, "Emotion recognition from eeg using rasm and lstm," in *International Conference on Internet Multimedia Computing and Service*, pp. 310–318, Springer, 2017.

[400] Z. Y. Chin, K. K. Ang, C. Wang, C. Guan, and H. Zhang, "Multi-class filter bank common spatial pattern for four-class motor imagery bci," in *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 571–574, IEEE, 2009.

[401] Y. Wang, S. Gao, and X. Gao, "Common spatial pattern method for channel selelction in motor imagery based brain-computer interface," in *Engineering in medicine and biology society, 2005. IEEE-EMBS 2005. 27th Annual international conference of the*, pp. 5392–5395, IEEE, 2006.

[402] T. Song, W. Zheng, P. Song, and Z. Cui, "Eeg emotion recognition using dynamical graph convolutional neural networks," *IEEE Transactions on Affective Computing*, 2018.

[403] D. C. Ciresan, U. Meier, L. M. Gambardella, and J. Schmidhuber, "Convolutional neural network committees for handwritten character classification," in *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pp. 1135–1139, IEEE, 2011.

[404] R. Ning, C. Wang, C. Xin, J. Li, and H. Wu, "Deepmag: Sniffing mobile apps in magnetic field through deep convolutional neural networks," in *2018 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pp. 1–10, IEEE, 2018.

[405] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 6, pp. 1137–1149, 2017.

[406] H. Meisheri, N. Ramrao, and S. Mitra, "Multiclass common spatial pattern for eeg based brain computer interface with adaptive learning classifier," *arXiv preprint arXiv:1802.09046*, 2018.

[407] H. Ramoser, J. Muller-Gerking, and G. Pfurtscheller, "Optimal spatial filtering of single trial eeg during imagined hand movement," *IEEE transactions on rehabilitation engineering*, vol. 8, no. 4, pp. 441–446, 2000.

[408] K. K. Ang, Z. Y. Chin, C. Wang, C. Guan, and H. Zhang, "Filter bank common spatial pattern algorithm on bci competition iv datasets 2a and 2b," *Frontiers in neuroscience*, vol. 6, p. 39, 2012.

[409] L. C. Parra, C. D. Spence, A. D. Gerson, and P. Sajda, "Recipes for the linear analysis of eeg," *Neuroimage*, vol. 28, no. 2, pp. 326–341, 2005.

[410] C. Park, C. C. Took, and D. P. Mandic, "Augmented complex common spatial patterns for classification of noncircular eeg from motor imagery tasks," *IEEE Transactions on neural systems and rehabilitation engineering*, vol. 22, no. 1, pp. 1–10, 2014.

[411] K. P. Thomas and A. Vinod, "Eeg-based biometric authentication using gamma band power during rest state," *Circuits, Systems, and Signal Processing*, pp. 1–13, 2017.

[412] J. V. Odom, M. Bach, M. Brigell, G. E. Holder, D. L. McCulloch, A. P. Tormene, *et al.*, "Iscev standard for clinical visual evoked potentials (2009 update)," *Documenta ophthalmologica*, vol. 120, no. 1, pp. 111–119, 2010.

[413] K. Seeliger, U. Güçlü, L. Ambrogioni, Y. Güçlütürk, and M. Van Gerven, "Generative adversarial networks for reconstructing natural images from brain activity," *NeuroImage*, vol. 181, pp. 775–785, 2018.

[414] S. Nishimoto, A. T. Vu, T. Naselaris, Y. Benjamini, B. Yu, and J. L. Gallant, "Reconstructing visual experiences from brain activity evoked by natural movies," *Current Biology*, vol. 21, no. 19, pp. 1641–1646, 2011.

[415] J.-D. Haynes and G. Rees, "Neuroimaging: decoding mental states from brain activity in humans," *Nature Reviews Neuroscience*, vol. 7, no. 7, p. 523, 2006.

[416] T. Naselaris, R. J. Prenger, K. N. Kay, M. Oliver, and J. L. Gallant, "Bayesian reconstruction of natural images from human brain activity," *Neuron*, vol. 63, no. 6, pp. 902–915, 2009.

[417] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier gans," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 2642–2651, JMLR. org, 2017.

[418] X. Zhang and D. Wu, "On the vulnerability of cnn classifiers in eeg-based bcis," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 27, no. 5, pp. 814–825, 2019.

[419] W.-L. Zheng and B.-L. Lu, "Personalizing eeg-based affective models with transfer learning," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pp. 2732–2738, AAAI Press, 2016.

[420] M. Pacharra, S. Debener, and E. Wascher, "Concealed around-the-ear eeg captures cognitive processing in a visual simon task," *Frontiers in human neuroscience*, vol. 11, p. 290, 2017.

[421] K. B. Mikkelsen, S. L. Kappel, D. P. Mandic, and P. Kidmose, "Eeg recorded from the ear: Characterizing the ear-eeg method," *Frontiers in neuroscience*, vol. 9, p. 438, 2015.